Assisted migration of application systems to cloud environments

Von der Fakultät für Wirtschafts- und Sozialwissenschaften der Universität Stuttgart zur Erlangung der Würde eines Doktors der Wirtschafts- und Sozialwissenschaften (Dr. rer. pol.) genehmigte Abhandlung

Vorgelegt von

Adrián Juan Verdejo

aus Pobladura de Pelayo García (León), Spanien

Hauptberichter: Prof. Dr. Hans-Georg Kemper

Mitbericther: Prof. Dr. Georg Herzwurm

Tag der mündlichen Prüfung: 01.02.2023

Betriebswirtschaftliches Institut der Universität Stuttgart

2023

Outline

Table of contents

List of figures

List of tables

Acronyms

ACID	Atomicity, Consistency, Isolation, Durability.
ACM	Association for Computing Machinery.
ADM	Architecture-Driven Modernisation.
ADT	Atlantic Daylight Time.
AJAX	Asynchronous JavaScript and XML.
AHP	Analytic Hierarchy Process.
AMI	Amazon Machine Image.
ANP	Analytic Network Process.
API	Application Programming Interface.
AR	Augmented reality.
ARTS	Association for Retail Technology Standards.
AWS	Amazon Web Services.
BAS	Building Automation System.
BASE	Basically Available, Soft state, Eventual consistency.
BDD	Behaviour-driven development.
BI	Business Intelligence.
BOIKA	Bag Of Information and Knowledge Assets.
BMBF	Bundesministerium für Bildung und Forschung (Ministry of Education and Research).
CA	Certification Authorities.
CAD	Computer-Aided Design.
CADF	Cloud Auditing Data Federation.
CAMP	Cloud Application Management for Platforms.
CAP	Consistency (C), high Availability (A), and resiliency to network Partitions (P).
CAPEX	Capital expenditure or capital expense.

CD	Continuous Delivery or Continuous Deployment.
CDMI	Cloud Data Management Interface.
CEC	A cloud environment configuration.
CI	Continuous Integration.
CIMI	Cloud Infrastructure Management Interface.
CN	Collaborative Network.
CoIN	Collaborative Innovation Networks.
CPS	Cyber-Physical Systems.
CPU	Central Processing Unit.
CRM	Customer Relationship Management.
CRUD	Create, Read, Update and Delete.
CSA	Cloud Security Alliance.
CSC	Cloud Standards Coordination.
DAO	Data Object.
DAS	Dynamically Adaptive Systems.
DBMS	Database Management System.
DDOS	Distributed Denial of Service.
DHT	Distributed Hash Table.
DMTF	The Distributed Management Task Force.
DSL	Domain-Specific Language.
DS	Decision Support.
DSS	Decision Support System.
DWH	Data Warehouse.
EC	European Commission.
EC2	Elastic Cloud Compute.
EJB	Enterprise JavaBeans.
EMF	Eclipse Modelling Framework.
EMOF	Essential Meta-Object Facility.
ENISA	European Union Agency for Network and Information Security.
EPC	Engineering, Procurement and Construction.
ERG	European Reintegration Grants.

ERP	Enterprise Resource Planning.
ESR	Early-Stage Researcher.
ETL	Extract, transform, load.
ETSI	European Telecommunications Standards Institute.
EU	European Union.
FP7	(European) Seventh Framework Programme.
GDPR	General Data Protection Rights.
GI	Gesellschaft für Informatik.
GI FB WI	Fachbereich Wirtschaftsinformatik in der Gesellschaft für Informatik e.V..
GICTF	Global Inter-Cloud Technology Forum.
GQM	Goal, question, and metric.
GMF	Graphical Modelling Framework.
GPRS	General Packet Radio Service.
GUID	Globally Unique Identifier.
HR	Human Resources.
HTML5	HyperText Markup Language 5.
HTTP	Hypertext Transfer Protocol.
IaaS	Infrastructure as a Service.
ICT	Information and communications technology.
ID	Identifier.
IDS	Intrusion Detection Systems.
IEC	International Electro-technical Commission.
IEEE	Institute of Electrical and Electronics Engineers.
IoT	Internet of Things.
IP	Internet Protocol.
IS	Information System.
ISO	International Organization for Standardization.
ISOC	Internet Society.
IT	Information Technology.
ITIL	ITIL, formerly Information Technology Infrastructure Library.

ITN Initial Training Network.

ITSM Information Technology Service Management.

ITU International Telecommunications Union.

JDBC Java Database Connectivity.

JDEECo Java Dependable Ensembles of Emerging Components.

JEE Java Enterprise Edition.

JIT Just in Time.

JMS Java Message Service.

JSF JavaServer Faces.

JSON JavaScript Object Notation.

JVM Java Virtual Machine.

KDM Knowledge Discovery Meta model.

KM Knowledge Management.

KPI Key Performance Indicator.

KRI Key Risk Indicator.

KVM Kernel Virtual Machine.

LDAP Lightweight Directory Access Protocol.

MAN Mobile Area Network.

MAPE-K Monitor, Analyse, Plan, and Execute with Knowledge (loop).

MCDM Multi-criteria Decision Making.

MAN Management Information System.

MDA Model-Driven Architecture.

MDE Model-Driven Engineering.

MOF Meta-Object Facility.

MPLS Multi-protocol Label Switching.

n/a not applicable.

NESSI Networked European Software and Service Initiative.

NIST National Institute of Standards and Technology.

NLP National Language Processing.

OASIS	Organization for the Advancement of Structured Information Standards.
OCC	Open Cloud Consortium.
OCCI	Open Cloud Computing Interface.
OCL	Open Constraint Language.
OGF	Open Grid Forum.
OLAP	On-line analytical processing.
OMG	Object Management Group.
OPEX	Operating expense, operating expenditure, operational expense, or operational expenditure.
OS	Operating System.
OSGi	Open Services Gateway initiative.
OSI	Open System Interconnect.
OVF	Open Virtualization Format.
PaaS	Platform as a Service.
PAC	Programmable Automation Controllers.
PC	Personal Computer.
PKI	Public Key Infrastructures.
PR	Performance Ration.
PSDR	Product and Service Discovery and Recommendation Engine.
PSS	Product Specification System.
PST	Product Specification Tool.
PVC	Professional Virtual Community.
QoS	Quality of Service.
QoE	Quality of Experience.
RAP	Remote Application Platform.
RAM	Random-Access Memory.
RDS	(Amazon) Relational Database Service.
RDMS	Relational Database Management System.
REST	REpresentational State Transfer.

RESTful	REpresentational State Transfer Application Programming Interface.
RCP	Rich Client Platform.
RHEV	Red Hat Enterprise Virtualization.
RMI	Remote Method Invocation.
ROI	Return of Investment.
SaaS	Software as a Service.
SAFECode	Software Assurance Forum for Excellence in Code.
SAN	Storage Area Network.
SCM	Supply Chain Management.
SEP	Service-Enhanced Products.
SLA	Service-Level Agreement.
SME	Small and Medium Enterprise.
SMI	Service Measurement Index.
SMM	Structured Metrics Meta model.
SNIA	Storage Networking Industry Association.
SNMP	Simple Network Management Protocol.
SOA	Service-Oriented Architecture.
SOAP	Simple Object Access Protocol.
SQL	Structured Query Language.
SRC	Student Research Competition.
SQS	(Amazon) Simple Queue Service.
SSO	Single Sign-On.
SST	Service Specification Tool.
SWT	Standard Widget Tool-kit.
TDD	Test-Driven Development.
TCO	Total Cost of Ownership.
TOSCA	Topology and Orchestration Specification for Cloud Applications.
UI	User Interface.
UML	Unified Modelling Language.
URL	Uniform Resource Locator.

UTC Coordinated Universal Time.

VBE Virtual Breeding Environment.

VE Virtual Enterprises.

VLAN Virtual Local Area Network.

VM Virtual Machine.

VMM Virtual Machine Monitor.

VO Virtual organisations.

XaaS Everything as a Service.

XMI XML Metadata Interchange.

XML eXtensive Mark-up Language.

xRM Everything Relationship Management.

W3C World Wide Web Consortium.

WAN Wide Area Network.

WfMC Workflow Management Coalition.

WSDL Web Services Description Language.

WP Work Package.

WS Web Service.

WKWI Wissenschaftlichen Kommission Wirtschaftsinformatik im Verband der Hochschullehrer für Betriebswirtschaft e.V..

YAML YAML Ain't Markup Language. A human-readable data serialisation language.

Abstract

Organisations can now architect their software application systems so that they run on cloud-based software environments and make direct use of the advantages that these environments offer in terms of scalability, cost reduction, and business flexibility. Designing software with particular Infrastructure-, Platform-, or Software-as-a-Service cloud-based deployments in mind make it easier to exploit the potential of those environments without incurring extra effort. However, many organisations are already running their application systems on their premises but want to profit from the potential improvements that running them on cloud environments would offer them.

Some organisations run production application systems that are so complex that re-doing them to use a particular cloud environment as their backbone software infrastructure is not doable. Therefore, organisations can address the challenges of migrating complex component-based software application systems to cloud environments while using the potential of virtualised environments to scale in a cost-efficient manner. Some organisations migrate their application systems in an ad-hoc fashion by just moving them to a cloud environment they have not methodologically chosen.

Arguably, organisations do not profit from the capabilities that cloud environments offer to scale up or down and in or out when they adapt their application systems like that. For example, a support system could help organisations to adapt the behaviour of their application systems to the pay-per-use pricing models to improve them in terms of costs. The opportunities cloud environments offer come at the expense of challenges such as those imposing the need for organisations to adapt their application systems to the particularities of the cloud environment target for

migration. Some cloud environments might restrict the kind of permissions that a component running in their infrastructure possess such as restrictions to handle sub-systems in charge of persisting data —like not allowing to access the file system or to run a specific database management system— or limiting the ways in which components can communicate with external services —not allowing to open a socket, as an example. These constraints come from both the cloud provider's side and organisations migrating their application systems. These might include for example limitations to the cloud-enabled application system related to the data they can host out of the organisations' premises due to data privacy and sensitivity concerns.

Organisations can benefit from taking a methodological approach to migrate their application systems and analyse and plan it beforehand so as to better achieve their goals in moving data and computation while respecting their exogenous and endogenous constraints. The different moving parts of these migrated application systems make it difficult to assess the different criteria —cloud migration criteria for this dissertation— and goals that organisations want to achieve with the migration. The application systems present high variability with different cloud migration criteria affecting one another and imposing that organisations must find trade-offs to prioritise some criteria to the detriment of others. Additionally, the extent to which migrated application systems perform across the different dimensions that organisations consider when moving their application systems to cloud environments depend on the cloud environment configuration to which they migrate their application systems.

Cloud providers deliver service offerings at different levels of abstraction and multiple configurations per abstraction level. The abstraction levels refer to the three usual service models: the Infrastructure-, Platform-, and Software-as-a-Service levels. Cloud configurations at each of those levels might vary by offering different target virtual machines, instance types, programming environments, cloud storage options, machine images, runtime environments, or deployment architectures. All this variability might entail different consequences in how an organisation has to plan the cloud migration to meet its cloud migration criteria while respecting software constraints. The different levels on the cloud stack

bring about particular challenges and opportunities, whose effects a migrated application system can mitigate or use to its profit. Moreover, these effects might vary over time. Therefore, organisations might need to react to this changing environment and might require migrating from one cloud provider to another or from one particular configuration to another one that includes (or not) their local premises.

The cloud migration DS concept is proposed in this dissertation to support organisations in the migration of their software application systems to cloud environments and adapt their application systems target for migration. The cloud migration concept allows for the modelling of the variability involved in the cloud migration decision; that is, the target application system, the organisation cloud migration criteria, and the cloud environment configurations. The System Modelling Module allows for modelling the target application system at the component level while offering the possibility to reverse-engineer existing source code to extract the appropriate structure. This module also lets organisations model the architectural constraints that their application systems impose into the migrated system. The Migration Criteria Modelling Module applies the Analytic Hierarchy Process to multi-criteria decision analysis in order to consider the trade-offs that organisations must consider in regard to the aspects they value as the factor driving their decision to migrate to cloud environments. The Cloud Environments Modelling Module allows for modelling the cloud environment configurations cloud providers supply so that the proposed concept can generate cloud migration alternatives. Alternatives stemming from the coupling of the cloud service descriptions and the other two input models previously explained.

The cloud migration alternatives generator takes into account the three models to compute valid cloud migration alternatives that respect constraints to the application system and migration. The migration alternatives generator combines different deployment options to the selected cloud environments and the local premises and assesses the potential of those alternatives according to the cloud migration criteria that organisations state. These are alternatives to cloud migration that plan to deploy software components to the cloud environments and the local premises. The cloud migration concept assesses the cloud migration alternatives either automatically or with human intervention. Such human intervention has organisations manually weighing cloud migration alternatives for the particular cloud migration criteria for which there are not any automatic metrics to assess them.

The experiments and evaluation show the applicability of the cloud migration DS concept and its relevance for research and market domains. The evaluation follows prototyping and scenarios research methods to develop and use the prototype of the proposed concept. The prototype brings together the above modules for experimentation purposes. Prior to putting the prototype into play in real settings, a preliminary analysis is conducted to study whether the prototype is a functional and faithful implementation of the proposed concept. The evaluation of the proposed concept uses three realistic and complex scenarios. In each scenario, an organisation intends cloud migrating its heterogeneous application systems and has the potential to benefit from the cloud migration support offered by the concept proposed in this dissertation.

Zusammenfassung

Die cloud-computing-basierte Nutzung von IT-Ressourcen hat im letzten Jahrzehnt stetig an Bedeutung gewonnen. Der Vorteil einer Cloud-Nutzung besteht vor allem darin, dass die IT-Dienstleistungen dynamisch skalierbar sind, eine schnelle Erschließung von Innovationen erlauben und es den Anwendern ermöglicht wird, von den Economies-of-Scale des Cloud-Providers zu profitieren. Bei einer Cloud-Nutzung zahlen Anwender- Organisationen nur für tatsächlich genutzte Dienste, was dabei helfen kann Betriebskosten zu senken – bei gleichzeitiger Steigerung der Flexibilität.

Die Anwender werden mit der Entscheidung konfrontiert, welche Teile ihrer Anwendungen sie wie in welche Cloud-Umgebungen migrieren wollen. Dabei kommen drei Dimensionen zum Tragen: *Erstens* die Merkmale des zu migrierenden Systems selbst: Die Architektur, die Softwareanforderungen, bestehende Restriktionen. *Zweitens* die relevanten Kriterien (die Cloud-Migrations-Kriterien dieser Dissertation), mit denen die Ziele erfasst werden, die durch die solche Migration erreicht werden sollen. Beispielsweise könnten in einem Fall Kosteneffizienz, Sicherheit und Vertraulichkeit im Mittelpunkt stehen, während in einem anderen Flexibilität und eine Vermeidung von Vendor-Lock-in-Effekten priorisiert werden. *Drittens* die Angebote der Cloud-Provider mit ihren oftmals umfangreichen Optionen für Servicemodelle und Konfigurationen. Diese drei Dimensionen – die Merkmale der zu migrierenden Anwendungssysteme, die Cloud-Migrations-Kriterien und die Eigenschaften der Cloud-Angebote – erzeugen einen großen und komplexen Suchraum, weshalb es sinnvoll ist, hier ein Entscheidungsunterstützungskonzept bereitzustellen.

Idealerweise unterstützt dies nicht nur die Entscheidung, ob ein System in Gänze in eine Cloud migriert werden soll oder nicht, sondern erlaubt eine komponentengenaue Betrachtung, die beispielsweise berücksichtigt, dass unkritische, ressourcenintensive Prozesse in eine (Public) Cloud überführt werden, während solche mit höheren Sicherheitsstandards im Hoheitsbereich des Anwenders verbleiben können (Private Cloud). Außerdem sollten Multi-Cloud-Umgebungen in die Betrachtung einbezogen werden können. Bei einer solch breiten Problemabgrenzung entsteht eine nicht mehr einfach manuell handhabbare Anzahl an Migrationsoptionen, die sich ohne Hilfsmittel nur schwierig darstellen und bewerten lassen. Vor diesem Hintergrund ist es das **Ziel dieser Dissertation**, ein tool-basiertes Konzept zu entwickeln und zu evaluieren, das den komplexen Entscheidungsprozess der Migration von Anwendungssystemen in Cloud-Umgebungen unterstützt. Das beinhaltet dabei ein Systemkonzept für das Entscheidungsunterstützungs-Tool, das prototypenbasiert erprobt und evaluiert wird.

Ausgehend von einem gestaltungsorientierten Forschungsansatz wird das oben genannte Ziel in mehrere Teilziele heruntergebrochen. Methodisch werden basierend auf einer Literaturanalyse und einer Projektanalyse zunächst konkrete Anforderungen an die Entwicklung des Konzepts erhoben. Für das Systemkonzept wird dabei zwischen Modulen unterschieden. Das Tool wurde auf Basis der Anforderungen prototypisch realisiert. Der Prototyp demonstriert dabei die Machbarkeit und wurde auch im Rahmen der Evaluation eingesetzt. Für diese wurden Befragungen und Beobachtung genutzt. Hierbei wurde auf drei realitätsnahe Anwendungsszenarien aufgesetzt.

Im Konzept wird zwischen drei Modellen unterschieden, aus deren Zusammenspiel die Migrationsoptionen abgeleitet und bewertet werden und die im Tool visuell erfasst und modifiziert werden können: 1) ein Modell des zu migrierenden Anwendungssystems; 2) das System der Cloud-Migrations-Kriterien und 3) ein Modell für das Cloud-Angebot. Das System bietet dabei ein Modul für eine komponentenbasierte Modellierung des Anwendungssystems inklusive möglicher Restriktionen. Das Modul ist dabei auch in der Lage, Quellcode für ein Reverse Engineering der Anwendungssystem-Architektur zu nutzen. Für die Modelling der Kriterien wird auf ein Modul bereitgestellt, das den Ansatz des Analytic Hierarchy Process (AHP) implementiert, der eine multikriterielle Entscheidungsanalyse unterstützt und dabei auch Trade-Offs zwischen den Kriterien berücksichtigt. Für die Modellierung der Cloud-Umgebungen der unterschiedlichen Cloud-Anbieter wird ein separates Modul bereitgestellt, mit dem sich die verschiedenen angebotenen Konfigurationen erfasst werden können.

Basierend auf den drei oben genannten Modulen werden in einem weiteren Moulen Alternativen zur Cloud-Migration entwickelt. Diese berücksichtigt die drei oben genannte Inputmodelle und nutzt diese auch zur Eingrenzung des Suchraums. Auf dieser Grundlage leitet es verschiedene Deployment-Architekturen dar, unterzieht diese einer Konsistenzprüfung mithilfe von Rankings anhand der Cloud-Migrations-Kriterien und dem AHP, sortiert die relevanten Alternativen und präsentiert diese visuell. Die Anwender-Organisation kann mithilfe der Lösung jederzeit die Kriterien und Rahmenbedingungen anpassen und so angepasste Hierarchien von Cloud-Migrationsoptionen generieren lassen.

Die Anwendung des entwickelten Konzepts wurde für drei Anwendungsszenarien erprobt, verdeutlicht und dabei evaluiert. Die Szenarien betreffen eine Cloud-Migration von 1) Anwendungssystemen zu Platform-as-a-Service-Angeboten, 2); die Migration von Business Intelligence-Umgebungen und 3) Systeme für Kollaboration-Netzwerke. Hierbei wurden zusätzliche Erkenntnisse gewonnen, die insbes. vertiefende Einblicke in die Einsatzmöglichkeiten des Konzeptes sowie allgemein in Herausforderungen von Cloud-Migrations-Projekten erlauben.

Die Erkenntnisse dieser Dissertation leisten einen Beitrag zum Stand der Forschung zur Cloud-Migration von Anwendungssystemen im Allgemeinen und zur Entscheidungsunterstützung für solche Projekte im Besonderen. Das vorgeschlagene Konzept nutzt modellbasierte Ansätze und AHP, um Architekturoptionen für existierende Anwendungssysteme zu generieren, die in Teilen oder im Ganzen in eine oder mehrere Clouds verlagert werden sollen. Die Ergebnisse legen nahe, dass ein solches Konzept erhebliche Nutzenpotenziale bietet.

1. Introduction

1.1 Motivation and problem statement

Cloud computing, according to the most widely accepted definition by the National Institute of Standards of Technology of the United States of America, provides:

> "a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction"[1]

Over the last ten years, organisations have been increasingly adopting cloud computing as their computational model. Its use offers the chance to implement relatively easy changes to reduce cost while improving on flexibility, business agility, or scalability. By the time of writing this thesis, in addition to early adopters and start-ups, an increasing number of large for-profit enterprises and non-profit organisations intend to run their services in cloud-based environments to try to benefit from improving the technologies they use[2].

Cloud migration entails some challenges, in addition to the potential for improvement, when organisations start running their application systems in cloud environ-

[1] The initial definition of cloud computing: US-NIST (2009), p. 2

[2] cf. Novais et al. (2019), pp. 296–300; cf. Sadeeq et al. (2021), pp. 176–181; Marston et al. (2016), pp. 180–186, URL see Bibliography; Fox et al. (2009), p. 19; Vaquero et al. (2008), p. 54

ments. The decision of how to migrate application systems to cloud environments is a complex one due to the different ways of measuring the success of a cloud migration project according to the priorities of the responsible organisation. While an organisation might want to reduce the cost of running their infrastructure by using idle computational resources. Another organisation, with different priorities, might be more interested in improving the performance of their software infrastructure by implementing cloud-based upscaling mechanisms. These are different drivers motivating organisations to cloud migrate a particular application system and that, very often, bring them to commit to trade-offs. For example, organisations usually have to make compromises between performance and cost. Increasing the number of machines tackling a task could finish that computation quicker at the expense of higher costs to run the architecture. Similarly, increasing the potential of organisations to collaborate using data stored in cloud infrastructures might put those data at higher levels of risk. Indeed, a security and privacy versus collaboration trade-off. In addition to these drivers of the cloud migration decision, organisations usually consider the particularities of the application system they intend to migrate and the cloud infrastructure chosen for the migration. These particularities include the requirements, architecture, and design constraints of the target applications system[3]. To sum up, decision makers can benefit from being assisted to assess the repercussions of their choices given the complexity of their multidimensional *decision-making problem*: the migration of application systems to cloud environments. The assistance arguable helps them analyse whether they align with their priorities related to the cloud migration, the architecture of their application systems, and the selected cloud environments[4].

This research work intends to address the cloud migration decision-making problem by assisting the selection of an alternative to cloud migrate an application system according to multiple decision criteria dependent on many interdependent and structured criteria[5]. In addition, the dissertation studies how to adapt the target

[3]cf. Novais et al. (2019), pp. 296–300; Gholami et al. (2016), pp. 31–33; Andrikopoulos et al. (2013a), p. 493; Juan-Verdejo et al. (2014b), p. 472; Odun-Ayo et al. (2018), pp. 2–3; Andrikopoulos et al. (2013a), p. 493

[4]cf. Juan-Verdejo et al. (2014b), p. 467

[5]cf. Balobaid (2020), pp. 120–123; García-Galán et al. (2016), pp. 200–202; Khajeh-Hosseini et al. (2011), p. 541; Sáez et al. (2018), pp. 1–2; Binz et al. (2011), p. 4; Juan-Verdejo et al. (2014b), p. 467

application system to the selected cloud environment while taking into account the full breadth of effects of the architectural changes. This dissertation applies to organisations that intend to reorganise the application system components to keep the functionality of the application system while achieving their cloud migration goals. In doing that, an organisation migrating an application system to cloud environments might benefit from provisioning its multiple application system components to different cloud environments to also comply to privacy and security constraints as well as to applicable Service-Level Agreements (SLAs), which usually include Quality-of-Service (QoS) requirements[6]. Organisations can benefit from using multi-criteria decision support that helps them structure outside and inside information to improve how they realise their requirements. Decision support helps by holistically analysing the implications related to the adaptation of application systems to cloud environments from business- and economics-related, technical, provider-related, and security and privacy standpoints. The concept proposed in this dissertation supports organisations in their decisions on how to cloud migrate their application systems. The proposed cloud migration decision support concept helps organisations facing such complex decision-making problems to overcome their challenges related to cloud migration, as explained below.

1.1.1 The definition of organisations' criteria.

That is, the definition of organisations' criteria to migrate their application system to cloud environments and how they trade some of these criteria off with others. Organisations' criteria to assess the migration of their application system to cloud environments can drive the cloud migration decision-making process by being tailored to the organisation and to the application system they intend to migrate (the target application system)[7].

The difficulty resides in how to define the importance of the criteria when cloud migrating an application system and how to assess how they interplay. In order to assess the cloud migration process, it can be necessary to understand

[6] cf. Sharma et al. (2020), pp. 26–30; Andrikopoulos et al. (2013b), p. 496; Khajeh-Hosseini et al. (2010c), p. 3; Juan-Verdejo and Baars (2013), p. 35

[7] cf. Li et al. (2017), pp. 185–187; Garg et al. (2011), p. 218

how organisations trade some of these criteria off with others according to their priorities[8].

1.1.2 The selection of particular cloud environments to deploy the computation.

Cloud environments offer different functional and non-functional properties that an organisation can leverage when running its target application system in their infrastructures[9]. The cloud environments selection is a fundamental factor influencing the selection of the target computational environments and how to design the software components placement for them (as mentioned in the paragraph immediately below). These decisions, if taken right, can improve target application systems with respect to the needs of the organisation cloud migrating them[10].

1.1.3 The selection of a cloud-based architecture fulfilling the organisation's criteria.

An organisation might benefit from being helped to design a cloud-based architecture according to the multiple factors that determine the success of its cloud migration project. Success in terms of whether the architecture facilitates fulfilling the organisation's requirements for cloud migration and offering the function of the target application system[11]. The functionality of the application system has to be guaranteed and its constraints respected. Organisations might profit from being assisted in the selection from multiple alternatives to cloud migrate its application systems to measurably improve them with the migration[12]. Achieving the goals of the organisation is arguably facilitated by performing a beneficial cloud migration selection (as described above) and by deciding on the software components placement. The components could be run on both the cloud-based and on-premises computing environments. The component placement at the local premises intends to overcome some organisations' reluctance to move their data

[8] cf. Ardagna et al. (2017), pp. 381–383; Fox et al. (2009), p. 1; Khajeh-Hosseini et al. (2010a), p. 450

[9] cf. Solomon (2017), pp. 35–45; Wittern et al. (2012), p. 141; Quinton et al. (2013), p. 26

[10] cf. Botto-Tobar and Insfran (2017), pp. 171–173; REMICS Consortium (2013), p. n/a, URL see Bibliography; Mohagheghi and Sæther (2011), p. 507

[11] cf. Sharma et al. (2020), pp. 26–30; Andrikopoulos et al. (2013a), p. 535

[12] cf. Li et al. (2017), p. 190; Garg et al. (2011), p. 210

to cloud environments due to trust issues. Hence, software components can run at the local premises of organisations while other computation and data software components can be moved to cloud environments depending on their computation needs and sensitivity-related requirements.

1.2 Aim of the dissertation

This doctoral dissertation aims to provide a cloud migration decision support (DS) concept to assist organisations to migrate their application systems to cloud environments. The concept also considers the local premises as a target computinv environment and analyses the heterogeneous and interrelated criteria that drive their cloud migration projects. It helps organisations pick the best alternative to cloud migrate their application systems according to those personalised criteria and after assessing all the alternatives.

The main contribution of this dissertation is a cloud migration decision support concept that uses model-based approaches and the Analytic Hierarchy Process (AHP)[13] to assist and enhance the design and structured management of the partial migration of existing application systems to cloud environments. Partial migration refers to the fact that the concept also assesses the effects of placing computing, data, and networking components of the application system at the local premises as well as at the cloud premises. The cloud migration concept outputs a prioritised list of alternatives for the application system's cloud migration by using three input models that capture:

- the application system's requirements,
- the organisation's requirements,
- and the cloud environments capabilities and properties.

Each of the alternatives to cloud migrate an application system includes a component placement model that re-scatters the application system components to the local and cloud-based premises. The development of the cloud migration concept is planned to iteratively and incrementally fulfil the aim of this dissertation. The aim of this dissertation comes down to answering the following four research questions.

[13]cf. Saaty (1977), pp. 234–281; Saaty and Gonzalez Vargas (2001), pp. 23–39

Research question 1: How can a decision support concept assist organisations in the decision of how to cloud migrate their application systems?

The proposed cloud migration decision support (DS) concept answers this question by applying the Analytic Hierarchy Process (AHP) as the appropriate multi-criteria decision-making technique. The cloud migration concept assists organisations in their decision of how to migrate their application systems to cloud environments within a structured process. This process facilitates taking into account the multiple factors that affect the decision-making process. The proposed concept semi-automatically assists organisations in the selecting one of the multiple alternatives to migrate the application system to both their local premises and the cloud environments.

Research question 2: What are the decision-making criteria driving organisations to decide how to migrate their application systems to cloud environments?

These are the criteria that drive the decision of organisations regarding how to migrate an application system to cloud environments. To answer this research question, the factors that affect the decision-making process are defined and formalised in this dissertation according to the requirements of the organisation that migrates the application system to cloud environments. Additionally, this research question entails defining how these decision-making criteria affect each other and how organisations trade some of them off with others. This dissertation will describe types of criteria with interrelated criteria and will formalise how to measure them, their interdependencies, and their subjective or objective characteristics. The cloud migration concept helps organisation describe their priorities when migrating application systems to cloud environments. In addition, it gathers organisation's feedback to evaluate subjective factors that affect their decision making of how to implement the migration to cloud environments. In the case of objective decision-making factors, the concept mathematically formalises how to measure them according to different technical and non-technical criteria.

Research question 3: What are the criteria or requirements organisations consider to select cloud environments to migrate their application systems?

A framework to model the cloud environments and their particular offerings or cloud environment configurations could be a part of the proposed cloud migration DS concept. The concept assists organisations in taking into account the cloud environments and their configurations to select the cloud environment best aligned with their needs. The proposed concept does not target independently the cloud selection but rather uses the cloud environments modelling as an input model to consider the different alternatives that organisations have in order to migrate their application systems and then help them rank them according to the decision-making drivers or criteria that organisations define.

Research question 4: Which requirements and characteristics of the application system prior to the cloud migration do organisations consider in the decision-making process?

The application system target for migration to cloud environments fulfils its requirements prior to the migration and presents some constraints and design characteristics that affect how to migrate this application system to cloud environments. The answer to this research question combined with the answer to research questions number 2 and 3 provide the input that the cloud migration concept will use to assist organisations in migrating their application systems according to their requirements. This research question entails selecting the paradigm to model the status-quo application system to decide the appropriate way to migrate it while delivering the existing functionalities and improving it in the dimensions that the migrating organisation requires. The answer entails deciding how to architect the cloud-migrated application system, including its partial migration to cope with requirements related to privacy, data security, or trust. As mentioned above, partial migrations use the premises local to the organisation in addition to cloud environments.

Figure 1.1 shows the four goals of this dissertation, which stem from the four research questions as a natural evolution of following the design-oriented information systems research principles (as Section 1.5 explains).

[14]Own illustration

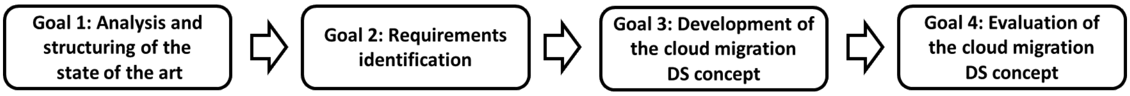

Figure 1.1: The four goals of this dissertation to answer its four research questions[14]

The four goals incrementally build the foundational work knowledge used to answer the four research questions as the research work moves from analysing and structuring the state of the art to evaluating the proposed cloud migration concept. The four research questions relate to each other; questions 2 to 4 serve as the input to question number 1 while, at the same time, this research question sets the foundation to answer the remaining questions. The research design of this dissertation makes an effort to iteratively examine these questions in depth to provide more details that help discover the answers to them. These iterations start with the analysis and structuring of the state of the art to then identify the requirements of the proposed concept. Based on these requirements, the design of the proposed concept will be proposed. Its implementation in a prototype will be used to evaluate the proposed concept with the analysis of the prototype and its use in three real-life evaluation scenarios. Figure 1.2 in Section 1.5, Chapter 4, puts together the research design and the goals of this dissertation in a cohesive manner.

1.3 Scientific research context

This scientific work is part of the German Wirtschaftsinformatik discipline —that is, the German Information Systems (ISs) discipline[15]. It relates to the fields of business informatics or Information Systems as disciplines that combine information technology (IT), informatics and management concepts. This dissertation deals with the "information systems in economics, management, and the private realm"[16]. Information systems refer to socio-technical systems which include human and electronic components. They enhance information and communication to maximise

[15] According to the Scientific Commission of the German Information Systems (WKWI or wissenschaftlichen Kommission Wirtschaftsinformatik im Verband der Hochschullehrer für Betriebswirtschaft e.V.) and the Informatics Association within the Field of Information Systems (GI FB WI or Fachbereich Wirtschaftsinformatik in der Gesellschaft für Informatik e.V.)

[16] Schoder et al. (2011), p. 1

economic criteria[17]. Information Systems study the broad subject of organised systems that collect, organise, store, and communicate information; additionally, they study complementary networks of people and organisations to collect, filter, process, create, and distribute data. IS is an overarching concept that integrates operational tasks, the stakeholders responsible for them, and information and communication techniques in organisational structures that participate in complex design processes[18].

This research work on the migration of information systems focuses on application systems; that is, software, hardware, and data that carry out concrete operational tasks or range of tasks[19]. This dissertation follows the principles of design-oriented information systems research[20] and not those of behavioural sciences. Behaviour-oriented research aims at observing the properties of information systems and the users' behaviour to develop theories and verify them. In turn, design-oriented business informatics or information systems design innovative artifacts to improve information systems[21]. The business informatics or information systems memorandum[22] takes into account the standpoint of 111 business administration professors confirming the relevance of design-oriented research in business informatics and its associated criteria classified into the following explained concepts[23].

1.3.1 Stakeholder groups

Relevant stakeholders within the information systems field and design-oriented information systems research include individuals and organisations that provide resources for the research while expecting favourable results for themselves. The stakeholder groups include cloud consumers such as companies and other organisations that intend to use cloud environments to run their application systems. The cloud migration concept focuses on addressing their by supporting their decision-making processes about how to cloud migrate their application systems.

[17] cf. Krcmar (2015), p. 25; Abts and Mülder (2009), p. 12; Mertens et al. (2001), p. 47
[18] cf. Mertens et al. (2001), p. 47; Ferstl and Sinz (2012), p. 3; Laudon et al. (2009), p. 17
[19] cf. Stahlknecht and Hasenkamp (2005), p. 326; Ferstl and Sinz (2012), p. 6; Laudon et al. (2009), p. 16; Klarl (2011), p. 10; Mertens et al. (2001), p. 46
[20] cf. Becker et al. (2009a), p. 1
[21] cf. Österle et al. (2011), p. 8; Hevner et al. (2004), p. 76 and pp. 75–100; Peffers et al. (2007), pp. 55–59; Jones and Gregor (2007), p. 312–314
[22] cf. Österle et al. (2011), pp. 7–10
[23] All the concepts follow the definitions of Österle et al.: cf. Österle et al. (2011), pp. 7–8

These processes include selecting a fitting cloud environment configuration running any cloud service model. Typically, models like Infrastructure, Platform, or Software as a Service.

1.3.2 Research object

Design-oriented information systems research considers organisations and individuals in society and economy and their relation to information systems. Information systems are socio-technical in nature as they include people, or human task bearers, and information in addition to the communications technology or technical task bearers and organisational concepts. That is, information systems consider functions, structures, processes, and the interrelationships between them. An application system is the research object of this dissertation[24]. An application system together with IT governance and the actual IT system conform an information system.

1.3.3 Research objectives

Design-oriented information systems research aims to develop and provide instructions for designing and operating information systems and innovative concepts for particular instances of ISs. The instructions for action are normative and applicable means-ends conclusions. The development of specific IS results follows design-oriented information systems research to build upon a conception of how the system should be to then create the system according to this model and the existing restrictions and limitations. Accordingly, the research concept will be developed into a research prototype to prove the fitness-for-use of the proposed cloud migration concept. It will have to respect the restrictions and requirements of decision-making processes and concepts designed to cloud migrate application systems[25]. Its fitness-for-use will be evaluated by applying the proposed concept, via its prototypical implementation, to real-life cloud migration projects.

1.3.4 Result types

Design-oriented information systems research advocates for developing artifacts such as constructs —from concepts to terminologies and languages—, models,

[24] cf. Österle et al. (2011), p. 8
[25] Hevner et al. (2004), p. 76 and pp. 75–100; cf. Österle et al. (2011), pp. 7–10

methods, and instantiations —that is, particular solutions as prototypes or production systems. The concrete manifestation of these artifacts in this dissertation include the cloud migration DS concept as the decision-making support solution, its methodology to drive the cloud migration, and the input, output, and intermediary models to facilitate the decision-making process[26].

1.3.5 Research process

Design-oriented information systems research follows an iterative process with four basic phases that include the analysis, design, evaluation, and diffusion phases. This dissertation follows this iterative research process and its phases to deliver the aim of the dissertation, as Section 1.2 describes, according to the research design and methods used. Section 4.7 in Chapter 4 shows the research design in depth and adds details to the research methodology explained in Section 1.5.

1.3.6 Research methods

Design-oriented information systems research uses research methods coming from business, social, computer, and engineering sciences. In the analysis phase, researchers use methods such as surveys, scenarios, case studies, expert interviews, and information systems analysis like database analysis. This dissertation understands research as qualitative and empiric effort to find the relevant requirements for the development of the identified conceptual needs. The requirements were gathered by analysing literature and case studies from industrial and research projects. The core of the research method includes the development of the conceptual research approach and its prototypical implementation and evaluation. The evaluation applies the proposed concept to three scenarios of cloud migration.

1.4 Main contributions

Figure 1.2 shows the overall research design of this dissertation and how it aims at fulfilling the main contributions of this dissertation. The research activities fall into three particular blocks and result in applying the research methodology that Section 1.5 explains. The figure also shows how these working blocks of the

[26] Peffers et al. (2007), pp. 55–59; Jones and Gregor (2007), p. 312–314

research design relate to the aim of this dissertation and to the four goals shown in Section 1.2.

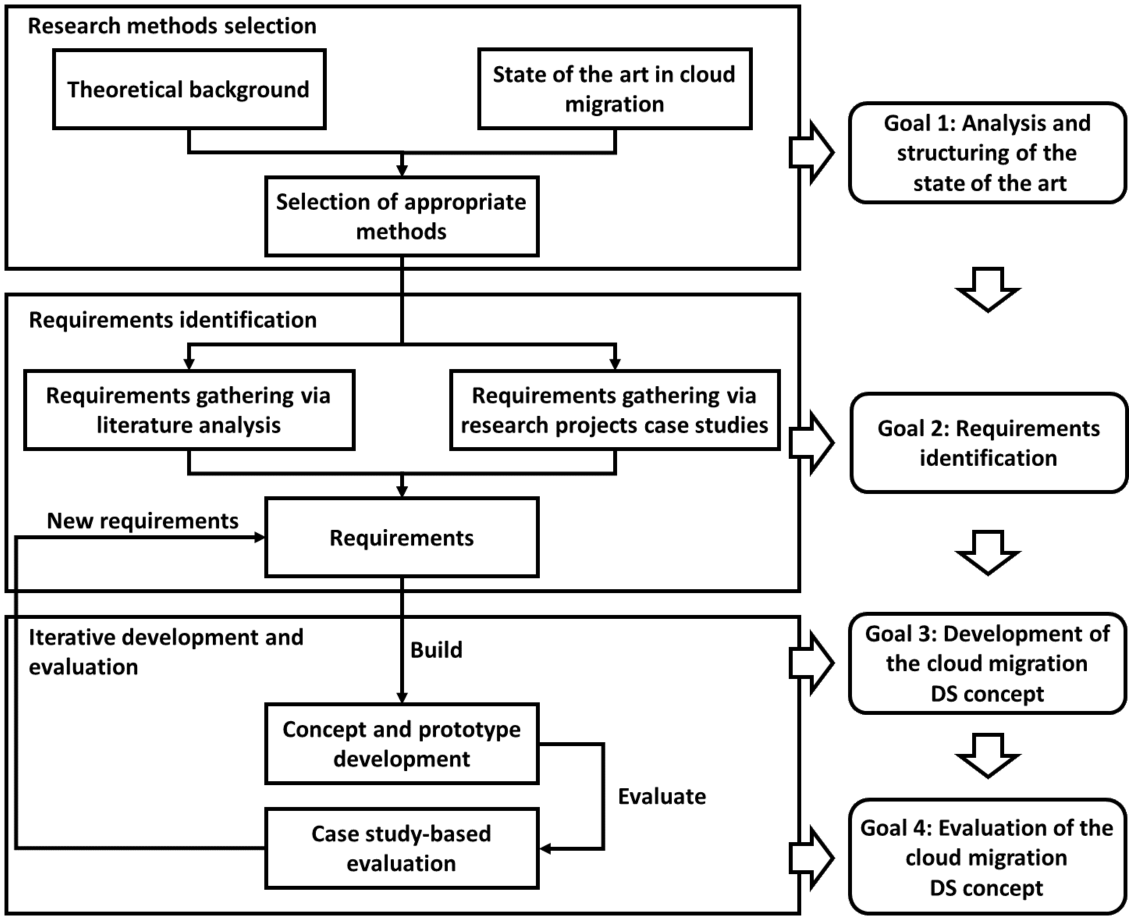

Figure 1.2: Research design stemming from the research methodology[27]

The main contribution of this dissertation will be *the multi-criteria and model-based decision-making support concept to the migration of existing application systems to cloud environments*; that is, the cloud migration DS concept.

Specifically, the concept will provide structured management of the migration of application systems to cloud environments by offering a cloud migration decision support concept based on the Analytic Hierarchy Process (AHP)[28]. The main contribution of this thesis can be broken down as follows.

Assisting and partially automating the cloud migration decision-making process. The proposed decision support concept suggests how to adapt the existing application system by moving its components and software tiers to virtualised

[27]Own illustration
[28]cf. Saaty (1977), pp. 234–281; Saaty and Gonzalez Vargas (2001), pp. 23–39

cloud environments. The complex decision depends on multiple interdependent criteria that the decision support concept assesses to facilitate the cloud migration.

Selecting the right cloud environment. This selection depends on the application system's characteristics, the decision criteria of organisations that migrate the application system to cloud environments, and the cloud environment characteristics. The decision support concept relies on the three models described below to select the cloud services which best fit the migrated application system. This research work delivers a multi-cloud decision support concept by mixing and matching cloud environments to select the cloud-enabled architecture that maximises the score of the criteria that organisations deemed as important to move their application systems to cloud environments. This research work can select cloud environments that maximise application systems portability and interoperability.

Adapting the application system to the target cloud environment in compliance with the organisation's requirements to migrate it. The solution space contains different alternatives for cloud-enabling an application system according to the requirements of the organisation intending to migrate it. The decision support concept considers all the potential components' placement models which respect the application system's constraints. The cloud migration concept suggests these multiple alternatives to carry out the cloud migration. It shows them in descending order according to how the alternatives achieve the goals of the organisation migrating its application system to the available target cloud environments. The concept uses the three models that Figure 1.3 shows: the on-premises architecture model of the migrated application system, its properties, and requirements; the model of the multiple decision criteria driving the migration; and the cloud environments model. These models work as the input of the proposed concept that, as an output, produces the recommended migration decisions. The model of the criteria that shape the decision of how to migrate an application system to cloud environments incorporates the subjective and objective criteria that concern the organisation in charge of the migration; the importance of each criterion to one another; and the metrics to automatically quantify some of them. The flexibility of this model that incorporates the criteria and priorities of organisations when migrating application systems to cloud environments allows for its easy extension. Likewise, the cloud environments model describes the available cloud services based on variability points, the services properties, and capabilities.

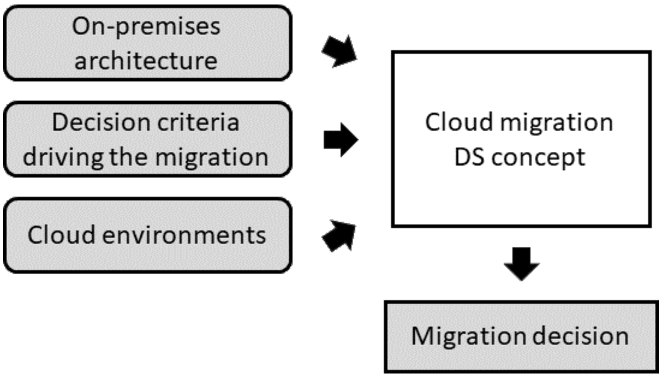

Figure 1.3: Cloud migration DS concept input models[29]

Estimating how the migrated application system fulfils the organisation's migration criteria. The decision support concept helps quantifying the effects of migrating an application system to cloud environments following a particular architectural design and according to the priorities of organisations when migrating their application systems to cloud environments.

Improving the application system's requirements and the organisation's criteria fulfilment. The decision support concept proposes how to re-scatter the different components of the target application system to the local and the cloud computing environments while respecting the organisation's policies and the constraints of the application system. The concept uses nearby computation and incorporates cross-functional components for data management and security enhancement. The concept partially migrates application systems to cloud environments in order to cope with data sensitivity, security, privacy, and latency requirements. That is, sensitive data is kept on the local premises to respect the privacy requirements of the migrating organisation, which potentially avoids deploying dependent components on geographically distant computation environments to minimise potential performance erosion due to the newly introduced distance for data transfers. This is a co-result of this dissertation acting as an add-on to the main contribution to cloud migration. This dissertation explains this concept with the example in Section 6.9 and some publications on how to use nearby

[29]Own illustration

cloudlets[30], which are also related to edge, fog, or nearby computing. The cloud migration concept suggests, in some evaluation scenarios, the use cross-functional components implementing security, and data synchronisation mechanisms that organisations can reuse in their particular cloud migration projects that might need other functionalities.

Designing the migration of application systems taking into account multiple criteria. Criteria that include the interest of stakeholders with different roles, the users of the proposed concept —that is, the organisation cloud migrating the application system—, and the granularity of the migrated artifacts. One of the contributions of this dissertation supports the stakeholders undertaking the cloud migration regardless of their role from product owners and managers to software engineers in charge of the development, evolution, or maintenance of the target application system. Although the contribution of this dissertation lets cloud providers model their cloud offerings. They are not the main user target group of the proposed concept, which are the organisations migrating their application systems to cloud environments. They can nevertheless model their own offerings given their additional knowledge of them if they would find incentives to do it. If they were to model their cloud offerings as an additional option for target computing environment, organisations could then more easily tend to select their offerings to migrate their application systems. The cloud migration projects are arguably affected by the users of the cloud-migrated application system and whether they access the application system in a local or remote manner. The proposed cloud migration concept accommodates different levels of abstraction to cope with the software artifacts running in the target application system deployment. Examples of software artifacts at different levels of abstraction include, among others, a legacy proprietary database, a script to run a packaged software component without available source code and the component, source code of some particular functionality, and software components communicating through an interface or not.

Migrating application systems to IaaS, PaaS, and SaaS cloud services with different levels of granularity. The most frequent models of cloud service (in ascending level of abstraction) are the Infrastructure-, Platform-, and Software-as-

[30]cf. Al-Ali et al. (2018b), pp. 45–48; Al Ali et al. (2014), pp. 1–3; Tesgera et al. (2014), pp. 253–253

a-Service models: IaaS, PaaS, and SaaS. A cloud consumer can configure IaaS down to the levels of middleware, operating system, and runtime environments. A SaaS cloud consumer instead does not have that level of control and just uses the offered services and migrate data and computation to them. The cloud service model selected affects the decision making and how to adapt the application system so that it can be migrated. When migrating to cloud services within the IaaS model organisations deal with virtual machines (VM), servers, network, or storage. On the other hand, PaaS providers supply a computing platform on a higher level of abstraction, the migration could leverage the operating system at the level of Platform as a Service, programming language execution environment, database, and web server. This opens up the door to using customised provider-dependent tools. Finally, the SaaS model allows less flexibility and entails agreeing on the legal constraints, the SLA, and the pay model prior to moving the data layer to the cloud premises.

Evaluating the proposed concept. The evaluation of the cloud migration concept and the findings outlined in this dissertation are shown at the end of this dissertation. The evaluation of these findings use developed prototypes and experimentation to evaluate the proposed concept within three scenarios for: migrating to PaaS offerings, the cloud migration of business intelligence, and the move to the cloud of collaborative networks systems.

1.5 Research methodology

This dissertation abides by the design-oriented information systems research memorandum[31]. According to the memorandum, the conceptual design guides the implementation of the decision support concept while taking into account the applicable restrictions and limitations. The dissertation starts with a literature review that focuses on cloud computing, Software Oriented Architecture (SOA), and virtualised environments to then study the migration of application systems to cloud environments. This analysis frames the research of this dissertation to the bodies of knowledge related to application system modelling techniques

[31] cf. Österle et al. (2011), pp. 7–10; Hevner et al. (2004), p. 76; Peffers et al. (2007), pp. 55–59; Jones and Gregor (2007), p. 312–314

and the characterisation of the criteria affecting the decision of how to cloud-enable an application system. Furthermore, this dissertation is centred on how the cloud migration affects the fulfilment of the decision criteria for cloud migration of application systems as well as on the cloud environment selection across different levels of abstraction within the cloud stack: IaaS, PaaS, and SaaS.

This dissertation breaks the theoretical framework down into the set of requirements of the cloud migration DS concept. The evaluation of the proposed concept required developing it into a research prototype as a method to proof its validity and evaluate it. Research in different disciplines helps in building the cloud migration concept; for example, model-driven engineering (MDE) and model-driven architectures (MDAs) as well as multi-criteria decision making. Model-driven approaches help in modelling the target application systems for the cloud migration, the available cloud environment configurations, and the partially cloud-enabled application systems. Decision support systems and multi-criteria decision support systems and methodologies —such as the Analytic Hierarchy Process (AHP)— are investigated as decision support concepts that can be embedded into the research concept and prototype in order to partially automate the cloud migration decision-making process. This requires pinpointing the relevant criteria that drive the cloud migration decision making according to the cloud migration project and the metrics to quantify them. This dissertation uses state-of-the-art approaches to model the target application system and target cloud environments as part of the input to the decision-making assistance that intends to improve on the organisation's criteria to migrate their application systems to cloud environments.

A prototype is built and evaluated to prove the validity of the theoretical work and the cloud migration concept. This implementation finds place in parallel to improvements in the theoretical framework and to initial experimental analyses of the prototypes. The technical implementation uses Java and several related frameworks such as Eclipse RCP, Ecore, and JSF. The research prototype of the decision support copes with the adaptation of application systems to effectively migrate them to multi-cloud architectures and the premises local to the migrating organisation. The prototypical implementation of the proposed concept is used to evaluate the concept in three real-life cloud migration scenarios. These are cloud migration projects in the field of migration to Platform-as-a-Service offerings, the cloud migration of business intelligence systems, and the migration to cloud-based collaborative network systems for service-enhanced products. This last step evaluates the decision support concept and demonstrates the validity of the concept and prototype for realistic scenarios, as stated in the design-oriented memorandum. The evaluation-based analyses of the effects of the cloud migration are used to improve the proposed concept.

1.6 FP7 RELATE research frame of reference

During the first years of this dissertation, the RELATE project[32] offered a research ecosystem around cloud computing with multiple chances to interact within and without its boundaries. The project works as a network offering post-graduate education in how to engineer and provide service-based cloud application systems[33]. The goal of the project is to collaboratively train academic researchers and experts in the area of engineering and provisioning of service-based cloud application systems. Within this context, the research performed for this dissertation included researching and publishing for four months at the King's College London under the supervision of Steffen Zschaler[34]. This section summarises the collaborative research work in the span of the doctoral research of this dissertation and closely relates to an article published by the consortium[35].

The project addresses emerging challenges related to the changes in provisioning and engineering services for cloud computing and mobile or cyber-physical systems that also use cloud computing. It incorporates everything as a service (XaaS) provisioning models by decoupling the access layers and operation of those services[37]. In addition, one of the challenges is overcoming that the different layers in the cloud stack —the Infrastructure, Platform, and Software as a Service layers— lie within different administrative control boundaries, which makes

[32] cf. RELATE ITN (2015), p. n/a, URL see Bibliography
[33] cf. Kounev et al. (2013), pp. 51–54. The European Union in the Marie Curie Actions programme of the European Union funded the RELATE Initial Training Network (ITN) as a multidisciplinary training network of European academic and industrial partners
[34] cf. Juan-Verdejo et al. (2014b), pp. 467–474
[35] cf. Kounev et al. (2013), pp. 51–54
[36] Own illustration based on Kounev et al. (2013), p. 52
[37] cf. Durkee (2010), pp. 1–2

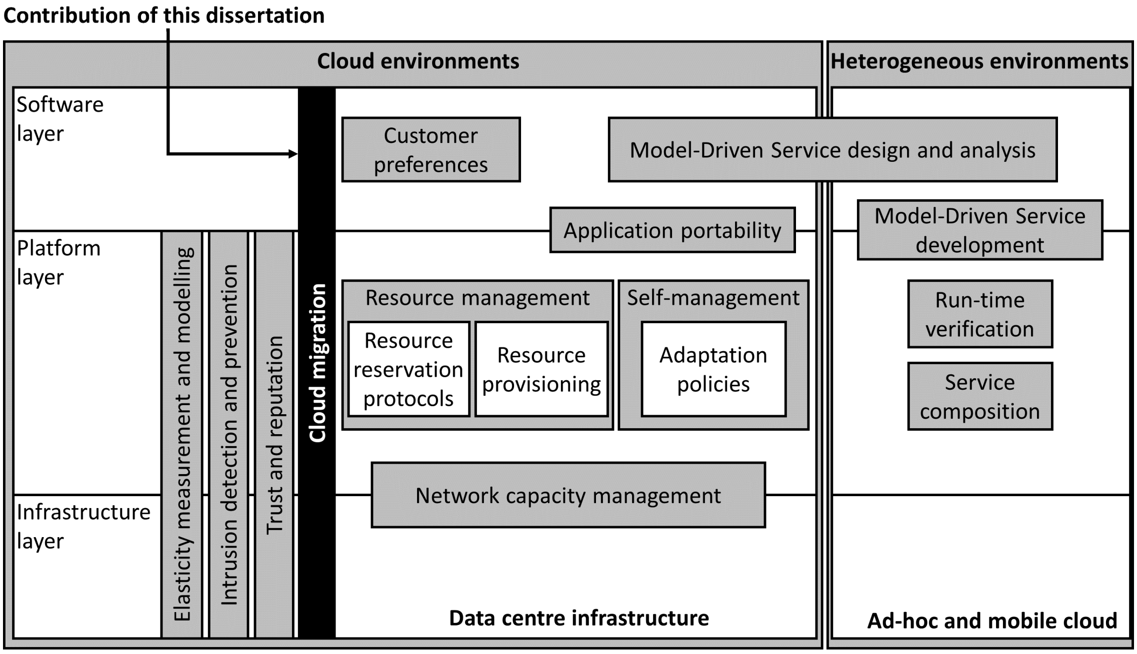

Figure 1.4: The contribution of this dissertation in the context of the RELATE Marie Curie Initial Training Network[36]

it difficult to design and deliver cloud services. The increased complexity of the application systems, including the separation between service and infrastructure providers, combined with the dynamic characteristics of cloud service ecosystems lead to difficulties to meet quality of service and dependability guarantees. Threats and vulnerabilities specific to cloud computing deployments appear due to resource sharing and might compromise security and lead to trust problems. Seven partners —five academic and two industrial partners— across Europe make up the cooperative training network to train twenty-one research fellows. With each research fellow leading an individual research project tackling a particular research topic —including the topic of this dissertation: cloud migration— within the area of service-based cloud application systems including the fields of service engineering, optimisation, performance, trust, and quality management.

Figure 1.4 shows an overview of the project that puts into perspective the contribution of each research fellow to advance the cloud computing state of the art. The contribution of this dissertation, marked in black, contributes to the field of cloud migration across the entire cloud stack. This dissertation relates to and interacts with other individual doctoral and post-doctoral research projects across four categories: model-driven service engineering, run time service management, optimisation and adaptation, and cross-cutting concerns. The latter is the umbrella term containing the research work that this dissertation describes and the work of other on topics such as security, trust, performance modelling, and customer-driven service evaluation.

1.7 Thesis outline

Chapter 1 — Introduction. The research topic of this dissertation is introduced here in order to provide the reader with some understanding of the context, motivation, and high-level aim of this dissertation. Figure 1.5 shows the structure of this dissertation —composed of eight Chapters and the bibliography— with respect to its four goals shown on its right-hand side.

Chapter 2 — Cloud computing. This chapter explains the foundations of the literature and practice of cloud computing and cloud-based application systems. It starts with the standard definition of what cloud computing means[38] and continues to explain the particularities of cloud systems and how the technology emerged. The technologies related to cloud computing are explained here as the foundational blocks that shape what cloud computing is and what makes it different from other computing paradigms. A discussion follows about the options cloud computing offers to deploy and service application systems. Chapter 2 provides the background information so as to identify the challenges and opportunities that cloud computing deployments offer to organisations looking for a computational model for their application systems.

Chapter 3 — Software migration. Chapter 3 explains the state of the art of software migration, the adaptive maintenance activity that aims at complying to new application system requirements[40]. Both software migration and modernisation relate to software maintenance, which aims at adapting software application systems to their changing requirements[41]. Software migration has been a field of its own for a long time due to the continuous appearance of new computing technologies and the need to adapt application systems to them. In addition, this

[38] cf. US-NIST (2009), p. 2
[39] Own illustration
[40] cf. Williams and Carver (2010), pp. 37–38; Jamshidi et al. (2013), pp. 142–157
[41] cf. ISO/IEC (2006), pp. 1–44

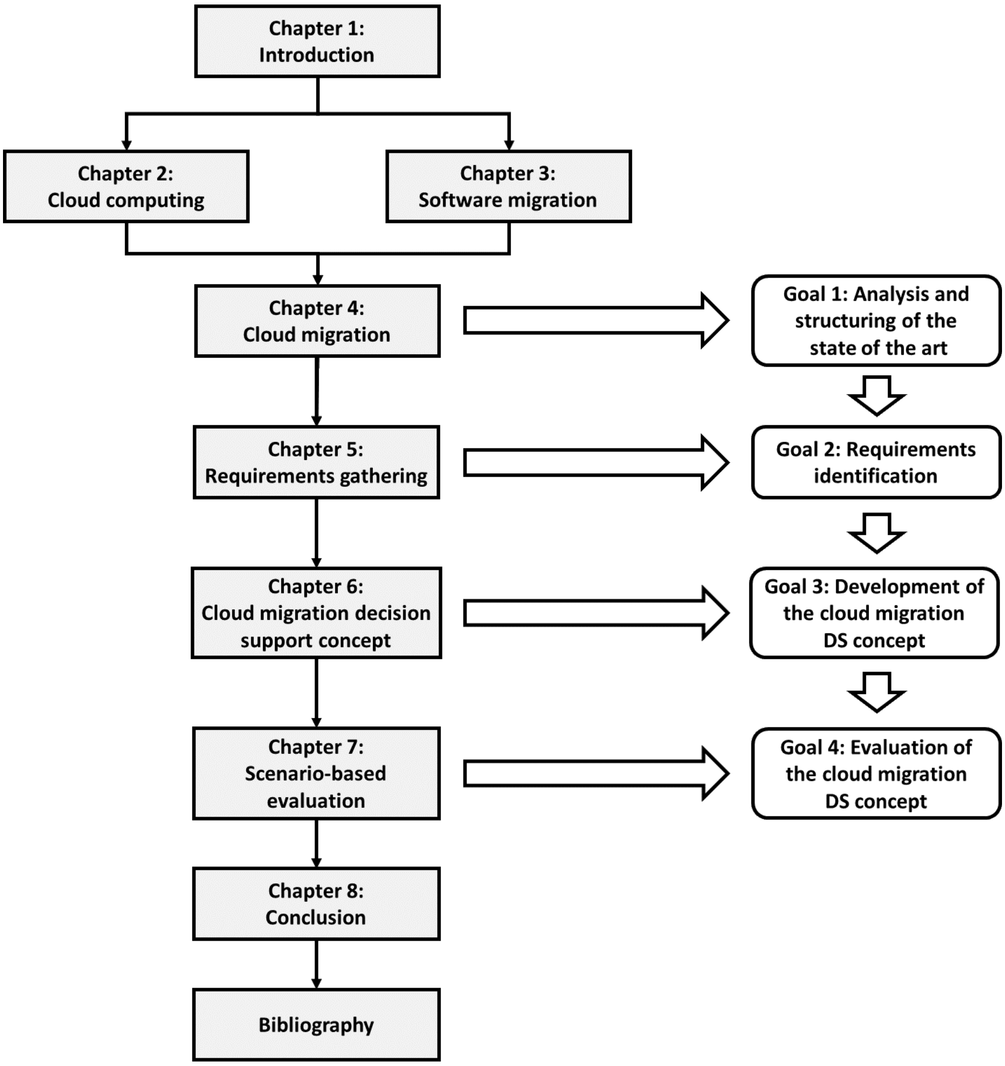

Figure 1.5: Roadmap of this dissertation[39]

chapter clarifies the differences and similarities between software modernisation, migration, and re-engineering.

Chapter 4 — Cloud migration. This dissertation applies the analysis of the state of the art of software migration (Chapter 3) to cloud migration. Chapter 4 delves into the particular challenges and opportunities that the new cloud computation model offers to the migration of application systems. It highlights why cloud migration is a relevant and complex research problem that this dissertation ought to address. It also shows how other researchers cope with the migration of application systems to cloud and multi-cloud architectures. Furthermore, this chapter dives into the cloud migration topic building up from the concepts explained for software modernisation, migration, and re-engineering in general to find the particular research works on cloud-enabling existing application systems and the associated research gaps.

Finally, Section 4.7 describes the research design and the research frame of reference of this dissertation.

Chapter 5 — Requirements gathering. The requirements relevant to the development of the research concept are explained here, right after the research design phase. The requirements gathering phase undertakes a methodological process to find the requirements of the research concept proposed in this dissertation. The research concept will have to fulfil the requirements described here[42]. The requirements come from analysing the foundational research work, secondary research, and existing projects and products on cloud migration in order to identify the needs that the proposed concept should fulfil.

Chapter 6 — Cloud migration decision support concept. It presents the proposed research concept as well as its description and design. Chapter 6 explains the design that addresses the research gap identified by this dissertation from the general to the specific. Based on the analysis of the state of the art to cloud migration, a selection of the approaches detailed in the state of the art will be combined to build the foundations of the proposed concept[43]. This chapter presents the proposed cloud migration decision support concept, which helps organisations cloud migrate their application systems. It describes the conceptual constructs that the proposed concept uses, builds, and combines to create the research design that fulfils the aim of this thesis. The description of the proposed concept includes detailing its architecture and the steps and the workflow it follows. In addition, Chapter 6 explains how to apply the proposed concept to the migration to different service models (IaaS, PaaS, and SaaS) that might offer mechanisms to plan soft reservations or to deploy cloudlets. Finally, this chapter explains how to tailor the proposed concept to the different scenarios of the evaluation.

Chapter 7 — Scenario-based evaluation. Firstly, the chapter discusses the evaluation plan. Next, it details the implementation of the prototype of the proposed concept. This prototype is used to evaluate the research concept by applying it to three real-life scenarios. The intention is to better determine whether the concept

[42]Then, Chapter 6 explains the research concept and Chapter 7 describes its evaluation by implementing a prototype based on it

[43]cf. Wittern et al. (2012), p. 141; Andrikopoulos et al. (2013a), p. 493; Garg et al. (2011), pp. 212–216

can be beneficially applicable to realistic cloud migration projects. Chapter 7 explains the design decisions taken to build the prototype according to the design of the cloud migration concept. Lessons learnt through the development and use of the cloud migration prototype help to iteratively and incrementally derive further design decisions applicable to it. The three real-world cloud migration scenarios of the evaluation emanate from industrial and research collaborations. The core idea is to evaluate what is needed to apply the proposed concept to three cloud migration projects and potentially to other cloud migration scenarios in the future. The three evaluation scenarios pertain to different disciplines: PaaS offerings, Business Intelligence, and IoT-powered collaborative systems.

Chapter 8 — Conclusion. The conclusion looks beyond the direct application of the research concept to the three scenarios described in Chapter 7 by shedding light into the outlook of this dissertation and how to extend it according to new requirements. It discusses how the research concept might be applicable to different research scopes in the future in addition to summarising the dissertation and critically analysing its results.

2. Cloud computing

Cloud computing emerges from the combination of several existing technologies and concepts related to virtualisation, distributed computing, internet technologies, and autonomic computing[44]. Laying the foundations of these technologies and how they relate to cloud computing provides the common ground and overall research framework narrowing the scope of this dissertation. Concepts coming from all these technologies merge and evolve in this computational paradigm with the potential to increase business agility, smooth the execution of business processes, and reduce costs[45]. This dissertation uses the most widely accepted description of what cloud computing is —and what it is not— by researchers and professionals. The definition by the National Institute of Standards and Technology (NIST) applies to the proposed cloud migration concept and the research that shapes this dissertation around these technological changes. The following sections explain the particularities of cloud computing and why organisations can greatly benefit from the assistance of the proposed concept to address the challenges of migrating application systems to cloud environments. NIST describes the cloud model not only in terms of the definition but it also depicts cloud computing as a composition of five essential characteristics, three service models, and four deployment models[46]. This dissertation contribution builds on top of this definition.

[44]cf. Sadeeq et al. (2021), pp. 176–181; cf. Marston et al. (2016), pp. 176–181; Fox et al. (2009), pp. 1–19; Vouk (2004), p. 235

[45]cf. Jonas et al. (2019), pp. 15–19; Jonas et al. (2017), pp. 445–447; Armbrust et al. (2010), pp. 50–51; Buyya et al. (2009), p. 599

[46]cf. US-NIST (2009), p. 2; cf. Miyachi (2018), p. 6–11

2.1 Business and cloud computing

The analysis of the risks and benefits of adopting cloud environments, as the computing environment for an application system, can provide the grounds on which to base the business decision of whether to migrate an application system to cloud environments and how. The *advantages and benefits* of running application systems in cloud environments include increased business agility and cost savings, both capital and operational, through the infrastructure accessible anytime from anywhere and using any kind of device. The new computing environment is usually the core business of the cloud provider, which offers an optimised and secured IT environment to which organisations outsource failover, recovery, back-ups, and redundancy. The additional functionalities and properties make organisations potentially enjoy additional ease of installation and maintenance. Likewise, cloud migration also entail some *risks and threats*. From a security standpoint, the cloud environment becomes a single point of failure and poses some risks to data privacy and security as data move to the cloud environment. Business continuity threats might come from vendor lock-in issues, a potential loss of internet connectivity, or from data availability problems. For example, a law enforcement agency might seize servers at data-hosting provider stored in the same machine, even if unrelated to the services of the affected application system. Disaster recovery is now managed by the cloud provider, which is a double-edged sword as at the same organisations do not need to take this responsibility but they assume risks as they now depend on the ability of thei rcloud provider. An additional question is the extent to which the cloud provider offer extensive levels of control and flexibility to cloud consumers over their data while the provider respects the data retention requirements imposed to them by applicable regulation.

Organisations adopting cloud computing environments to run their application systems have got the potential to increase their business agility. Business agility is an outcome of organisational intelligence. It is the ability of business application systems to adapt their initial stable configuration to rapidly and effectively respond to change. High levels of business agility involve putting products and services in harmony with the changing business environment to fulfil the customer requirements[47]. Service-oriented architectures (SOA), which foster reusing software components to fulfil business requirements, combined with the ability of cloud application systems to scale up and down make up for a perfect mix that can be used to improve business agility[48]. An application system running on cloud environments can use scalability to react to changes in demand in the market. Cloud-based application system can scale up or out to accommodate a market highly demanding the services the application system provides. Likewise, scaling down or in helps reducing incurring costs as a result of over-provisioning the application system. Business agility can also result from combining the use of cloud environments workflows to compose heterogeneous resources and complex processes[49].

Cloud computing can potentially have ripple effects on how organisations internally work and how they interact with their employees and other organisations. The internal business operations and processes of organisations and enterprises might change and alter how they conduct their businesses and how they organise themselves; their organisational structure. The financial implications of adopting cloud computing entail potential for cost savings but might also entail financial burdens due to changes in how the organisation's personnel works or the training they might need. Outsourcing the infrastructure hosting has got consequences for the organisation's personnel as the organisation might not need such a large IT department anymore. As a result, at least this department might oppose the change. Other developers might do it as well if they are reluctant to learn new technologies or ways of working[50].

The decision of adopting cloud environments carries some level of risk with it. Organisations have to be ready to handle these risks if they want to mitigate the different strategic challenges as well as legal risks and cybersecurity hazards. In addition to the technical implications, organisations might have to consider that their business processes might have to change across the entire organisation structure

[47]cf. Alpar and Polyviou (2017), pp. 124–126
[48]cf. Noran and Bernus (2017), pp. 355–358; cf. Hirzalla (2010), pp. 379–380
[49]cf. Ren et al. (2017), pp. 512–514
[50]cf. ElMalah and Nasr (2019), pp. 4120–4124; cf. Kern et al. (2002), pp. 290–292; Cloud computing offers netsourcing whereby multiple customers pay to access supplier-managed business application systems over the Internet or other networks

in order to reduce the impact of adopting those new technologies in their finances, support, service delivery, sales, risk, operations, and human resources[51]. From a technical viewpoint, business processes might have to change for development and testing, storage and back-ups, and networks. Even if these changes in procedures and processes could ideally streamline organisations and help them optimise their daily operations, they also entail the usual risks associated to any change[52].

2.2 The five cloud computing essential characteristics

NIST defines the five essential characteristics, below, related to cloud computing in order to set a common ground for the discussion on what it is cloud computing and what it is not[53]. This five cloud characteristics provide the framework that demarcates the context and assumptions of this dissertation and research. Based on them, the cloud migration DS concept focuses its contribution to software migration in the field of cloud migration.

On-demand self-service have users consume cloud resources without human intervention over the Internet through *broad network access* in a standard manner from any client device. In addition cloud providers perform *resource pooling* to offer them to multiple cloud consumers depending on their demand, which they accommodate with *rapid elasticity*. These pooled elastic resources seem to be infinite. Finally, cloud providers offer their *measured service* and charge their clients per use of their cloud resources[54].

These five cloud characteristics affect the design of the proposed concept as they are fundamental to how to undertake the cloud migration decision. However, it is complex to measure their effects on the decision. Adding to this complexity in the distribution of software components, the research concept proposed in this dissertation is designed to overcome challenges related to privacy and trust by distributing software components to both the premises local to the organisation

[51]cf. Eisa et al. (2020), p. 102–108; cf. Polyviou et al. (2014), p. 5060
[52]cf. Khayer et al. (2020), pp. 12–14; cf. Marston et al. (2011), pp. 176–179
[53]cf. Miyachi (2018), p. 6–11; cf. US-NIST (2009), p. 2
[54]The initial definition of cloud computing: US-NIST (2009), p. 2 and further ones cf. Miyachi (2018), p. 6–11

migrating an application systems and to the cloud environments. The idea is to avoid the reluctance some organisations have, at least at first, to move their data and sensitive computation to cloud environments. Cloud computing is an innovative computational model that emerges as a combination of different technologies whose understanding, study, and research help unveil the possibilities that organisations have in order to better use this technology and migrate their application systems to cloud environments.

2.3 Technologies which evolved into cloud computing

Cloud computing does not appear out of the blue but emerges from four main preceding technologies. This section intends to uncover the differences between each of them because they are fundamental to design the proposed concept[55]. This section compares cloud computing with the four technologies shown in Figure 2.1[56]. They summarise technological advancements at the roots of cloud computing: distributed computing, virtualisation, Internet technologies, and systems management and autonomic computing. The cloud migration of application systems requires specific processes and concepts that cater to such a combination of technologies which does not end up in a simple composition of technologies. Instead, new computational challenges and opportunities appear with the new cloud computational model and not simply as a combination of its components. The cloud migration concept proposed in this dissertation aims at making it easier for organisations to consider all the different technologies, challenges, and opportunities by assisting them to migrate their application systems to cloud environments in a structured and methodological manner.

2.3.1 Distributed computing

Cloud computing naturally evolves from grid computing, which offered computing power as easily as an electric power grid, combined with mechanisms coming

[55]cf. Miyachi (2018), p. 6–11; cf. Zhang et al. (2010), p. 7, 18
[56]cf. Voorsluys et al. (2011), pp. 1–5
[57]Own illustration based on Voorsluys et al. (2011), p. 6

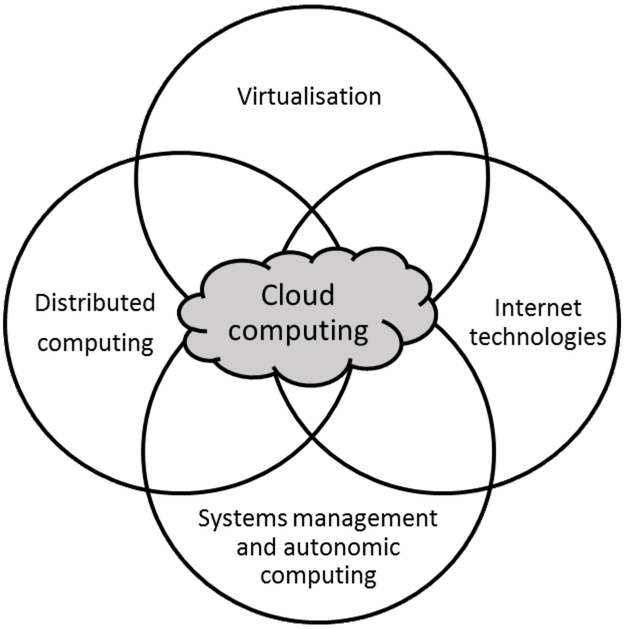

Figure 2.1: Technologies related to cloud computing inspired by Voorsluys in 2011[57]

from application systems management and autonomic computing applied to automatically provision and bill for cloud services without human intervention. Cloud computing represents the intersection of loosely-coupled services that follow an architectural variant of the service-oriented architecture (SOA) and other distributed computing concepts[58].

Grid computing

Grid computing uses distributed systems to reach a common goal by running workloads that are not interactive and that use a large number of files[59]. Its application to real-world problems started by leveraging unused processor time to process large amount of data for scientific application systems. It continued with climate modelling, drug design, or protein analysis. Grid computing appeared before the cloud computing model, which might overlap today due to having the latter evolved out of grid computing[60]. Cloud computing environments arguably rely on grid computing as their backbone and infrastructure support but they both differ in that cloud computing incorporates a business model and economic concerns[61].

[58] Fog computing, cloudlets, and edge computing, in addition to software containers. cf. Cao et al. (2020), pp. 85718–85722; cf. Carr (2008), pp. 67–68; Buyya et al. (2010a), p. 11

[59] cf. Foster and Kesselman (1999), p. 12; Anderson et al. (2021), p. n/a, URL see Bibliography

[60] cf. Foster et al. (2008), pp. 1–2

[61] cf. Susanto et al. (2019), pp. 322–328; cf. Stanoevska-Slabeva et al. (2010), pp. 12–21; Buyya et al. (2008), p. 10; Pérez et al. (2010), pp. 1335—1337

Cloud and grid computing employ distributed technologies to reach application-level goals. However, cloud computing uses virtualisation technologies at multiple levels to share resources and dynamically provision them. Similarly, grid computing delivers storage and computing resources, whereas, cloud computing provides resources and services in a virtualised environment on a higher abstraction level[62]. It is the abstraction of virtualisation that solves typical problems related to grid computing such as lack of flexibility and allows for addressing Quality of Service and reliability problems[63].

Utility computing

Companies used to offer common data processing tasks —such as invoicing, accounting, or payroll automation— by operating time-shared mainframes as utilities. They used to serve several application systems to often operate at close to 100% of their capacity[64]. In fact, mainframes had to operate at very high utilisation rates to justify their high cost by efficiently using them[65]. With the advent of fast and cheap microprocessors and IT data centres, computation moved from mainframes to commodity servers. This meant the disappearance of mainframes and led to the isolation of workloads running within dedicated servers due to incompatibilities between software stacks and operating systems[66]. In addition, the inefficiency of computer networks meant that the IT infrastructure had to be in the proximity to the end consumer.

These particularities arguably prevented utility computing from really taking off[67]. Utility computing works with differing utility values according to the computing job. Utility varies according to quality-of-service constraints[68] in valuations that vary (or not) with time. Valuation refers to the amount of money users would be willing to pay to a service provider to satisfy their demands. Utility computing can therefore capture more subtle user requirements —in addition to throughput, waiting time, or slowdown— and provides users with incentives to be flexible when

[62] cf. Robertazzi (2012), p. 65, 68; Alcala Alvarez et al. (2012), p. 11
[63] cf. Rings et al. (2009), p. 375; Stantchev and Schröpfer (2009), p. 25
[64] cf. Susanto et al. (2019), pp. 322–328; cf. Buyya et al. (2010a), p. 22
[65] cf. Carr (2008), pp. 67–68
[66] cf. Foster and Tuecke (2005), p. 26
[67] cf. Foster and Tuecke (2005), p. 29
[68] Deadline, importance, or satisfaction.

describing their resource requirements and job deadlines[69]. Similarly, service providers are encouraged to maximise their own utility to economically profit from offering computation time to more clients or from highly prioritising high-yield user jobs. These incentives for consumers and providers help accommodating spikes in resources demand by fairly and equitably distributing resources to prevent adversarial user behaviour. These kind of shared systems work as a marketplace whose users compete for computing resources based on the perceived utility or value of their jobs[70].

The usual cloud computing business model implements parts of the utility computing model. Cloud infrastructures provide resources on-demand and charge their consumers according to how they consume them[71]. These charges are similar to how utility computing bills for hardware and software usage rather than according to a flat-rate pricing model. With on-demand resource provisioning and utility-based pricing, cloud service providers can truly maximise resource utilisation and minimise their operating costs[72].

2.3.2 Virtualisation

Cloud computing services usually run on large-scale data centres composed of thousands of computers. Such data centres serve lots of users and host many heterogeneous application systems. Hardware virtualisation is a key technology to cope with many of the operational challenges that arise when building and maintaining data centres. Additionally, it is used to ensure the realisation of the cloud computing model. Virtualisation supplies the abstractions used to pool resources to increase resource sharing and utilisation. Physical machines usually run several virtual machines, which supply many users. Virtualisation abstracts the details of both the hardware and the operating system and offers virtualised resources (such as processors, memory, networks, and input/output devices) for high-level application systems[73].

[69]cf. Susanto et al. (2019), pp. 322–328; cf. Buyya et al. (2010a), p. 53
[70]cf. Buyya et al. (2010a), p. 11
[71]cf. Broberg et al. (2008), p. 255
[72]cf. Parkhill (1966), p. 15; Zhang et al. (2010), p. 18
[73]cf. Bhardwaj and Krishna (2021), p. 8591–8597; cf. Goldberg (1974), p. 34

Virtual resources run in clusters of servers and are dynamically assigned or reassigned to application systems on-demand[74]. The study of virtualisation came back to the research spotlight in part due to the appearance of inexpensive and powerful servers and client machines[75]. Virtualisation facilitates creating virtual resources: operating systems, servers, or storage devices. The focus of this dissertation lies on the virtualisation at the hardware virtualisation within the context of the cloud migration to IaaS, PaaS, or SaaS[76].

The study of virtualisation concepts is at the core of cloud migration and of software containerisation or operating-system-level virtualisation. Containerisation uses OS kernel features that allow for multiple isolated user-space instances, also called containers. Architectures that use containerisation with, for example, Docker[77] and Kubernetes[78] arguably benefit from less overhead. These architectures use the normal system call instead of emulation or intermediate virtual machines. In addition to containers, some other similar technological solutions call them partitions, virtualisation engines, or jails depending on the implementation. These kind of systems suffer from being less flexible than other virtualisation approaches as they cannot host different operating systems and it is difficult to roll back full snapshots of the system. However, containers usually improve storage efficiency, hardware independence, security, and resource management features.

Multi-tenancy is used in virtualisation to allow multiple tenants and users to share pooled resources (storage space, processing power, memory, and network bandwidth). One of the keys to multi-tenancy on the application level includes delivering a higher level of customisation based on the metadata specific to each tenant. This approach leads to addressing the isolation on the data layer depending on the tenant. Architectures map tenant structures to database schemas so as to make it possible to customise data models for each specific tenant. Three basic approaches enable multi-tenancy at the data layer. Namely, one database per

[74] cf. Xing and Zhan (2012), p. 312

[75] cf. Hauck et al. (2010), p. 13 after the initial focus on virtualisation in the 1970s Goldberg (1974), pp. 34–40

[76] Virtualisation solutions work at full, partial virtualisation or para-virtualisation and include different classes of virtualisation such as hardware, operating-system-level, application, desktop, service, storage, memory, network, or data virtualisation

[77] Docker is the de facto standard for software containerisation

[78] Kubernetes usually orchestrates containers for automating deployment, scaling, and management

tenant, one schema per tenant, or a shared database[79]. The first approach lets each tenant store data in a separate physical database. One separate local unit, or schema, within a single physical database maps each tenant to a separate logical unit.

Finally, with a shared database, the same physical database and schema stores all tenants. Nevertheless, data experts design the database use primary keys to separate the information of each tenant within the same physical tables[80].

Figure 2.2: The continuum between isolated and shared data based on Chong in 2006[81]

The first two approaches —one database per tenant and one schema per tenant— are better at isolating each tenant and increase the flexibility of the data model. These approaches keep the data of each tenant logically or even physically isolated. With usual database techniques data experts can customise the data model according to the needs of each tenant. The shared-database approach decreases the hosting costs because it needs less database server resources. As a downside, it is harder to implement flexible data models tailored to each tenant but there are solutions to address this issue. One idea is to add a pre-set number of data fields for tenants to store custom information[82]. An alternative approach uses linked separate key-value tables. Finally, a third approach entails using huge generic data tables with metadata tables to guarantee type-safety and pivot tables to run optimised queries[83].

The distinction between shared and isolated data moves along a continuum with many options between these two extremes[84] as Figure 2.2 shows. With multi-tenancy, data and configuration are partitioned to offer each client organization its customised virtual application. Application systems running in cloud environments

[79]cf. Chong and Carraro (2006), p. 9
[80]cf. Chong and Carraro (2006), p. 10
[81]Own illustration based on Chong and Carraro (2006), p. 10
[82]cf. Bhardwaj and Krishna (2021), p. 8591–8597; cf. Chong and Carraro (2006), p. 10
[83]cf. McKinnon (2008), p. 5
[84]cf. Chong and Carraro (2006), p. 9

seamlessly share their resources between them as multiple concurrent tenants are running in the system[85]. Some patterns can be used to implement tenant isolation in which every tenant perceives the application (and its security, performance, availability, and administration) as if they were the sole tenant in the system.

2.3.3 Internet technologies

Service-Oriented Architectures, Web Services, Web 2.0, and Mashups are Internet technologies composing the cloud computing model. The progressive standardization of these technologies in general and web services (WS) in particular has fostered its integration in the cloud computing model and facilitated its implementation. With web services, developers can compose services from different providers to supply cloud application systems. According to the World Wide Web Consortium (W3C), a web service is a software system which supports interoperable machine-to-machine interaction over a network. Web services provide functionality in a standardised manner and enhance software integration[86]. Web services expose an XML-based machine-processable interface or Web Services Description Language (WSDL).

Applications can offer internal information and functionality to other application systems running on different messaging product platforms over the internet through them[87]. REST-compliant or arbitrary web services let heterogeneous systems interact with them typically over the HTTP protocol and by using XML serialisation and other web standards[88]. Since the appearance of internet technologies, the web services software stack has been increasingly specified and standardised. This processed has rendered it easier to operate services[89] that communicate via packaged messages while fulfilling non-functional requirements of security and quality of service provision.

The implementation of the service-oriented architecture (SOA) benefits from web services standardisation and the use of common mechanisms to provide services.

[85]cf. Fehling et al. (2010), p. 259
[86]cf. Bellman et al. (2021), pp. 32–38; cf. Papazoglou and Van Den Heuvel (2007), p. 389
[87]cf. Bouguettaya et al. (2017), pp. 64–66; Papazoglou and Van Den Heuvel (2007), p. 415
[88]cf. Bouguettaya et al. (2017), pp. 70–72; Haas and Brown (2004), p. n/a, URL see Bibliography
[89]Operations to describe, publish, discover, orchestrate, and compose services

Service-oriented architectures uses services to package software sesources to be used in distributed computing. These services are usually loosely coupled an while they follow standards, they tend to remain protocol-independent. Services are organised as standard business functionalities that are well defined and that run in a self-contained manner. Services are usually described in a standard definition language and do not rely neither on the state of other services nor in their context or on how they publish their interfaces. State-of-the-art web services let other services and end-user application systems access them on-demand in a uniform manner. With SOA, services are bundled together to allow application systems implement their needed functionalities[90].

Service provision moved from the enterprise web to the consumer realm with the Web 2.0. In the consumer web, it becomes easier to compose services to develop web application systems that combine information and services in service mashups, which act as the basic building blocks of more sophisticated compositions of services[91]. Within the Software- and Platform-as-a-Service domains, developers may compose services from different providers to build cloud application systems. Cloud application systems often need similar business functionalities such as authentication, e-mail, calendar, or payroll management that can be reused and combined. Many of these building blocks are available in public marketplaces[92].

2.3.4 Systems management and autonomic computing

Autonomic computing[93] let distributed computing systems manage themselves by reacting to its internal and external observations, which do not require human intervention. Autonomic computing systems rely on monitoring sensor data to let their adaptation engine actuate on the effectors with as little human intervention as possible.

IBM created a reference model to conceptually describe the autonomic control loop: the MAPE-K[94] loop. It addresses the four autonomic systems' properties

[90] cf. Bouguettaya et al. (2017), pp. 64–70; Papazoglou and Van Den Heuvel (2007), pp. 390–391
[91] cf. Bellman et al. (2021), pp. 32–38; cf. Benslimane et al. (2008), p. 13; Blau et al. (2008), p. 19
[92] cf. Zhou et al. (2017), pp. 720–722; Buyya et al. (2010a), p. 35
[93] The term autonomic computing was coined by IBM: cf. Stamou et al. (2019), pp. 3281–3287; cf. Horn (2001), pp. 21–30; IBM (2006), p. 10
[94] MAPE-K stands for Monitor, Analyse, Plan, Execute, and Knowledge

identified by IBM, that is, self-configuration, self-optimisation, self-healing, and self-protection[95].

Cloud providers use large data centres to build their infrastructures and usually need to manage these data centres efficiently to make them economically profitable. Autonomic computing comes in handy here to perform data centre management automation. Cloud providers use autonomic computing to minimise human intervention for the management of service levels of application systems running in their infrastructures. In addition, they can automate the data centre capacity management and the virtual machine provisioning. Autonomic computing also applies to proactive disaster recovery in addition to allowing for on-demand provision of cloud resources in a faster and less error-prone manner[96].

2.4 Distinctive characteristics of cloud computing

Apart from the technologies cloud computing originates from, this dissertation points out the main representative features of cloud computing that differentiates it from previous computing models. Highlighting these boundaries narrows the scope of the proposed cloud migration concept and helps clearly identify to which conditions and cases the research findings and contributions of this dissertation apply. The following are the seven most representative differentiating features of cloud computing:

Virtualisation: virtualisation abstracts the details of physical hardware. Thus, high-level application systems can be provided with virtualised resources in order to let them perform their tasks while letting cloud providers increase the utilisation of their resources[97].

Availability: cloud providers guarantee via their SLAs uptime percentages ranging from 99.5 to 99.99%. In fact, availability is ranked as one of the top-ranked issues related to the cloud computing[98]. For example, each Amazon EC2 region should, according to its Service-Level Agreement, be 99.95% of the time available[99].

[95] cf. Kephart and Chess (2003), p. 41; Ganek and Corbi (2003), p. 18; Huebscher and McCann (2008), p. 7
[96] cf. Stamou et al. (2019), pp. 3281–3287; cf. IBM (2006), p. 10; Lowe (2011), p. 23
[97] See Section 2.3 on virtualisation for more details
[98] cf. Sadeeq et al. (2021), pp. 176–181; cf. IDC (2010), p. 1
[99] cf. Amazon.com, Inc. (2021a), p. n/a, URL see Bibliography

Broad network access: mixed kinds of thin or thick clients systems or devices (including mobile phones, tablets, laptops, or workstations) can access the resources of cloud providers by using standard mechanisms[100].

Elasticity: contrary to traditional over- and under-provisioning, an elastic system provides the means to flexible and rapid provisioning and de-provisioning of resources on the fly. It represents a core capability of cloud computing to allow an application system to have the necessary resources always available and tightly aligned to the actual demands of the application systems[101]. Figure 2.3 provides a view of elasticity that this cloud migration concept uses including the soft reservations approach.

Pay as you go: one of the business fundamentals of cloud computing is to move from capital to operational expenditure[102]. Cloud consumers do not buy IT resources and hire IT personnel but they *rent* them. Cloud providers today offer implementations of this model that enable cloud consumers to avoid the risks of over-provisioning (underutilisation) and under-provisioning (saturation) their resources[103].

Cloud storage: in order to support the needs of large highly fault tolerant datasets with extreme scalability, cloud environments use logical pools to make data accessible and available with properly running distributed servers running on various locations. The cloud storage infrastructure manages how and where the data is stored, secured, and even backed up[104]. Cloud storage consumers buy or lease storage to persist application, end-user, or organisation data. Cloud application systems sometimes work with large amounts of data which damages the performance of the traditional relational database for on-line storage; issues that NoSQL and distributed memory database technologies try to simplify or eliminate.

Multi-tenancy: multiple client organisations or tenants work with a single instance of the software running on a server. They contrast the architectures with separate

[100] cf. US-NIST (2009), p. 3
[101] cf. Islam et al. (2012), pp. 85–90
[102] cf. Alpar and Polyviou (2017), pp. 124–126; cf. Leymann and Fritsch (2009), p. 3
[103] cf. Pires Barbosa and Schwertner Charão (2012), p. 417
[104] cf. Stonebraker (2010), pp. 10–11
[105] Own illustration offering a view of elasticity based on Herbst et al. (2013), p. 4 and used to include this decision-making aspect affecting the cloud migration of software systems (see Section 4.6 in Chapter 4) and into the soft reservations approach

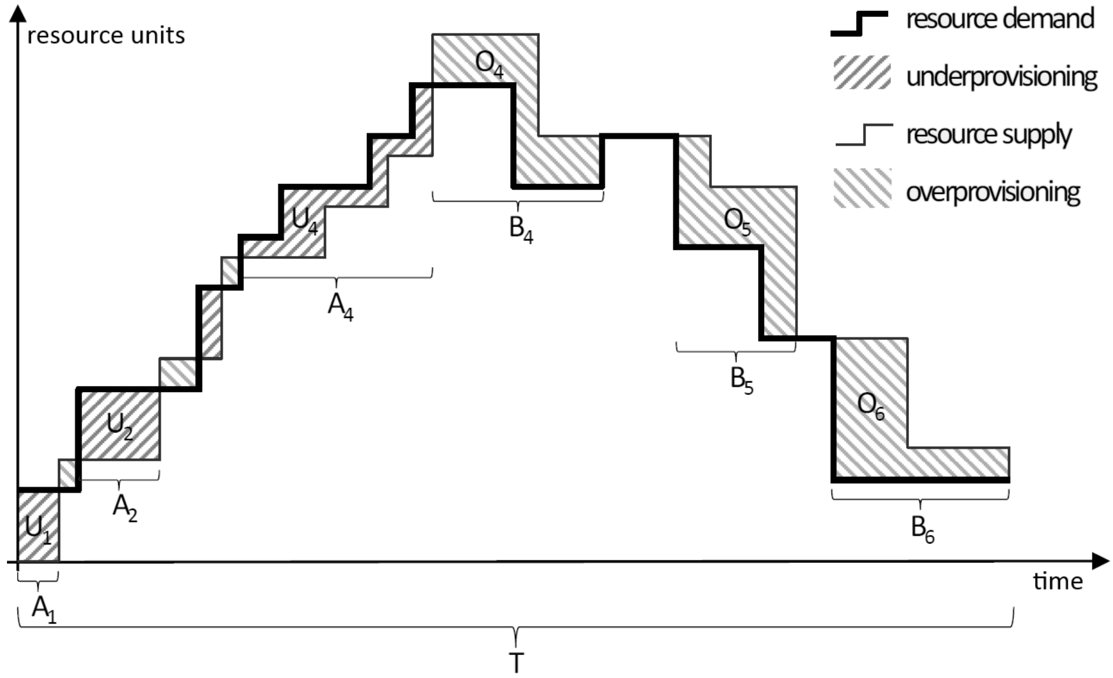

Figure 2.3: Elasticity with the hard and soft reservations approach[105]

software instances or hardware systems that operate on behalf of those different client organisations[106].

2.5 Service models

The differences across the cloud stack affect the decision-making process of migrating application systems to cloud environments and, hence, the design of the cloud migration concept. Each cloud service works with different constructs with varying levels of customisation and degrees of freedom. This section describes the cloud service models in descending level of abstraction. The top level offers the higher level of abstraction at the Software-as-a-Service level, which limits the cloud consumer's usage freedom and allows for less degree of customisation. The service models abstract from the details of the lower models with service providers using services of underlying layers to compose a new one in a higher layer. Figure 2.4[107] shows a comprehensive example of the three standard service models[108] including the Infrastructure-, Platform-, and Software-as-a-Service models. In addition, this dissertation incorporates the notion of the XaaS or Everything-as-

[106]See Section 2.3 on virtualisation for more details about multi-tenancy
[107]cf. Sadeeq et al. (2021), pp. 176–181; cf. Liu et al. (2011), p. 12
[108]Standard service models according to NIST: cf. US-NIST (2009), pp. 1–3

a-Service model in the study of how cloud consumers use services[109]. The proposed concept takes into account the different service models that organisations potentially use to cloudify application systems to use their different advantages according to the cloudification project.

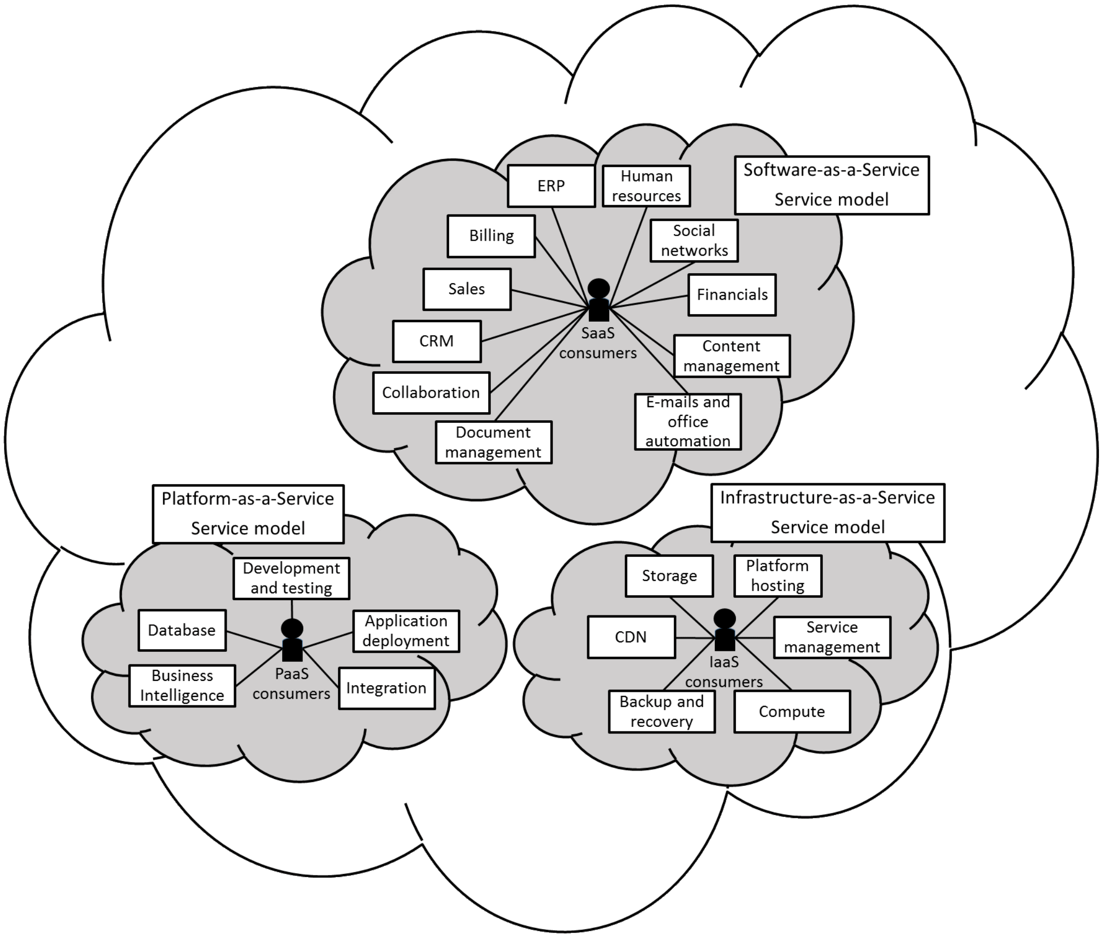

Figure 2.4: Example services available to a cloud consumer based on NIST[110]

2.5.1 Software as a Service (SaaS)

Cloud consumers at the SaaS level use application systems that run on cloud infrastructures. That is, the consumer can usually tailor the application by changing some configuration settings that only affect that specific user but the underlying cloud infrastructure cannot be controlled. The provider of SaaS manages and controls the application systems and the infrastructure to a great extent, while the cloud consumers have limited administrative control of the application systems. Consumers do not manage or govern the resource abstracted from the two lower

[109]cf. Juan-Verdejo and Surajbali (2016), pp. 11–23
[110]Own illustration based on Liu et al. (2011), p. 6

levels. This typically means relinquishing control on the operating systems and the servers running on them in addition to not completely governing the network. It can even entail the inability to control individual application or storage capabilities[111]. The cloud provider configures, deploys, updates, and maintains the operation of the application systems on a cloud infrastructure according to the promised Service-Level Agreement[112].

2.5.2 Platform as a Service (PaaS)

The PaaS consumer can control application systems the deployed to the platform in addition to some PaaS configuration settings. At this level, consumers deploy application systems developed for or by them tailored to the programming environment of the PaaS provider. Providers usually offer some development and software tools in addition to support of some libraries and programming languages. In addition, PaaS providers increasingly offer value-added services to differentiate from their competition.

2.5.3 Infrastructure as a Service (IaaS)

Consumers of IaaS cloud environments have additional degrees of freedom to customise the usage of networking, processing, and storage resources. They can exert control over the infrastructure at the level of operating systems and storage as well as over the application systems they deploy. The level of control of networking components might be a bit more limited.

2.5.4 Everything as a Service (XaaS)

Researchers and practitioners have been coining a myriad of new names and acronyms for service models beyond the widely accepted service delivery models described above[113]. Given the immaturity of these terms, and the fact that the community did not standardise a consistent definition of them; this dissertation does not delve much into their details. Some of them are synonyms such as Applications

[111] cf. US-NIST (2009), p. 3
[112] cf. Liu et al. (2011), p. 12
[113] cf. US-NIST (2009), p. 3

as a Service which refers to the same concept as SaaS; likewise Hardware as a Service refers to IaaS[114]. Given the relative novelty of the concepts and the misunderstanding behind them, some researchers and businesses consider ecosystems with everything as a Service (XaaS)[115] and coin things like Storage, Security, Data, Information, Human, Database, Tests, Monitoring, Network, APIs (Application Programming Interfaces), Backend, or Mobile Backend as a service[116]. Although some —e.g. Backend as a Service— seem more mature and founded than others, accepted definitions for them still vary widely and therefore go beyond the scope of this dissertation.

2.6 Deployment models

Cloud computing stems from public computing utilities but incorporates other deployments to different physical locations following various distribution models. NIST standardises four deployment models independently of their service models[117]. This section clarifies all of them as they play an important role into how the concept distributes software components according to the privacy and security requirements of the cloud migration project. The concept also considers the levels of trust and privacy each of these deployment models because they greatly influence how the proposed concept will have to assess cloud-enabled architectures. These architecture include the components placement and selected alternative to migrate the system to cloud environments.

The cloud migration concept includes into the decision-making process whether to use the local premises of the organisation and which of the following cloud deployment to adopt.

Private cloud: may be operated by a third party in addition to the organisation which enjoys exclusive use of the infrastructure. Moreover, the organisation and the third party might have to share ownership and management responsibilities and the application system might be deployed both on the premises of the organisation and on cloud infrastructures.

[114] cf. Xu (2012), p. 86
[115] cf. Duan (2012), pp. 162–163
[116] cf. Xu (2012), p. 75; Lenk et al. (2009), p. 23; Juan-Verdejo and Surajbali (2016), pp. 11–23
[117] cf. Miyachi (2018), p. 6–11; cf. US-NIST (2009), p. 2

Community cloud: has organisations with similar policies and compliance requirements, as well as security constraints, use this deployment models as a community of consumers.

Public cloud: is provisioned for open use by the general public, it operates on the cloud provider infrastructures.

Hybrid cloud: bounds together several of the previous models which adopt proprietary or standardised technology for the portability of data and applications.

The proposed concept work assists organisations to migrate to cloud ecosystems according to these four deployment models. In addition to considering the premises local to the organisations migrating their application systems to cloud environments: partial migration. The concept also takes into account both the private and public cloud deployment models.

2.7 Cloud standards and related organisations

Multiple standardisation bodies exist within the field of cloud computing to bring together a broad range of individuals and organisations —from technical and geographical standpoints— to develop standards through collaboration and foster their adoption. Standardisation promotes innovation through maximising compatibility, interoperability, safety, repeatability, and quality. Likewise, standards create and expand international markets through commoditising formerly custom processes and solve a coordination problem whereby all involved parties achieve mutual gains through making mutually consistent decisions. In addition to top-down coordinated action, some standards arise organically in a bottom-up manner like spontaneous standardisation activities producing de facto standards. Such is the case of Amazon appearing as a sort of standard for the cloud computing market.

The list of standardisation bodies is huge but for matters of clarity and space, this section lists the most important standards to address some of the issues when cloud migrating an application system related to portability, interoperability, vendor lock-in, inter-cloud architectures, and data security, ownership, and privacy[118] .

[118] The Distributed Management Task Force (DMTF), the Global Inter-Cloud Technology Forum (GICTF), the European Telecommunications Standards Institute (ETSI), the Cloud Standards

Some cloud standards address interoperability and portability in addition to vendor lock-in problems. Two different concepts that can work together and help each other but do not necessarily depend on each other. Interoperability describes the ability of different systems to work, preferably seamlessly, together. Portability refers to the ability of a system to transfer its data, computing, or networking from one machine or system to another one. The difference between the two concepts lies in the idea of interoperating or cooperating systems. The following describes cloud standards and considerations for the three most important service models (IaaS, PaaS, or SaaS).

Infrastructure as a Service. The cost of porting an application systems from an IaaS environment to a new one can be alleviated by using cloud standards. Standards promote the avoidance of vendor lock-in at this level as consumers can more freely move their applications and data to a different platform. IaaS cloud consumers usually port their software runtime environments including configurations and APIs, typically at the VM level. The Distributed Management Task Force (DMTF) proposes the fairly standard Open Virtualisation Format (OVF) as a format for open packaging and distribution outlining how to deploy, manage, and run a virtual machine[119].

Platform as a Service. PaaS offerings adopt standards to enhance application and data portability. Multiple standards appeared in the years around 2013 with two of them being kept more up to date with updates in 2019 and 2020: CAMP and TOSCA. Some of the more prominent PaaS standards include the Open Cloud Computing Interface (OCCI), the aforementioned Topology and Orchestration Specification for Cloud Applications (TOSCA), and the Cloud Infrastructure Management Interface (CIMI)[120]. Additionally, PaaS standards such as the OASIS Cloud Application Management for Platforms (CAMP)[121] mentioned above help

Customer Council, the ISO/IEC JTC 1, the International Telecommunications Union (ITU), the National Institute of Standards and Technology (NIST), the Open Grid Forum (OGF), the Object Management Group (OMG), the Open Cloud Consortium (OCC), the Organization for the Advancement of Structured Information Standards (OASIS), the Storage Networking Industry Association (SNIA), the Open Group, the Association for Retail Technology Standards (ARTS), the TM Forum, the OpenCloud Connect

[119] cf. Alpar and Polyviou (2017), pp. 124–126; cf. Crosby et al. (2010), pp. 9–11

[120] cf. Metsch et al. (2010), p. 3; Binz et al. (2013), pp. 692–695; Cloud Management Initiative (2021), p. n/a

[121] cf. OASIS (2014), pp. 29–39, URL see Bibliography

avoiding vendor lock-in issues. Multiple projects target these portability and interoperability issues and have as a result brought about some cloud interoperability standards such as CAMP for PaaS standardisation and TOSCA for the cloud topology. CAMP is a resource model describing an application in terms of its artifacts and components as well as the services to use or execute them. CAMP allows for specifying how artifacts and services work together in addition to configuration and metadata at the platform level. Additional runtime information for PaaS resources can be described with this standard and it prescribes using a RESTful API protocol to manipulate these resources. Additionally to the most known TOSCA and CAMP, the European Telecommunications Standards Institute (ETSI) has also tried to standardise several aspects related to cloud computing including service provision with TC CLOUD and the Cloud Standards Coordination(CSC)[122]

Software as a Service: Finally, within the SaaS level, data portability enables using data components across different application systems at the same cloud provider or across different cloud service providers. In those cases, SaaS consumers might change to another Software-as-a-Service provider to reduce their costs or increase their quality after repetitive SLA violations, performance issues, or business stability problems affecting their provider.

[122]cf. Miyachi (2018), p. 6–11; cf. Oberle and Fisher (2010), pp. 105–108; ETSI (2016), pp. 10–11.

3. Software migration

This research work tackles a specific case of software migration and, therefore, this chapter describes the state of the art on software modernisation, migration, and re-engineering. This chapter documents the analysis of the existing research approaches and methodologies so as to inform the design of the cloud migration concept based on the state of the art of cloud migration in addition to the practice of cloud migration. The proposed concept will intend to support the decision of how to migrate to cloud environments to modernise the existing software. The adaptation of software application systems to a different computation environment —or software migration— is not a new research field but its application to cloud migration is a new field.

Section 3.1 explains the concept of software modernisation including software maintenance and evolution because software modernisation relates to the software maintenance as it adapts a software system to evolving requirements. The proposed cloud migration concept is informed by the software re-engineering foundational works to modernise software that Section 3.2 describes together with software migration concepts as Section 3.3 does.

3.1 Software modernisation

The software maintenance and evolution fields relate to software modernisation as they deal with adapting a software system to a new computing environment[123]. This

[123]cf. Mall (2018), pp. 544–553; Boehm (1976), p. 1226

section describes the software modernisation research area relevant in the context of the proposed cloud migration concept that Chapter 6 describes. Then, the following Section 3.2 delves into software re-engineering or developing software to keep the functionality of an application system while improving its software quality. Finally, Section 3.3 explains software migration in the context of re-engineering and software modernisation efforts.

Software modernisation focuses on adapting software to a different computing environment that evolved in line with technological and conceptual advances[124], which is very related to the migration of legacy systems to a different computing world[125]. Organisations usually modernise their legacy systems in projects that takes them several years to complete due to their complexity and the critical character of most legacy systems[126]. Legacy systems usually remain so because they need to be always operational which calls for avoiding big-bang strategies but for step-by-step redeployments instead so that the decision of how to migrate the system ensures its correct operation[127].

Within the context of software modernisation, organisations might decide how to modernise their systems based on information whose certainty, completeness, and availability is often limited[128]. Hence, business organisations make software modernisation decisions within their organisational context based on rationality bounded by the information they posses, the cognitive limitations of their minds, and the finite amount of time they have at their disposal. To accommodate these limitations in human decision making, the cloud migration concept might incorporate the conflicting decision criteria in its core for decision support based on the decision criteria that drive organisation's decision on how to migrate their systems to cloud environments, criteria that organisations can define or reuse. Organisations or migration experts build a hierarchy of criteria including their inter-

[124] cf. Anthony and Cantu Jr. (2018), pp. 110–112; Brooks (1975), pp. 115–116
[125] cf. Mall (2018), pp. 546–548; Haziza et al. (1992), p. 18
[126] Organisations undertake the software modernisation during the evolution and servicing stages as they study, expose and facilitate refactoring, redesigning, and redeploying the software system's architecture with the aim of fulfilling critical business requirements while minimising the risks on a cost- and time-efficient manner. Organisations abandon the software modernisation during the phase-out stage to then finally replace the software system during the close-down stage; cf. Qureshi et al. (2017), pp. 725–728; Ulrich (2004), pp. 2–3
[127] cf. Rashid and Lo (2017), pp. 36–38; Yau et al. (1988), p. 1144; Hale et al. (1990), p. 118
[128] cf. Dick et al. (2017), pp. 153–156; Pigoski and Nelson (1994), p. 392; Simon (1955), p. 118

dependencies to reflect the conflicting criteria and delimit how they are willing to trade some criteria off with others.

3.1.1 Software maintenance

Software maintenance is a software engineering phase that includes any adaptation of an already implemented, tested, and deployed software system[129]. According to the IEEE Standard 1219-1998, software maintenance activities include modifications to a software solution after delivering it to a client to improve it in some dimension (such as performance), correct any malfunction, or adapt the application system to a computing environment that has changed[130]. According to this definition, the cloud migration that that the proposed concept will support might relate to the concept of software maintenance as it has to adapt the target application system to the new computing environment —the cloud-based infrastructure. The proposed concept could fall into the adaptive maintenance to adapt the target the application system to the changed environment while, at the same time, showing similarities with the perfective maintenance as it improves quality attributes such as performance and business agility.

Software migration challenges relate to software maintenance and to its adaptive activities. Adaptive maintenance modifies a delivered software product to keep it usable in a changed or changing environment. In that sense, the software migration is a special kind of adaptive maintenance activity to comply to new application system requirements[131] The International Organisation for Standardization (ISO) standardised in 2006 the software maintenance process and suggested six phases, as Figure 3.1 shows, involved in the software maintenance process and the

[129] cf. Mall (2018), pp. 544–553; Boehm (1976), p. 1231; Lehman (1969), p. 31

[130] The standard distinguishes four types of software maintenance activities. Adaptive when an organisation modifies the software system after delivery to keep it usable in a changed or changing environment. Corrective to fix discovered faults after delivery. Perfective software maintenance activities focus on maintainability, performance, or any other quality attributes. Preventive activities in turn intend to detect faulty behaviour after software delivery to correct remaining bugs before any failures appear; cf. De Lemos et al. (2013), pp. 2–4; Swanson (1976), p. 492; IEEE (1998), pp. 75–76

[131] Standardisation organisations targeted the migration process by providing software maintenance mechanisms and definitions and by defining adaptive maintenance and its relation to software migration; cf. Mellegård et al. (2015), pp. 245–247; Lientz et al. (1978), p. 466; ISO/IEC (2006), pp. 1–11; Vertes and Dufour (2009), pp. 4–7, URL see Bibliography; Draper et al. (2013), pp. 4–5, URL see Bibliography; Swanson (1976), p. 497; ISO/IEC (2006), pp. 11–15; Williams and Carver (2010), p. 31; Jamshidi et al. (2013), pp. 142–143.

specific tasks within them[132]. The proposed concept is informed by how software maintenance transforms existing software artifacts and the data management while respecting their functionality and state, prior to the cloud migration, and in accordance with the metrics that associated processes define in terms of standard key performance indicators of software quality assurance and software configuration management, verification and validation.

The cloud migration concept is framed within the six maintenance activities that the ISO standard 14764 proposes. The proposed concept could, informed by these maintenance activities, particularly focus on the execution of the process implementation as well as the problem and modification analysis. The cloud migration concept might also work during the modification, implementation, and migration phases[133]. During the *modification implementation* (3) activity, the organisation develops and tests the software product modification. As for the *migration phase* (5), the organisation modifies the software product to run it in different environments by determining the actions needed to accomplish the migration, developing the necessary mechanisms, and documenting the migration steps.

The migration step will be fundamental to the approach of the proposed concept and therefore Chapter 6 addresses it.

3.1.2 Software evolution

Software evolution is very related to software maintenance and refers to initially developing application systems to then repeatedly update them for various reasons.

[132] Each of the phases of the ISO standard follows conventions by IEEE to guide the maintenance activities to produce the correct output (data or software artifacts) after consuming or transforming its input; cf. Aloran et al. (2015), pp. 64–66; ISO/IEC (2006), p. 18; IEEE (1998), p. 76

[133] The *process implementation* (1) activity plans the maintenance procedures and plans to be executed during the rest of the maintenance process. The software maintenance process iteratively activates the rest of the activities. The *problem and modification analysis* (2) activity analyses, replicates, and verifies the modification or problem report; develops options to implement the modification; and documents and gets approval for the execution of the modification. Then, the *modification implementation* (3) activity. The *maintenance review/acceptance* (4) activity ensures that the modifications are correctly implemented. Then comes the *migration* (5) activity and once the software product has reached the end of its useful life, the software *retirement* (6) activity analyses, assists, and retires the product; cf. Aloran et al. (2015), pp. 64–66; Salehie and Tahvildari (2009), pp. 1–5; ISO/IEC (2006), p. 18

Software evolution strives to implement and validate the potential changes to a software system as the structure of evolving software will tend to degrade and therefore would require remedial action[134].

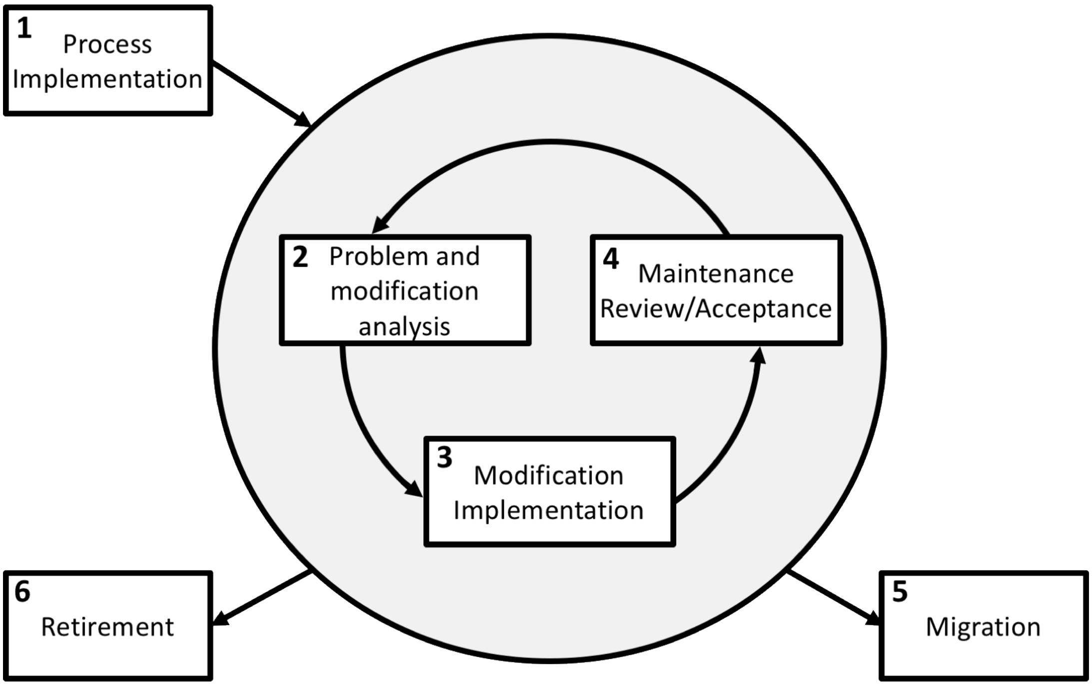

Figure 3.1: Software maintenance process as described in the ISO standard 14764 in 2006[135]

Once the proposed concept will reach a higher technology readiness level, continuous integration environments for application development could integrate it to monitor the conformance of the migrated application system, according to the decision criteria that drove the migration, either through tests run against the system components (even its source code if available). Software evolution considers how the existing software system evolves in an unpredictable manner to address changing requirements and business rules[136]. Unlike in the field of software maintenance, there is not any agreed standard for software evolution like the one Figure 3.1 shows. Nevertheless, many research works explain the challenges and activities involved and clarify the types of software evolution according to the performed activities and produced artifacts[137]. Software evolution might be a

[134]cf. Calinescu et al. (2018), pp. 1039–1043; Salehie and Tahvildari (2009), pp. 1–5; Bennett (1995), p. 19

[135]Own illustration based on ISO/IEC (2006), p. 18

[136]cf. Gao et al. (2016), pp. 2–7; Godfrey and German (2008), p. 129; Cook et al. (2001), p. 592

[137]cf. Dintzner et al. (2018), pp. 911–917; Mens et al. (2005), p. 13; Chapin et al. (2001), pp. 3–6; Kemerer and Slaughter (1999), p. 493

better option than maintenance. Considering software maintenance as general post-delivery activities and software evolution as a particular phase in its staged model.

The proposed cloud migration concept relates to the software evolution in that it could assist the evolution of an existing application system after its initial development similarly to the sequence of stages in the staged model[138] as Figure 3.2 shows. The maintenance phase falls into stages of evolution, servicing, and phase-out[139]. The proposed concept finds itself in the middle of the simple staged model. Compared to Figure 3.1, Figure 3.2 includes the servicing phase. The two concepts are complementary and the proposed concept might benefit from including their rationales in the phases of the cloud migration process.

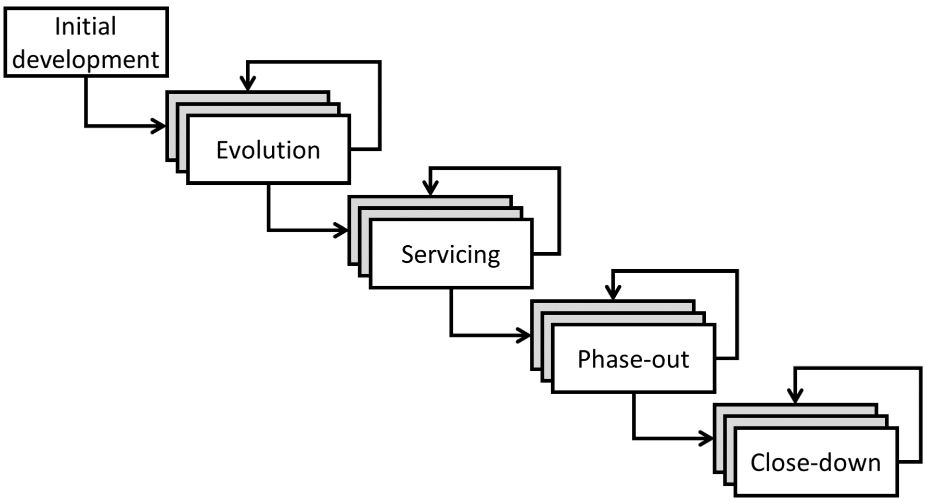

Figure 3.2: Simple staged model by Bennet in 1999[140]

The proposed cloud migration concept is informed by software evolution processes that help present the potential consequences of migrating to a different environment and of continuing with the system prior to migration. Tailoring the decision of how to migrate the application system to cloud environments to the particular decision criteria of organisations when presented with the cloud migration could be one

[138] cf. Calinescu et al. (2018), pp. 1039–1043; Salehie and Tahvildari (2009), pp. 1–5; Bennett and Rajlich (2000), p. 2, 73

[139] The servicing stage or software maturity stage occurs when the software product is no longer evolvable and organisations only perform small tactical changes such as bug fixes, basic maintenance, patches, code changes, and wrappers. During the phase-out stage, organisations do not undertake the servicing stage any more but they might keep the application system in production cf. Krupitzer et al. (2015), pp. 184–185; Alegre et al. (2016), p. 63; Weyns (2017), p. 8; Salehie and Tahvildari (2009), pp. 1–5; Lehner (1991), p. 603; Bennett and Rajlich (2000), p. 73, 80

[140] Own illustration based on Bennett and Rajlich (1999), p. 2

of the fundamental components of the proposed concept. In doing so, it would participate in the software modernisation phase until the phase-out stage. That is, the cloud migration might take place although it can play again the same role were the organisation to deem mandatory to re-run the software adaptability phase in the future.

3.2 Software re-engineering

Software re-engineering closely relates to software redevelopment because most of the re-engineering efforts propose a complete application system re-implementation[141]. Although the proposed concept might not perform a holistic re-engineering effort, it might use some of the concepts stemming from the software re-engineering knowledge base. Such is the case with reverse and forward engineering as well as the re-engineering model and its visualisation in the horseshoe model that inform the proposed cloud migration concept. The horseshoe model complements the latter by adding the architectural layer and focusing on architectural transformations towards the desired architecture.

Software re-engineering processes modify an application system or portions thereof to transform the existing "bad" system into a "good" system[142]. That an existing system needs improvement is the main inherent assumption behind software re-engineering projects[143]. An organisation re-engineering a application system usually reverse engineers it to have a more abstract description or vision; forward engineering or restructuring efforts might follow and probably include modifying the application system to meet the new requirements that the original application system could not to meet before. The proposed concept could use reverse engineering to extract the component-based structure and constraints of the target application system as well as forward engineering to generate the different alternatives to cloud-enable a system by migrating some of its components[144].

[141] cf. Bruneliere (2018), pp. 35–79; Chikofsky and Cross (1990), p. 17; Bisbal et al. (1999a), pp. 3–4

[142] cf. Noroozi and Seifzadeh (2018), pp. 134–136; Sommerville (2010), pp. 22–32; Yu (1991), p. 22

[143] cf. Bruneliere (2018), pp. 35–79; Chikofsky and Cross (1990), p. 13; *Software re-engineering examines and modifies a system to design and implement it in a new fashion. Re-engineering relates to both reclamation and renovation*

[144] cf. Singh et al. (2019), pp. 2045–2046; Suárez-Figueroa et al. (2012), pp. 9–13; Byrne (1992), p.

3.2.1 Reverse engineering

The proposed concept will use abstractions to represent the component-based architecture and information abstraction to identify the component-to-component dependencies. When software engineers reverse engineer complex software systems, they usually start by firstly examining its existing artifacts such as software, logs, or code repositories metadata. Further, they can abstract from this information to find not so obvious dependencies, or the initially software architecture which might have eroded over the time as a result of the evolution of the system to cope with new requirements[145]. The cloud migration concept might use reverse engineering to re-use software components and its data artifacts and instil good software development methods that respect privacy. And all of that while improving other metrics that the organisation want to improve by running the system on the new technological infrastructure, which maintains the existing functionality[146].

Software visualisation researchers widely use reverse engineering to address the complexity and number of changes in software systems and so might the proposed concept to show effective and efficient visualisations of the software systems evolution[147]. This helps visualising software maintenance adaptive processes to adapt the target system to the cloud environment to minimise errors and allow for seeing the big picture before performing the migration decision.

3.2.2 Forward engineering

The forward engineering process starts with high-level abstractions and logical implementation-independent designs down to the physical implementation of a application system. It complements reverse engineering as the latter does not change or replicate the subject system but examines it. With forward engineering, software engineers design the system implementation from the software requirements. The proposed concept could benefit from using forward engineering as a step in

235

[145] The IEEE SA-1219-1998 standard for software maintenance recommended software reverse engineering to support systems with software as their only reliable representation cf. IEEE (1998), pp. 75–76; Canfora and Di Penta (2007), pp. 326–331, 341; Chikofsky and Cross (1990), p. 13, 17; Kazman et al. (1998), p. 163; Mueller et al. (2000), p. 47; Canfora et al. (2011), p. 142, 147–149; Canfora et al. (2005), p. 1075

[146] cf. Rashid and Khan (2013), p. 79; Abbas et al. (2012), pp. 292–293; Rosenberg and Hyatt (1996), pp. 2–3

[147] cf. González-Torres et al. (2016), pp. 55–59; Diehl (2007), pp. 129–147;

the generation of cloud migration alternatives; that is, the multiple alternatives to migrate an application system to cloud-enabled computing environments[148].

3.2.3 Re-engineering model

The software levels of abstraction capture system characteristics and explicitly express some of them[149]. In the end, the only accurate and up-to-date system representation lies at the implementation level with actual data and executable components or scripts. Therefore, the proposed concept can assist the cloud migration by working with abstract software components, databases, and services; in addition to actual software. The integration of the concept into the development environment software engineers use would offer the potential to support the software migration at lower levels of abstraction. Software engineers describe a software system with different system representations at these four levels of abstraction and they use these abstractions to ease the software system development. Likewise, Figure 3.3 shows how to use software levels of abstraction in the general model for software re-engineering.

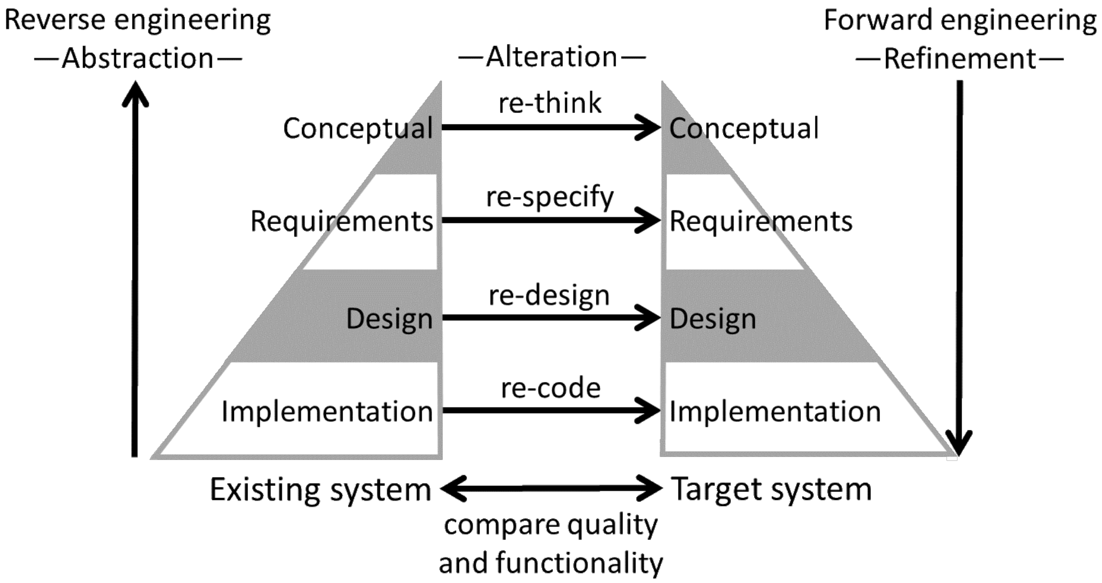

Figure 3.3: General model for software re-engineering as described by Byrne in 1992[150]

[148]cf. Bruneliere (2018), pp. 35–79; Chikofsky and Cross (1990), p. 13, 17; Crotty and Horrocks (2017), pp. 178–182; Sneed (1995), pp. 24–26; Abbas et al. (2012), pp. 292–293; Byrne (1991), p. 1349; Suárez-Figueroa et al. (2012), p. 11; Byrne (1992), p. 235

[149]cf. Rashid and Khan (2013), p. 79; Abbas et al. (2012), pp. 292–293; Byrne (1991), p. 1355, 1363–1364

[150]Own illustration based on cf. Singh et al. (2019), p. 2045; Byrne (1992), p. 230; Suárez-Figueroa et al. (2012), pp. 9–13; Figure 3.3 shows the different levels of abstraction that inform

The proposed concept might therefore allow for alteration, abstraction, and refinement as shown in Figure 3.3. Abstraction so that the cloud migration concept could use target profiling —either automatically from the executable software or letting the organisation doing it manually— as well as for the targeting of the selected cloud environment. The cloud migration concept could allow organisations model their application systems at different levels of abstraction including the computing, data storage, communications, sensitivity levels, and constraints. The modelling of the cloud environments will follow a model-driven approach to present a hierarchy that moves up and down in the level of abstraction in Figure 3.3. Alteration to keep the level of abstraction of the system representation constant but to add, modify, or delete information to it[151]. Alteration could be used in the cloud migration concept to generate the different options to migrate the target system by placing the application system components to the different cloud environments and local premises. Refinement moves in the opposite direction that abstraction does and decreases the abstraction level of a system representation by successively replacing existing system information with more details. The cloud migration concept might allow different alternatives to transform a system implementation into the target system implementation depending on the system characteristic(s) on which the software re-engineering effort focuses. Ideally, performing system characteristics changes at the level of abstraction that make them relevant.

The four types of system characteristic changes re-code, re-design, re-specify, or re-think[152]. The proposed concept could accordingly support re-coding through

the proposed concept as to how to support the decision making for software migration. The software levels of abstraction in Figure 3.3 define the software system at a particular level of detail from the conceptual specifying the reason why the system exists to the implementation level described in a language a computer understands. The conceptual and requirements levels do not describe the internal details of the system. The requirements abstraction level details the functional characteristics of the system. The design abstraction level encompasses the architectural structure of the system, the components and their interfaces, the algorithms, and the data structures. Each level of abstraction can include several degrees of abstraction. For example the high-level and detailed designs as two degrees of abstraction within the design level of abstraction. This model assumes that the software re-engineering project starts with the existing system represented at one of these levels of abstraction.

[151] Alteration typically bridges the gap between abstraction and refinement but is not essential to re-engineer a system and depends on the re-engineering strategy

[152] Re-code to change the implementation characteristics. Re-design to alter, among others, the design architecture, the data model, or the algorithms. Re-specify to add, change, or delete the system requirements characteristics. Re-think to manipulate the concepts embodied in a system to create a new system operating in a different problem domain, at least to some extent

application system profiling based on the actual code to come to the particular alternatives to migrate a system but still requiring human intervention to the very fine-grained tailoring that maximise the usage of the particularities of the cloud environment capabilities at the software binaries level. The proposed cloud migration concept might focus on the re-design of the target application system to adapt the design architecture, data model, and component-to-component communication. It could let re-specify the system requirements characteristics to maximise the return of migrating to cloud environments according to their priorities into moving cloud environments. The re-think aspect os system characteristic changes would require further implementation effort to integrate this aspect into the cloud migration concept[153].

3.2.4 The horseshoe model

The horseshoe model, which Figure 3.3 shows, is a visual metaphor of software re-engineering in harmony with the concepts behind the general model of software re-engineering adding the architectural layer and focusing on architectural transformations towards the desired architecture[154]. The model uses the basic principles of abstraction, alteration, and refinement with a strong focus on software architectures to link the existing and target software systems.

The horseshoe model inspires enabling the proposed concept so that it can work at different levels of abstraction from executable software to computing components, data components, databases, scripts, interfaces, connectors, or packages by providing its three main activities. Architecture recovery and conformance to increase the abstraction in the representations of the target application system in its base architecture. Architecture transformation to re-engineer the base architecture into the desired (or target) architecture. Architecture-based development moves the system down to more concrete representations starting at architectural representations and concepts.

[153] cf. Singh et al. (2019), p. 2045; Suárez-Figueroa et al. (2012), pp. 9–13; Byrne (1992), pp. 226–228

[154] cf. Singh et al. (2019), p. 2045; Kazman et al. (1998), p. 154; Suárez-Figueroa et al. (2012), p. 11; Byrne (1992), p. 235; Kozaczynski et al. (1992), p. 1065; Kozaczynski and Ning (1994), p. 61; Woods et al. (1998), pp. 43–64; Bass et al. (2021), pp. 22–35

[155] Own illustration based on Kazman et al. (1998), p. 155

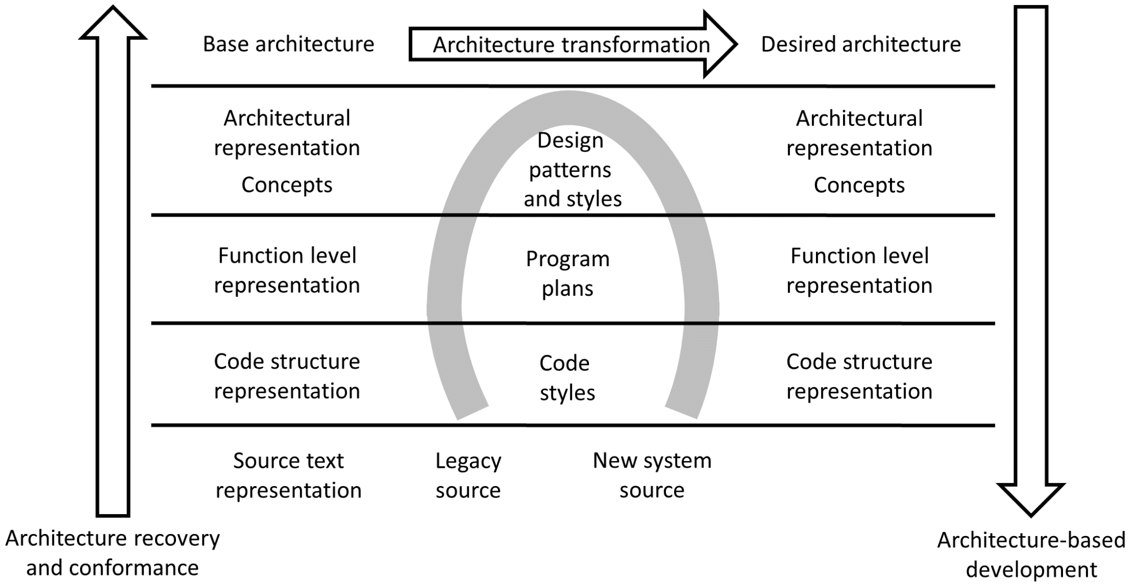

Figure 3.4: Horseshoe model for software re-engineering as described by Kazman in 1998[155]

Forward engineering allows for descending along the right-hand side of the horseshoe model to detail the abstract architectural representation of the target system. Reverse engineering moves across abstraction levels from the executable software to the base architecture[156]. The proposed cloud migration concept might enable modelling higher levels of abstraction by using documentation or developers' system acquaintance and extracting source models of the application system's functionality. This allows decision makers to work with abstract criteria to migrate a system to cloud environments and software architectures, which is the core focus of this dissertation and the proposed concept.

3.3 Software migration

The fifth software maintenance activity described in Section 3.1 includes the software migration maintenance activity [157]. This activity addresses how to consider the different software layers of the information systems being migrated, the different migration types, and how to plan them. Software migration is a software maintenance activity to move a software product from one platform or operating environment to another one which is supposedly better in some respect[158].

[156] cf. Bruneliere (2018), pp. 35–79; Bass et al. (2021), pp. 22–35; Kazman et al. (1998), p. 154

[157] Software migration is one of the six stages of the Software Maintenance Process described in the iso14764:2006 ISO standard; cf. ISO/IEC (2006), pp. 4–5

[158] Some dimensions motivating a migration might be easier maintenance of the application system in a cost-efficient manner that guarantees new business requirements without redeveloping

The proposed cloud migration concept could support the software migration to cloud environments by adapting the target application system to the new technology without intending to change its functionality. Targeting therefore organisations that want to keep the functionality approved by the customer while enhancing the maintainability of the system by improving the design of the system around the new technology. Hence, the proposed concept is informed by software migration to support the decision making on how to cloud migrate an application system to fulfil the requirements of that target group. Reverse and forward engineering techniques might help in analysing, comprehending, and transforming the application system target for migration by restructuring and adapting it. The proposed concept could, according to existing literature, analyse and evaluate the challenges, risks, and functional changes to keep the business features of the legacy system[159].

Software migration relates so closely to software re-engineering that some researchers use both terms interchangeably[160]. However, according to the definition in Section 3.2, the proposed concept does not support software re-engineering described as a complete system re-implementation. This dissertation positions software re-engineering closer to software redevelopment with a complete system overhaul or a re-implementation[161]. Software re-engineering extends re-implementation of the legacy information system whereas software migration moves a software product to another environment. Some re-engineering projects merely restructure the target software system to improve internal quality attributes such as availability or maintainability. In other cases, the lack of evolvability of a system motivates re-engineering whereby an organisation migrates its systems to a new environment to improve its ability to accommodate future changes. For example, if a system uses some underlying technologies that no organisation support anymore.

everything cf. Sneed (2016), pp. 1941–1946; Brodie and Stonebraker (1995), p. 12; Sneed et al. (2009), pp. 12–29; Winter (2004), pp. 1–2; Bisbal et al. (1997), pp. 2–5

[159] cf. Sobhani et al. (2018), pp. 20–25; Shoaib et al. (2017), pp. 192–195; Khadka et al. (2013), pp. 51–56; Fuhr et al. (2013), p. 65; Sneed et al. (2005), p. 3; Winter and Ziemann (2007), p. 107; Bisbal et al. (1999b), p. 2, 39 103

[160] cf. Sneed (2015), pp. 1–6; Sneed (1996), pp. 104–106; Aebi (1997), p. 305

[161] Re-engineering a software system entails examining, understanding, and altering the system to put it back together in a new form; cf. Assunção et al. (2017), pp. 2972–2981; Chikofsky and Cross (1990), p. 13; Bisbal et al. (1999a), p. 12; Ganti and Brayman (1995), pp. 157–160; Warren (1999), pp. 10–15; Tilley and Smith (1995), pp. 7–14

3.3.1 Software migration methodologies

Software migration projects differ in the extent of the adaptation effort involved. In some cases the executable software or services are under the control of the organisation migrating the system, which can then legally change them. The state of the art of software migration presents metrics to measure this effort in, for example, number of reused services or microservices, changed classes, modified lines of code, or new adapter layers or components[162]. In some cases third-party components compose the system target to migration and might require special migration measures and a licensing agreement to migrate an application system to a computing environment for which it was not designed and implemented. In addition to these non-technical aspects, the proposed concept might allow importing the executable software of application systems to extract the coarse-grained structure of the target application system to assess the development effort through standard metrics. The proposed concept could aim at migrating not only executable software but data, (micro)services, containers, and user interfaces as well.

Similarly to other re-engineering activities, software migration is an error-prone discipline that can benefit from combining several techniques —such as software visualisation, dynamic program analysis, and knowledge recovery— to cope with its complexity[163]. The cloud migration concept might adopt these techniques using them according to the type of migration performed and its complexity. By using them, it would become easier for decision makers to gain understanding of the legacy code, software components, and software services so that they can make an informed and fact-based adaptation of the application system to the newly selected environment. The large size of application systems make this a complex task that would benefit from an structured and methodical plan and execution of the migration project that the proposed concept will support.

3.3.2 The migration of different software layers

The proposed concept is informed by how software layers affect the migration as they are intrinsic to most of the application systems and systems; in addition to

[162]cf. Frey et al. (2018), pp. 3–7; Frey (2014), pp. 211–215; Sneed et al. (2005), p. 5
[163]cf. Satish and Mahendran (2017), pp. 3–7; Wu et al. (2005), p. 609

the application system constraints. The concept could consider the interplay of software components in different layers so as to help decision makers improve their application system's business processes in a holistic way.

Software engineers implement different adaptation activities depending on the layers they migrate and they usually pay attention to different concerns depending on the selected layer. In the usual three-layered architecture —with data, business-logic, and presentation layers— organisations could move data by migrating just the data layer or as a part of the migration of the whole application system. In any case and in this one in particular, security and confidentiality are fundamental concerns of the proposed concept in addition to the adaptation efforts when the former environment differs with the new one in terms of database models or database management systems. A different kind of cloud migration project entails moving business-logic components to the new environment to overcome integration problems due to the complexity of the software solution and to reduce management costs. The organisation in charge of the migration has to understand and adapt the affected business processes to the new environment without disrupting the provided functionality[164]. The proposed concept is informed by the interplay of these business-logic components with other layers and components and might use this input to help organisations enhance their business processes. In some other cases, the organisation migrate the presentation layer or together with both the data and business-logic and the concerns and issues affect all levels. Cross-cutting concerns include cost, performance, or quality of service. The proposed concept might facilitate the partial migration to cloud environments that uses the infrastructure at the premises of migrating organisation to overcome some of the challenges of cloud migrating an application system, such as privacy and trust issues.

3.3.3 Migration to new computing environments

The proposed concept targets the software migration to a particular kind of new computing environment, the cloud computing environment. Although each migration project entails specific challenges and issues that could be classified in

[164] cf. Bergmayr et al. (2018), pp. 5–7; Andrikopoulos et al. (2013a), p. 493, 525–530; Wu et al. (1997), pp. 129–130

migration types, they are not immiscible. A software migration project might pertain to several of these types with a specific migration type requiring applying activities from other migration types. Such a heterogeneous overlapping migration project would therefore require a mix of techniques to address all the migration challenges. For example, the type *migration to a new hardware platform* usually implies adapting the application system; hence, the realms of the *migration to a different operating system* type.

Different migration types exist depending on whether software engineers migrate their application systems to a new framework, fundamental library, or middleware; a new development tool; a new operating system; a new database management system; a new programming language; a new programming paradigm; a new hosting provider; or a new hardware platform[165]. Software migration projects are complex undertakings that benefit from using a decision support concept to decompose the legacy system to then cost-efficiently organise a simpler migration process[166] to a different programming paradigm. It happened to migrate to migrate to object-orientation, component-based systems, service-oriented architectures, and within this dissertation to cloud-based computing environments[167]. In the end, new target environments will always appear so the migration to those environments continues to be a hot topic as different environments appear. The proposed concept, in order to be applicable to the migration to new cloud environments, would support the migration to a new hosting provider, which is in fact a new computing platform.

3.3.4 Migration planning

An organisation can define the actions to migrate an application system to then carry out the steps required, in addition to documenting them[168]. Organisations

[165]cf. Frey et al. (2018), pp. 3–7; Frey (2014), p. 12

[166]cf. Shumaker et al. (2017), pp. 5442–5445; Wu et al. (2005), pp. 608–609; They applied dynamic program analysis and software visualisation together with knowledge recovery and divide-and-conquer techniques to prioritise their activities

[167]Migration to object-oriented programming paradigms; cf. Sobhani et al. (2018), pp. 20–25; Shoaib et al. (2017), pp. 192–195; Khadka et al. (2013), pp. 51–56; De Lucia et al. (1997), p. 125; Zou and Kontogiannis (2002), p. 530. Migration from the object-oriented to the component-based paradigm; Lee et al. (2003), p. 336. Migration to service-oriented architectures; Fuhr et al. (2012), p. 23; Canfora et al. (2008), p. 466; Van Hoorn et al. (2011), p. 12; Fuhr et al. (2013), p. 84

[168]According to the ISO/IEC 14764 standard; cf. ISO/IEC (2006), p. 3

must define those steps according to the particularities of the old and new computing environments as well as the old and new architecture. The organisation might accordingly have to modify the software products or data to migrate them from an old to a new operational environment following the ISO/IEC 12207 standard[169]. That is, the organisation develops a *migration plan*, notifies the organisations in charge of the migration, trains for migration, notifies the completion of the migration, assesses the impact of the new environment, and archives data. The proposed concept focuses on the migration plan and uses the *configuration management* process of controls all the artifacts of the *migration activity*.

The proposed concept considers the concepts and six main tasks of the ISO/IEC 12207 standard. It could involve the user —the organisation targeting the cloud migration— in the process to properly follow the migration plan, which includes the tasks to analyse and define the migration requirements. Organisations usually develop migration tools to convert software products and data to then execute and verify the migration and support the old environment in the future[170].

Migrating organisations usually include system users in the loop to analyse the migration requirements and determine the impact of migrating, its risks, and migration effort. Organisations in charge of maintenance identify the data collection requirements for post-operation review, identify the needed support activities for the old environment and migration tools. Subsequently, organisations decide whether to develop or acquire migration tools, decompose the software products and data to adapt them to the new environment, and prioritise these conversion tasks. After having scheduled the migration, the actual migration happen while both, the old and new system, operate in parallel. For a certain time, the organisation will probably need to support the old environment and test the new one. An organisation might implement sixteen task-steps to the develop the migration plan using the input from the users of the system: analyse the migration requirements; determine the impact of migrating the software product; establish a schedule for performing the migration; identify data collection requirements for post-operation review; define and document the migration effort; determine and mitigate risks; identify needed migration tools; identify support for the old environment; develop and/or

[169]cf. ISO/IEC (2008), p. 1
[170]cf. ISO/IEC (2008), p. 3, 9–11

acquire migration tools; incrementally decompose software products and data for conversion; prioritize conversion of software products and data; convert software products and data; migrate software products and data to new environment; run parallel operations; verify migration through testing; and provide support for old environment.

The cloud migration concept considers the ISO standard to demarcate as the final output of the migration process the migration plan and tools, the notification of intent, the migrated software product, the notification of completion, measurements, and archiving of the data[171]. The proposed concept might focus on one particular output of them; that is, the migration plan that the cloud migration concept generates. The proposed concept could produce alternative migration plans to support organisations in migrating their systems to cloud environments that fulfil their requirements and offer the pre-migration functionalities.

[171] cf. ISO/IEC (2006), p. 6

4. Cloud migration

This chapter explains the analysis of the state of the art on cloud migration and related work on software architecture modelling. It intends to frame the research design of the proposed concept. This research design guides the upcoming activities, see Chapter 5, to gather the requirements to finally fulfil them with the design of the concept proposed in Chapter 6. Figure 4.1 puts the sections of this chapter in the context of the scope of this dissertation (for more details on the scope of this dissertation refer to Section 4.7).

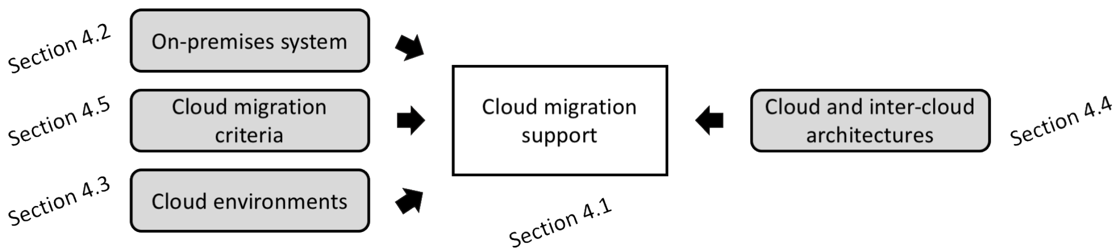

Figure 4.1: Structure of the cloud migration chapter in relation to the scope of this dissertation[172]

The chapter starts with Section 4.1 describing an overview of the systematic literature review and filled research gap to then pass to a section on cloud migration support (Section 4.2), which explains the "competition" to the cloud migration concept proposed in this dissertation. Next, this chapter describes the input models that the proposed concept will use to support the cloud migration decision. That is, the on-premises application system modelling explained in Section 4.3 to describe

[172] Own illustration

how contributions that model the target application system and its requirements prior to its cloud migration. The works on cloud environments modelling as well as cloud and inter-cloud architectures mentioned in Sections 4.4 and 4.5 to define the computing environments target for the migration. The cloud migration criteria come in Section 4.6 so as to detail which criteria drive organisations in deciding how they will migrate their application systems. Finally, Section 4.7 moves into the research design and research frame of reference. Both based on the state of the art and related work described in the aforementioned sections. And it includes contextualising the scope of this research, the research method followed, and the research frame of reference, processes, and methods.

4.1 Overview of the systematic literature review and filled research gap

4.1.1 Overview of the systematic literature review

The process for this literature review explored concepts related to cloud migration by searching specific keywords described in Sections 4.2 to 4.6 from 2012 to 2023 in ACM digital library, IEEE Xplore, ScienceDirect, SpringerLink, and Google scholar. Those specific keywords were selected based on own experience and literature knowledge. Figure 4.2 describes the process of that literature analysis carried out for each of the topics mentioned in Sections 4.2 to 4.6.

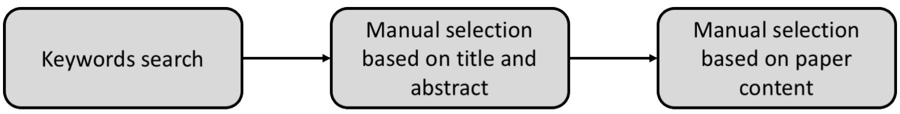

Figure 4.2: Literature analysis process followed per topic-specific group of keywords[173]

Analysed papers, standards, and white papers in cloud migration: 399

Analysed projects and software related to cloud migration: 30

The following table shows an overview of the literature review conducted to analyse the state of the art of cloud migration. Table 4.1 explains the number of papers

[173]Own illustration

found when researching on the following five concepts concepts and explains the number of papers reviewed out of the hits. The five concepts covered in the systematic literature review include: cloud migration support, on-premises application system modelling, cloud environments modelling, cloud and inter-cloud architectures, and cloud migration criteria. Sections 4.2 to 4.6 delve into the details and contributions found on those papers which were reviewed. The table highlights which sections explain each part of the systematic literature review per concept in addition to the related research question.

Table 4.1: Overview of the systematic literature review

Concept	**Reviewed**	**Section**	**RQ**[174]	
Cloud migration support	55	Section 4.2	RQ 1	
On-premises application system modelling	15	Section 4.3	RQ 4	
Cloud environments modelling	42	Section 4.4	RQ 3	
Cloud and inter-cloud architectures	95	Section 4.5	RQ 1 & RQ 3	
Cloud migration criteria	222	Section 4.6	RQ 2	

4.1.2 Used keywords

Per each of the concepts mentioned in the table, a number of keywords were used to find relevant articles and research work. For the case of cloud migration support, the following keywords were used: legacy software migration, cloud migration, software modernisation, cloud computing, cost reduction, reverse engineering,

[174] RQ stands for research question. That is, the research question the part of the systematic literature review wants to address. Section 1.2 in Chapter 1 explains the four research questions of this dissertation: *Research question 1:* On how a decision support concept assists organisations in the decision-making involved in cloud migrating their application systems. *Research question 2:* To investigate which criteria driving the cloud migration decision. *Research question 3:* What are the criteria or requirements of organisations to select to which cloud environments to migrate their application systems? *Research question 4:* Which requirements and characteristics of the application system prior to the cloud migration do organisations consider in the decision-making process?

software maintenance, model-driven engineering, software quality. Cloud and inter-cloud architectures: inter-cloud, Peer-to-peer inter-cloud, centralised inter-cloud, cloud federation, multi-cloud service, and multi-cloud service.

For the systematic review of literature about on-premises application system modelling: application systems modelling, Object Management Group (OMG), Unified Modelling Language (UML), systems architecture, and meta-modelling.

For the concept of cloud environments modelling: cloud computing, cloud modelling, cloud language, cloud ontology, cloud environment modelling, cloud selection. For cloud and inter-cloud architectures: inter-cloud, peer-to-peer inter-cloud, centralised inter-cloud, cloud federation, multi-cloud service, and multi-cloud service. And finally, for cloud migration criteria: cloud computing, cloud migration criteria, cloud adoption, cloud migration, and criteria for cloud selection.

4.1.3 Research gap

The research gap this dissertation fills stems first of all from the approach to the research problem from a information systems standpoint. That is, the research problem is not understood as a merely technical problem but a holistic undertaking influenced by regulatory and organisational aspects in addition to business concerns and guarantees in terms of security and privacy. In addition to closing the gap by considering the decision-making problem a multidimensional one, the proposed concept also adds to the state of the art by semi-automatically analysing cloud migration alternatives for some criteria while allowing organisations the freedom to tailor the project to their needs. In addition, by considering reverse engineering from the actual code of the target application system it lays the first stones to close the gap between cloud migration projects at design time to potentially move to the run time.

4.2 Cloud migration support

This section presents relevant research works, studies of cloud migration in industrial settings, and industrial best practices to address the migration of application systems, systems, and application systems to cloud computing environments. The study of the state of the art of cloud migration support includes characteristics of the proposed solutions and how this dissertation learns from them to apply the lessons to the cloud migration concept proposed in Chapter 6. This dissertation presents here the state of the art to inform the future design of the proposed cloud migration concept and includes contributions to cloud migration in research and industry. Likewise, experiences gained in industrial cases of cloud migration in the span of the research conducted for this dissertation[175] and how the state of the art of cloud migration looks like shaped the research work related to this dissertation. All together, this section helps define the scope of the research framework of the proposed concept and the cloud migration process to use in it.

The process for this literature review explored concepts for cloud migration support searching from 2012 to 2023 specific keywords in ACM digital library, IEEE Xplore, ScienceDirect, SpringerLink, and Google scholar: legacy software migration, cloud migration, software modernisation, cloud computing, cost reduction, reverse engineering, software maintenance, model-driven engineering, and software quality. These keywords were selected based on own experience and literature knowledge. The state of the art found falls into one of the four categories below (although they are not immiscible categories) depending on which subset of related research works they focus. That is, whether they come from model-driven engineering[176] or software evolution, whether they use the Analytic Hierarchy Process, and on which level of the cloud computing stack they focus: the IaaS, PaaS, and SaaS levels.

4.2.1 Model-driven engineering

As proposed in this dissertation's cloud migration concept, the Artist project[177] aimed at facilitating the modernisation and transformation of legacy systems, software assets, and business which are not yet using cloud-based environments. As documented in several published research papers, the consortium worked towards automating the evolution of application systems and software to cloud

[175] cf. Al-Ali et al. (2018b), pp. 45–53; Juan-Verdejo and Surajbali (2016), pp. 11–23; Juan-Verdejo et al. (2014a), pp. 43–48; Juan-Verdejo (2012), pp. 14–27; PaaSport Consortium (2016a), p. n/a, URL see Bibliography; KnowHow Consortium (2014), p. n/a, URL see Bibliography

[176] Model-driven engineering refers to software development methodologies using conceptual models of the topics related to a specific problem. Model-driven uses abstract representations of the knowledge and activities of a particular application domain rather than algorithmic concepts: cf. Mani et al. (2017), pp. 2354–2356; Poole (2001), pp. 7–13

[177] cf. ARTIST Consortium (2021), p. n/a

computing delivery models and focused on the SaaS deployment model[178]. The proposed concept learns from their methodology to the Software as-a-Service level and includes the migration to the Infrastructure- and Platform-as-a-Service abstraction levels[179].

A lesson learnt coming from the model-driven engineering discipline applied to cloud migration support[180] is to treat abstraction levels of the cloud stack differently to analyse the different aspects important to each of the cloud levels and how to measure them to the extents possible. That is, similarly to how Artist aimed at addressing architectural characteristics fundamental to cloud-enabled application systems such as multi-tenancy or scalability as well as analysing the technical feasibility of the execution of a cloud migration project prior to the beginning of the cloudification[181].

The research on cloud migration mentioned here motivates further developing concepts for re-engineering application systems to cloud migrate, maintain, and evolve them. The Artist project proposed the methodology in Figure 4.3 to benchmark and evaluate the different dimensions upon which the cloud migration has an effect. Based on that the methodology envisions assessing the maturity of the technological and business models of the target software system. All of these features inspired considering their inclusion in the proposed concept but with the focus put on the holistic cloud migration criteria, the feasibility of the cloud migration of the target application system, the costs involved, and the implications and benefits of using the target computing frameworks and platforms.

Figure 4.3 explains the stepwise procedure to reuse meta-modelled software components for specific processes according to the semantics of the modelled components. Inspired by this work and previous ones, the related work of this

[178]cf. Bruneliere (2018), pp. 71–79; Menychtas et al. (2013), p. 424

[179]Artist improved a prior project (Remics) to consider the non-functional requirements SaaS application systems such as performance, portability, or testability that Remics did not take into account. Similarly to the Artist project, Remics also motivated digging into how approaches using model-driven approaches in Service-Oriented Architectures (SOA). The project focuses on cost-efficiency and reliability and uses both MDA and ADM: cf. Bruneliere (2018), pp. 71–79; Mohagheghi and Sæther (2011), pp. 507–510; Khusidman and Ulrich (2007), pp. 1–7

[180]Ibid.

[181]cf. ARTIST Consortium (2021), p. n/a; Menychtas et al. (2014), pp. 137–139; Ibarra et al. (2014), p. 390

[182]Own illustration based on the Artist migration methodology based on Alonso Juncal et al. (2013), p. 66 and that the proposed concept tailored to its requirements as Figures 6.3 and 6.4 show

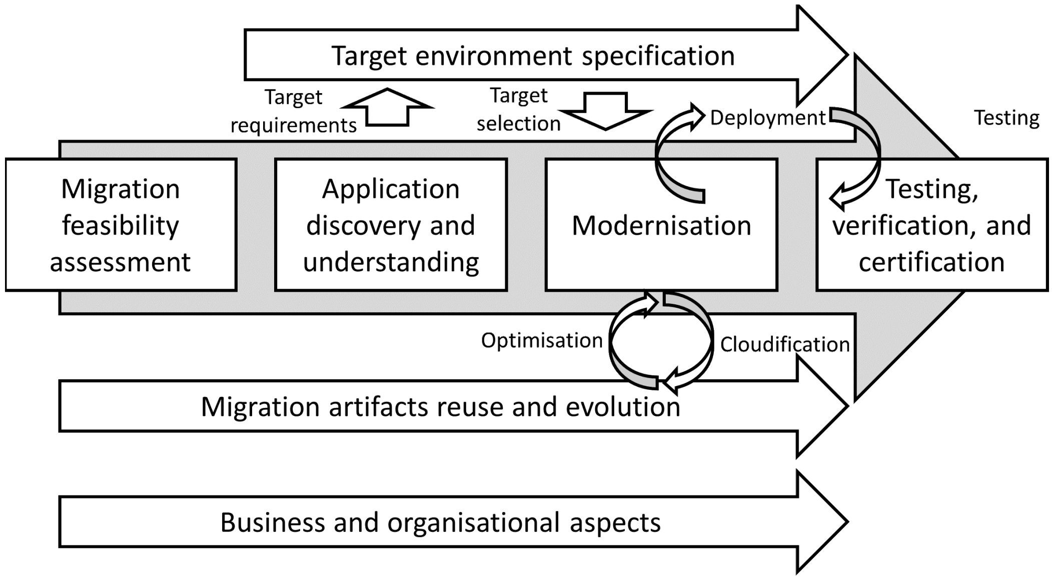

Figure 4.3: The scope of this dissertation fit within the Artist migration methodology[182]

dissertation shows how to use model transformations and model querying to handle the information about the target application system and adapt it to be run in the selected cloud environments[183]. The proposed concept might benefit from the use of model-driven to model the status-quo target application system and the multiple available cloud environments.

The CloudScale project[184] worked on scalable service engineering and published related papers. This research work motivated considering the inclusion of mechanisms to analyse, predict, and resolve scalability issues (at design time) in the proposed cloud migration concept. CloudScale uses existing modelling approaches described in some research papers to study the design and implementation of the control and data flow of the scalability processes by measuring the scalability of composed and simple cloud services[185]. That way, the proposed concept is informed on how they use model-driven tools and methods to model and analyse the scalability and cost of cloud-enabled application systems to exploit elasticity but not over-provisioning[186]. Very relevant to this dissertation, their mechanisms

[183] This dissertation includes the study and research related to these model-driven and modelling techniques —model transformations and model query— in an research exchange with King's College London: cf. Juan-Verdejo et al. (2014b), pp. 467–474

[184] cf. Cloudscale Consortium (2015), p. n/a, URL see Bibliography

[185] The Palladio Component Model for performance assessment: cf. Huber et al. (2016), pp. 432–452; Krebs et al. (2014a), pp. 914–921; Becker et al. (2009b), pp. 7–12

[186] cf. Kistowski et al. (2017), pp. 10–15; Brataas et al. (2016), pp. 7–11; Lehrig (2014), pp. 1–2

assess and offer architectures sub-optimal to the several technical-related cloud migration criteria. However, the focus of this dissertation does not only lie in these technical dimensions, like in CloudScale, as it takes a holistic approach to include other non-technical related cloud migration criteria[187].

4.2.2 Software evolution

Some research papers working on software evolution using decision points in cloud migration motivate the use of model-driven reverse engineering techniques in the cloud migration concept[188]. These two research papers proposed methodologies to estimate the cloud migration effort, from which the proposed concept learns as to how to assess the cost of the cloud migration. This research proposed using software complexity metrics or heuristics to automate the multi-criteria cloud migration assessment while relying on user intervention for criteria which cannot be automatically assessed.

Modelling techniques for reverse and forward engineering in cloud migration have got the potential to increase the understandability and evolvability of the target application system[189]. Understandability and evolvability are arguably cloud migration criteria that influence the cloud migration decision. Bruneliere et al. envision a repository of meta-models, transformations, specifications, and domain models for software transformation.

Their approach proposes using technology-specific meta-models at lower levels of abstraction and generic meta-models at higher levels. In that sense, the Artist project relates to the proposed concept in its take-home message of considering the different levels of abstraction to define the cloud environments and the target application system. Likewise, one could argue that it makes sense that the cloud migration criteria follows such a structure that comes from the abstract to the

[187] The CloudScale research partners publish on their website nightly build versions of their CloudScale Environment cf. CloudScale-Consortium (2021), p. n/a, URL see Bibliography as a desktop Java-based application integrating the different tools they produced: the analyser, extractor, dynamic spotter, and static spotter Brataas et al. (2016), pp. 1–14; Brataas et al. (2013), pp. 335–338. The consortium used for the CloudScale environment the tools fundamental to each research partners' field of expertise such as the analyser based on the Palladio Component model for self-adaptive application systems able to scale in and out Reussner et al. (2016), pp. 169–180; KIT et al. (2021), p. n/a, URL see Bibliography; Von Detten and Lehrig (2013), pp. 1–2

[188] cf. Ferdiana and Putra (2018), pp. 1–6; Tran et al. (2011), p. 265

[189] cf. Bruneliere (2018), pp. 71–79

specific depending on the desired level of detail with regards to a criterion. The lesson learnt here is that an acceptable compromise towards automatic cloud migration criteria assessment might entail using the hierarchically-defined cloud migration criteria for semi-automatic assessment of the objective cloud migration criteria and include the human in the loop for the subjective criteria or objective criteria without a provided metric.

Although in the specifics Artist and the proposed concept might differ, the overall approach to migration is very similar in terms of methodology[190]. That is, the methodology in Figure 4.3 for the Artist project and presented in the upcoming Figures 6.3 and 6.4 in Chapter 6 for the cloud migration concept. Their approach goes a bit deeper to the model-driven approach while the concept proposed in this dissertation focuses more in the cloud migration criteria. Figure 4.3 shows the use of the pre-migration phase to assess the potential improvements and disadvantages of the software modernisation using reverse engineering to discover the model of the target application and forward engineering to modernise the target software[191]. Reverse and forward engineering methods can help de-construct the pre-migrated application system to construct the cloud-enabled application system while using metrics to assess the functional and non-functional properties of the cloudified application system in comparison to the on-premises application system.

4.2.3 The Analytic Hierarchy Process

The cloud migration decision making is presented in this dissertation as multi-criteria decision making that depends on functional and non-functional cloud migration criteria. Some of these cloud migration criteria might not be easily or possibly measured while others might be difficult to quantify or purely subjective. Critical thinking is required to select a cloud migration alternative as this decision depends on multiple cloud migration criteria, which depend on each other. These interdependencies and the potentially large number of cloud migration criteria affecting the overall raking of the different cloud migration alternatives according to the priority organisations give to that particular cloud migration criterion. Arguably, this is a complex decision for which a human might need assistance. The Analytic

[190] cf. Alonso et al. (2017), pp. 397–404; Alonso Juncal et al. (2013), pp. 70–71
[191] cf. Bergmayr et al. (2018), pp. 9–12; Bergmayr et al. (2013), pp. 465–468

Hierarchy Process (AHP) fulfils these requirements for multi-criteria decision-making[192].

AHP is a structured technique for organising and analysing complex decisions such as the cloud migration multi-criteria decision[193]. Based on mathematics and psychology, it helps quantifying the weights of cloud migration criteria according to how organisations estimate their relative magnitudes through pairwise comparisons. Organisations migrating application systems compare the relative importance between two items with regard to the prioritised list of cloud migration criteria. The different cloud migration alternatives have got many key performance indicators and cloud migration criteria with sub-criteria, which makes the ranking process a complex task. Weighed sum-based methods cannot be directly applied because of the hierarchical structure and because some cloud migration criteria do not have a numerical value —such as trust. AHP helps overcome the challenge of quantifying each cloud migration criteria, aggregating them in a meaningful manner, and comparing cloud migration alternatives based on all cloud migration criteria. AHP can be easily adapted to any number of cloud migration criteria with sub-criteria in three phases to: decompose the problem, judge the priorities, and aggregate these priorities. AHP is not a new technique and has been used for cloud migration support before.

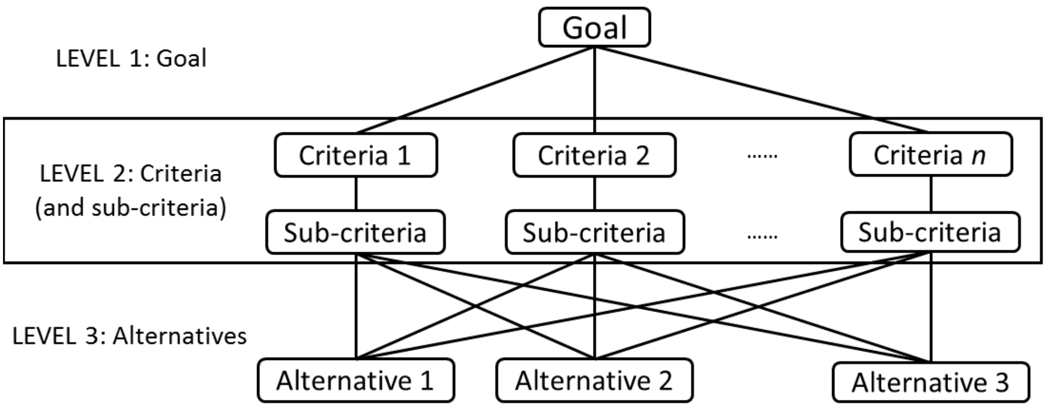

Figure 4.4: AHP's three levels[194]

[192] Tzeng et al. compare multi-criteria decision making: cf. Tzeng and Huang (2011), pp. 143–156; Saaty et al. proposed AHP Saaty (1977), pp. 245–252
[193] Ibid.
[194] Own illustration

Some research works, such as the Service Measurement Index framework or the research work by CLOUDS Laboratory of the University of Melbourne (SMI-Cloud)[195], use the Analytic Hierarchy Process to support the multi-criteria decision making[196]. The overall raking depends on each criterion's priority in the criteria weighing phase. The ranking mechanisms based on the Analytic Hierarchical Process considers how different features of the cloud-enabled application system relate and depend on each other to conform an overall ranking of cloud migration alternatives. In turn, this process provides a quantitative basis to perform an informed decision when migrating existing application systems to cloud environments. Figure 4.4 highlights the focus on the middle layer of the AHP conceptual description that informs this dissertation. Organisations pairwise compare each cloud migration criterion to one another to find their relative importance in comparison with one another.

4.2.4 The cloud computing stack: IaaS, PaaS, and SaaS

The provision of application systems to the IaaS, PaaS, and SaaS deployment models also lied at the core of the Remics project[197]. How Remics proposed reusing and migrating legacy application systems to interoperable cloud services informs this dissertation about how to address interoperability. Accordingly, Section 6.4 in Chapter 6 and Section 7.2 in Chapter 7 describes considerations about interoperability within the scope of this dissertation. An important lessons here relates to the potential of using model-driven techniques and engineering to adapt legacy application systems to cloud environments at different levels of the cloud stack[198].

Architecture-driven approaches to support cloud migration of application systems while increasing their portability to the the IaaS, PaaS, and SaaS deployment models[199]. The self-adaptivity part of Remics remains out of the scope of this

[195] cf. Ortiz et al. (2018), pp. 555–557; Siegel and Perdue (2012), pp. 411–415; Garg et al. (2011), pp. 212–214; Tran et al. (2009), p. 1378

[196] Tzeng et al. compare multi-criteria decision making: cf. Tzeng and Huang (2011), pp. 143–156; Saaty et al. proposed AHP Saaty (1977), pp. 245–252

[197] The Remics FP7 project: cf. REMICS Consortium (2013), p. n/a, URL see Bibliography

[198] cf. Alfraihi and Lano (2017), pp. 451–454; Krasteva et al. (2013), p. 1; Mohagheghi et al. (2010), pp. 3–6

[199] cf. Gholami et al. (2016), pp. 31–40; Ahmad and Babar (2014a), p. 7; Ahmad and Babar (2014b), p. 1

dissertation but the proposed concept does incorporate the understanding of Ahmad et al. of cloud migration as a multidimensional decision that includes regulations and policies in addition to business-related, legal, and cultural aspects. A multidimensional decision-making process that depends of the cloud provider and their level of service (IaaS, PaaS, and SaaS) and how this choice affects the technical cloud migration criteria[200].

4.3 On-premises application system modelling

The model of the on-premises application system, hence the status-quo application system prior to migration, could be one of the inputs to the cloud migration decision making to design how to migrate the target application system to cloud environments according to its architectural design, requirements, and constraints.

The process for this literature review explored concepts for application system modelling searching from 2012 to 2023 specific keywords in ACM digital library, IEEE Xplore, ScienceDirect, SpringerLink, and Google scholar: application systems modelling, Object Management Group (OMG), Unified Modelling Language (UML), systems architecture, meta-modelling. These keywords were selected based on own experience and literature knowledge. Accordingly this section describes contributions to model an existing software system to different levels of detail. The focus lies on existing research papers describing the topic and on explaining the meta-modelling of metrics and application systems —Knowledge Discovery Meta-model (KDM), Structured Metrics Metamodel (SMM), and Meta-Object Facility (MOF)[201]— as well as particular frameworks that might be of use to this dissertation (MoDisco).

Existing cloud migration literature motivate incorporating the application systems architecture prior to the cloud migration or the status-quo of the application system target for migration[202]. These research works motivated using, in this dissertation,

[200] cf. Alfraihi and Lano (2017), pp. 453–456; Juan-Verdejo and Baars (2013), pp. 35–36; Mohagheghi and Sæther (2011), pp. 507–510; Khusidman and Ulrich (2007), pp. 1–7

[201] The text below, within this section, explains the meta-modelling concepts and metrics such as the Knowledge Discovery Meta-model (KDM), Structured Metrics Metamodel (SMM), and Meta-Object Facility (MOF) in more detail

[202] Cloud reconfiguration; cf. Frey et al. (2018), pp. 12–19; Frey (2014), pp. 141–145. Cloud migration of an open source software application system; Rai et al. (2015), pp. 7–9; Babar and Chauhan (2011), p. 50. Others; Kaisler and Money (2011), pp. 1–4; Gustavo et al. (2004), pp. 91–92

approaches to allow software architects to model application and application system requirements in order and let them make architectural decisions. Organisations can model their deployments as a composition of different data and processing software components running in different computing nodes. For example, they can use UML deployment diagrams in Unified Modelling Language to model the physical deployment of artifacts to a node[203]. Organisations can use these models to define the different components composing their application systems. In addition, these models allow for detailing the nodes where data or application components run as well as the connections between them.

The use of these modelling approaches in this dissertation can help identifying the deployment, requirements, and constraints of the application system in order to respect the constraints and fulfil the requirements in the target application system after the migration. The premise of this dissertation is that an application system might migrated to cloud environments while keeping some of its components at the local premises of the organisation. The MoDisco framework could be an initial step to adopt existing model-driven tools while extending it to address the software modernisation with cloud environments[204]. The framework provides multiple tools that could help modernise legacy application systems focusing in their evolution. MoDisco follows model-driven approaches to support mature and flexible modernisation activities and includes specifications such as KDM and SMM. KDM is arguably the most complete meta-model in literature and thoroughly specified and standardised[205]. MoDisco implements this publicly available specification for architecture-driven modernization by the Object Management Group (OMG)[206] in Java. It is a framework typically used in the model-driven community for software modernisation efforts such as the contribution of this dissertation, which has been applied to research and industrial projects[207]. Likewise, MoDisco implements SMM that can be used to define, represent, and exchange measures and measurements related to a structured information model.

[203]cf. Platt and Thompson (2019), pp. 1452–1456; Torre (2016), pp. 53–54; OMG (2015), pp. 21–35, URL see Bibliography
[204]cf. Bruneliere (2018), pp. 42–55; Bruneliere et al. (2014), p. 1012; Madiot (2010), pp. 12–18
[205]cf. Arcelli Fontana et al. (2017), p. 1122; Pérez-Castillo et al. (2011), pp. 525–529
[206]cf. OMG (2016), pp. 1–350
[207]cf. Koltun et al. (2019), pp. 687–692; Bruneliere (2011), pp. 21–24

The Object Management Group designed the Knowledge Discovery Metamodel (KDM) that represents existing application systems via a shared intermediate metadata and facilitates modernising existing software, managing the IT portfolio, and improving software quality assurance. It uses Meta-Object Facility (MOF) that the OMG defines with an XML Metadata interchange or XMI to exchange metadata information between tools and ensure the interoperability between tools and prototypes for software maintenance, evolution, assessment, and modernisation. MoDisco gathers information about existing artifacts for analysis purposes with KDM, which does not represent an executable model but supports the Structured Metrics Metamodel or SMM. SMM describes the metrics used to quantify the criteria in the migrated application system and defines a standard Metrics Metamodel that allows for representing, defining, and exchanging measures and its metadata. The measurement information allows the metric extraction (prototypical) tools exchange information with this XMI format.

4.4 Cloud environments modelling

The process for this literature review explored concepts for cloud environments modelling searching from 2012 to 2023 specific keywords in ACM digital library, IEEE Xplore, ScienceDirect, SpringerLink, and Google scholar: cloud computing, cloud modelling, cloud language, cloud ontology, cloud environment modelling, and cloud selection. These keywords were selected based on own experience and literature knowledge.

A very fundamental part of the cloud migration decision includes how to make sense of the target environment; the cloud environment. Multiple cloud environments modelling approaches and cloud modelling languages exist to describe the cloud computing environments, their features, and how they support different kinds of cloud migration projects. Projects to either cloud migrate an application system, develop it on a cloud-first manner, or to optimise their distribution to cloud environments and the local premises. Existing modelling approaches have a different understanding on how to model the aspects of cloud application systems and target cloud environments; and to which level of detail[208]. This dissertation might

[208]cf. Bergmayr et al. (2018), p. 1

need to select the necessary mechanisms and tools to model cloud environments in order to consider the differences between them and whether they fulfil the needs of organisations when migrating their application systems.

4.4.1 Different contributions to modelling cloud environments

Some research contributions to cloud modelling, such as the Cloud Modelling Framework,[209] (CloudMF) used their domain-specific modelling language, CloudML, to enable cloud-based dynamically adaptive application systems. Cloud environments modelling has got the potential of working as an additional input model for the decision-making process supported with this dissertation.

Researchers have applied CloudMF to facilitate provisioning, deploying, and adapting software to run on multi-cloud application systems for Internet of Things and enterprise systems[210]. Likewise, the scope of this dissertation includes assessing the migration of different kinds of application systems to cloud environments and the local premises. Different kinds of application systems in terms of their functionalities, software architectures, capabilities, and fields of application. Therefore, the contribution of this dissertation will be evaluated in scenarios pertaining to different disciplines: PaaS offerings (Section 7.2 in Chapter 7), Business Intelligence (Section 7.3 in Chapter 7), and the Internet of Things or cyber-physical systems (Section 7.4 in Chapter 7).

The introduction of this section on cloud environment modelling mentioned the different level of detail to which modelling approaches go. The PaaSage project transformed cloud provider-independent models into cloud provider-specific models enriched with metadata tailored to that provider using CloudMF[211]. Figure 4.5 shows the ports and bindings between three artifacts using the requirements and communication ports. A requirement port models the requirement of a capability (for example a Java-based component that requires an orchestration engine)

[209] To continue tackling the challenges that Remics project left untouched, the FP7 project PaaSage extended the work to account for to include the modelling of the target cloud environments into their methodology and tools: cf. Sáez et al. (2018), pp. 19–20; Woitsch and Utz (2015), p. 128; Brandtzæg et al. (2012), pp. 213–214; Ferry et al. (2013b), p. 892

[210] Researchers contributing to different projects —Remics , PaaSage, and MODAClouds— used CloudMF and CloudML: cf. Ardagna et al. (2018), pp. 11–24; MODAClouds Consortium (2012), p. n/a, URL see Bibliography

[211] cf. Ferry et al. (2018), pp. 20–21; Ferry et al. (2013a), pp. 38–40

whereas a communication port specifies the port to provide that service (for example, the specific port of the orchestration engine). Additionally, artifacts possess two kind of bindings deployment dependencies establishing the deployment order of those artifacts (for example, the orchestration component comes first and then the java-based component) and communications channels to determine how an artifact communicates with another one (for example, the components use the port that the orchestration engine offers and communicate using JSON). The cloud provider-specific takes these ports and bindings in the provider-independent model to specify the actual solution fulfilling the application system requirements.

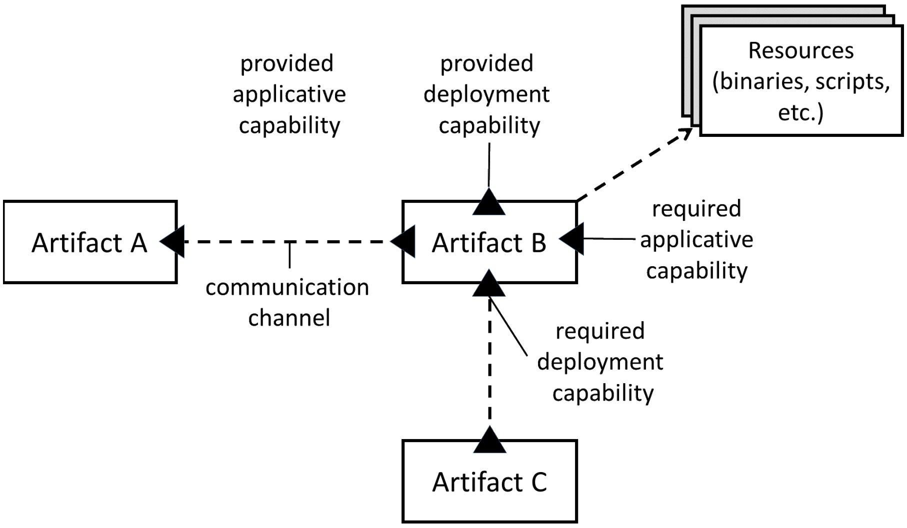

Figure 4.5: Cloud provider-independent model based on Ferry in 2013[212]

The research work explained with Figure 4.5 coupled selecting a viable cloud provider with performing software adaptations tailored to the specific cloud migration project[213]. The use of model-based techniques and tools aims at making a dynamically adaptive application system (DAS) out of cloud-based application systems. Dynamically adaptive application systems provide continuous application systems design and adaptations oblivious to the underlying software, platform, or infrastructure, which structures the software stack of the cloud computing environments. The proposed concept focuses on the design time of the cloud migration as this belongs to the scope of this dissertation. Section 8.4 in Chapter 8 envisions the future work to extend the scope of this dissertation (hence out of the scope

[212]Own illustration based on cf. Ferry et al. (2013b), p. 892 and Ferry et al. (2018), pp. 11–13 to describe the ports and bindings of several artifacts in a cloud provider-independent model that the provider-specific model could specify

[213]cf. Ferry et al. (2018), pp. 11–13,20–21; Ferry et al. (2013b), pp. 887–890

of this dissertation) by integrating techniques of dynamically adaptive application systems. The extension would be to adapt the target cloud-based application systems at run time according to varying cloud migration criteria.

Research works on vendor lock-in also motivated using two abstraction levels to specify in both a provider-independent and -specific abstraction levels in the proposed concept[214]. This way, cloud developers can design and implement their application systems with a paradigm and technology in mind while keeping the provider-specific implementation details at an agnostic level[215]; hence minimising vendor lock-in issues. The cloud migration concept assesses the fulfilment of the cloud migration criteria at a provider-independent level but also allows for defining metrics to estimate how provider-specific characteristics affect the cloud-migrated application system. Additionally, the proposed concept allows for exchanging the software methods or components that deliver provider-specific metrics for particular cloud migration criteria.

Cloud environments modelling based on feature models

Given this analysis of existing literature and of the state of the art together with the informal research experiments performed; the use of feature model-based cloud environments modelling approaches that use variability points emerged as a viable solution. Deepening into the research field of these modelling approaches exceeds the scope of this dissertation other than using them, integrating, and extending as deemed necessary for its application to cloud migration support. The decision taken is to seamlessly integrate a cloud environments modelling approach into the proposed concept in such a manner that it could be replaced with a different one. Therefore, Section 4.4 presents above the different modelling contributions and their characteristics to make the integration decision based on the thorough study of the options presented below.

Cloud environments modelling based on feature models use that kind of variability modelling to capture the commonalities and differences in application system

[214] These research works use domain-specific languages aiming for re-usability they use different abstractions with ontological and linguistic instantiations implementing the type-instance pattern and offering different components bound through their exposed ports or interfaces: cf. Lara et al. (2015), pp. 454–455; Atkinson and Kühne (2002), p. 290; Kühne (2006), p. 369; Szyperski et al. (1999), p. 184; Atkinson and Kühne (2003), pp. 39–41

[215] cf. Ferry et al. (2018), pp. 11–13; Ferry et al. (2014), pp. 269–270

families[216]. Here the application system families considered are the cloud environments. Variability modelling, and feature models in particular, allow for representing several application system configurations in a single model. Hence, the proposed concept could use it to consider the different cloud-based architectures of an application system to support the cloud migration decision making.

Cloud-oriented feature models can model cloud computing environments and their features to the desired level of detail. The proposed concept could integrate them to integrate the details of the cloud environments into the decision support of partial (or full) cloud migration projects fulfilling the cloud migration criteria. The approaches based on feature models mentioned above follow the same principle explained in Figure 4.5 explains and allow for cloud-specific and provider-independent modelling.

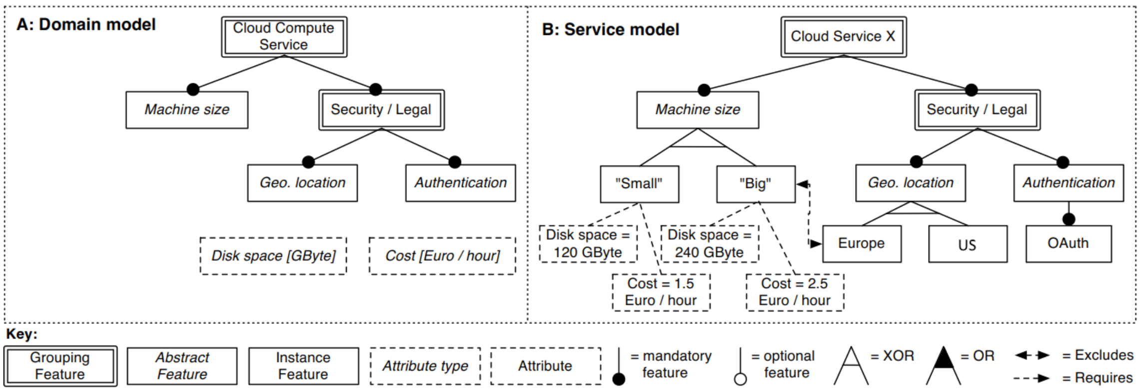

Figure 4.6: Example of a cloud environment modelling approach by Wittern in 2012 candidate for integration in the cloud migration concept[217]

Figure 4.6 shows an example of two model types describing cloud environments at an abstract and provider-specific levels, which feature the different modelling options provided by this approach[218]. The domain model in Figure 4.6 A describes a Cloud Compute Service (cloud environment configuration as proposed by this dissertation) as a feature grouping two abstract features: the quantitative Machine size and the qualitative Security/Legal. The domain model describes the abstract options a cloud environment offers as a blueprint.

[216]cf. Wittern and Zirpins (2016), p. 560; Wittern et al. (2012), p. 127; CloudMF: Quinton et al. (2013), pp. 21–23; Ferry et al. (2013a), pp. 38–40; Benavides et al. (2005), pp. 138–140

[217]Illustration by Erik Wittern on the framework that the proposed concept could integrate and use to model the cloud environments cf. Wittern et al. (2012), p. 131; Wittern and Zirpins (2016), p. 560

[218]cf. Wittern and Zirpins (2016), pp. 557–560; Wittern et al. (2012), pp. 138–140; Karataş et al. (2010), p. 286

The service model, instead, specifies the domain model. The service model in Figure 4.6 B describes the Cloud Service X grouping feature offered by a specific cloud provider with two instance features —"Big" and "Small"— for the machine size differing in their Disk space and Cost. Cloud providers instantiate the abstract features to provide different services. They represent variability points leading to multiple instances of that specific feature. The XOR relationship in the service model forces using one of the two Geo.location (Europe or US) except when violating the cross-tree relationship that specifies that the Europe Geo. location does not offer offers the "Big" instance.

Figure 4.7 visually describes how to model an exemplary cloud environment configuration, the CloudA Feature Model, with feature models using a hierarchy of features and ontologies applied to cloud environments modelling[219]. In this approach, feature models define the commonalities and variabilities of cloud environments (cloud offerings) and ontologies represent the set of existing cloud environments.

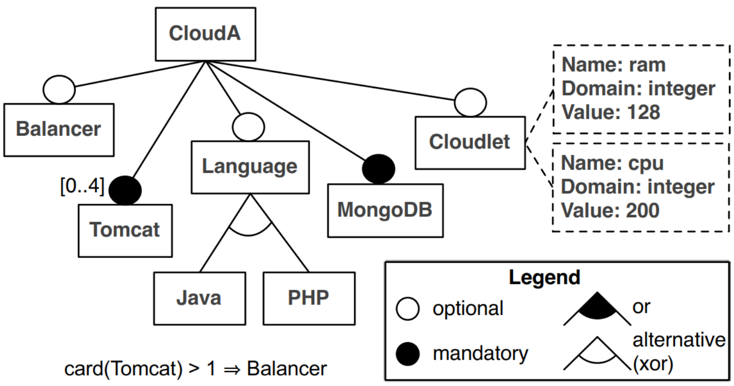

Figure 4.7: Example CloudA Feature Model of a cloud environment modelling approach by Quinton in 2013[220]

Mandatory features express commonalities in Figure 4.7 such as having from zero to four instances of the open-source web server Tomcat or the requirement of providing a MongoDB NoSQL database[221]. Meanwhile, optional features capture the variability that does not dictate having a balancer or a cloudlet and not restricting

[219]cf. Quinton et al. (2016), pp. 58–68; Quinton et al. (2013), pp. 1–3

[220]Illustration by Clément Quinton on the framework framework that the proposed concept could integrate and use to model the cloud environments: cf. Quinton et al. (2013), p. 24; Quinton et al. (2016), pp. 58–68;

[221]Apache Tomcat is a web server cf. Apache Software Foundation (2023), p. n/a, URL see Bibliography; and MongoDB is a free and open-source cross-platform document-oriented database MongoDB, Inc. (2022), p. n/a, URL see Bibliography

the language to neither Java nor PHP due to the XOR relationship included. Feature models result as a combination of feature diagrams and the associated constraints modelled with implies or excludes constructs and feature-to-feature dependencies[222]. CloudA could run several instances of Tomcat requiring load balancing; therefore, a constraint —$card(Tomcat) > 1 \rightarrow Balancer$— uses the card() function to query the cardinality of Tomcat instances to imply the use of a Balancer if it is greater than one[223].

4.4.2 Take on cloud environments modelling

The analysis of the presented feature model-based concepts and tools[224] revealed that the tools are easy to use to model existing cloud environments and their offerings and very easy to integrate in other tools or prototypes —such as an implementation of the concept proposed in this dissertation in a prototypical fashion— given their implementations based on the Eclipse Modelling Framework (EMF) in open source projects. The easiness of use of such modelling mechanisms and tools show the potential to facilitate modelling cloud environments and their variability; that is, their cloud environment configurations[225].

The proposed concept could use feature models to include the details of cloud environments into the decision-making process and generate alternatives. cloud migration alternatives that couple the cloud service descriptions and the input models that describe the on-premises application system modelling and the cloud migration criteria[226].

[222]cf. Quinton et al. (2016), pp. 76–77; Quinton et al. (2013), pp. 12–16

[223]Clément et al. extended feature models with feature cardinalities and attributes related to the features cf. Quinton et al. (2016), pp. 58–68; Pohl et al. (2005), pp. 91–114; Cardinality-based feature models use an interval $[m..n]$ with m and n as lower and higher bound respectively Czarnecki et al. (2005), p. 7; Attributes extend the information that the standard notation of feature models provide to offer feature attributes as a triplet $< name, domain, value >$ Benavides et al. (2005), p. 491; Gruber (1993), pp. 199–200; Corcho et al. (2006), pp. 1–3, 32–36; Ga et al. (2009), pp. 352–358

[224]cf. Leitner et al. (2019), pp. 340–344; Wittern et al. (2012), p. 127; CloudMF: Quinton et al. (2013), pp. 21–23; Ferry et al. (2013a), pp. 38–40

[225]This could even lead to it being used by the cloud providers (although the target user group now are organisations migrating their application systems as cloud consumers) to model their own offerings considering a very optimistic outcome whereby the proposed concept becomes a relevant tool for cloud selection in the industry

[226]cf. Leitner et al. (2019), pp. 353–356; Wittern et al. (2012), p. 127; Quinton et al. (2013), pp. 21–23

4.5 Cloud and inter-cloud architectures

Cloud architectures, and inter-cloud architectures in particular, that use brokering strategies for multi-cloud application systems are fundamental to deploy software components distributed to both the local and cloud premises; therefore, this section is fundamental to this dissertation. Inter-cloud deployments come in four architectures of volunteer federation with independent architectures[227]: centralised inter-cloud federations, peer-to-peer inter-cloud federations, multi-cloud services, and multi-cloud libraries. Figure 4.8 puts together all four.

The process for this literature review explored concepts for cloud and inter-cloud architectures searching from 2012 to 2023 specific keywords in ACM digital library, IEEE Xplore, ScienceDirect, SpringerLink, and Google scholar: inter-cloud, peer-to-peer inter-cloud, centralised inter-cloud, cloud federation, multi-cloud service, and multi-cloud service. These keywords were selected based on own experience and literature knowledge.

The literature found distinguishes between volunteer federations and independent inter-cloud architectures[228]. In volunteer federations a group of cloud providers voluntarily collaborate sharing their resources while independent inter-cloud architectures let an application system or its broker aggregate the resources of several cloud providers. Independent inter-cloud architectures or multi-cloud deployments have a service use multiple cloud providers which are independent from each other. Therefore, these providers do not necessarily interconnect and share their infrastructure[229]. Hence, multi-cloud clients or their brokers and middleware usually provision themselves with resources that they schedule ad-hoc[230]. Independent inter-cloud architectures include the multi-cloud services and libraries.

[227] cf. Varghese and Buyya (2018), p. 851,861–865; Grozev and Buyya (2014), p. 369; IEEE-SA (2017), p. n/a;
[228] cf. Varghese et al. (2016), pp. 170–176; Ferrer et al. (2012), p. 66; Rochwerger et al. (2009), p. 10
[229] cf. Singh et al. (2016), pp. 200–209; Toosi et al. (2014), p. 1
[230] cf. Hamdaqa et al. (2015), pp. 26–31; Rochwerger et al. (2009), p. 1
[231] Own illustration based on cf. Grozev and Buyya (2014), p. 374 and Varghese and Buyya (2018), p. 851. The proposed concept learns from these four deployments —two with inter-cloud federation and two as multi-cloud deployments— to suggest appropriate cloud deployments to accommodate the requirements of the organisation cloud migrating an application system.

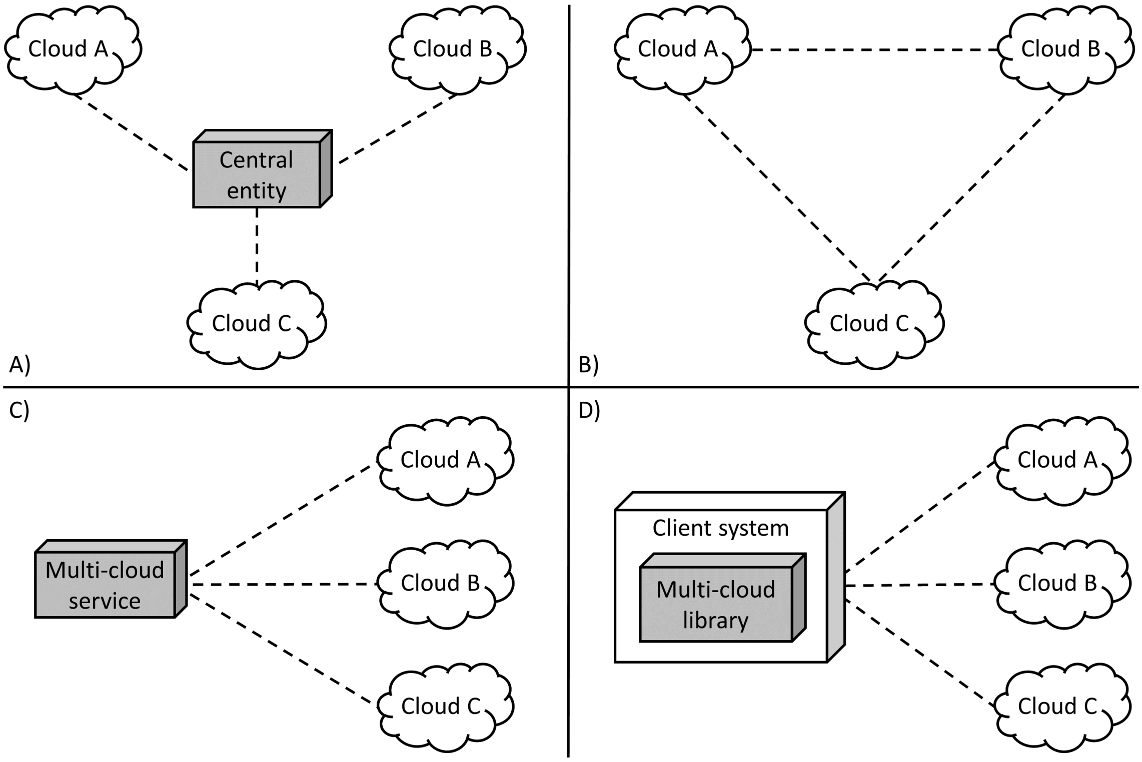

Figure 4.8: Four architectures for inter-cloud development inspired by Grozev et al.: A) centralised inter-cloud federation, B) peer-to-peer inter-cloud federation, C) multi-cloud service, D) multi-cloud library[231]

4.5.1 A) Centralised inter-cloud federation

As the scope of this dissertation targets assisting organisations in moving their application systems to cloud environments and their local premises; this dissertation benefits from being informed about the use of centralised federated inter-cloud deployments[232]. Some research works proposed sharing resources from different cloud providers across their limits. They envisioned the Cloud Coordinator to share resources (across cloud providers) through their marketplace (the Cloud Exchange) in a centralised manner so that the members of the federation —that is, cloud providers— offer their resources to other providers or brokers[233]. Location-aware brokering let cloud consumers define SLAs with a mediating cloud provider or directly consume resources listed in the marketplace. The relevance of this marketplace-oriented perspective inspired and motivated evaluating this dissertation and the proposed cloud migration concept in the cloud migration to PaaS offerings scenario presented in Section 7.2 of Chapter 7. This evaluation scenario

[232] InterCloud project: cf. Noël et al. (2015), pp. 4222–4228; Buyya et al. (2010b), pp. 18–21
[233] cf. Qu et al. (2018), pp. 9–16; Costanzo et al. (2009), p. 24; Cloud Coordinator and Cloud Exchange: Calheiros et al. (2012), p. 1350

analyses the potential of, in pursuit of higher levels of exploitation, conforming win-win strategies to bring PaaS consumers and providers together to service a marketplace of PaaS offerings.

A centralised composite entity (Federation Runtime Manager) acts as single entry point to the federation which the Contrail project proposes[234]. Their broker maps cloud consumers' requests to the appropriate cloud resources if the cloud providers offering them either: implement internal adapters if the cloud providers run their platform or external adapters for those which do not, which is a more realistic possibility. Their work influenced the evaluation of the concept proposed in this dissertation for Platform-as-a-Service offerings adoption. Likewise, it motivates including in this dissertation mechanisms to model detailed metadata of cloud environments, such as the available geographical locations and the legislative data location, as their differentiating aspects in addition to the cost and performance optimisation heuristics[235].

Federating cloud providers so that cloud consumers can seamlessly switch from one to another prevents vendor lock-in[236]. Cloud providers can achieve this offering open-source software stacks based on standardised interfaces[237]. Implementing standard interfaces help support and enforce SLAs at the federation and node level for each resource type[238]. The proposed concept incorporates into the decision making the fact that cloud providers offer value-added services; especially, at the PaaS level such as the persistence layer of Contrail[239]. Their cloud platform envisions feature-rich PaaS offerings on top of a federation of IaaS providers[240] that migrate running application system between providers without

[234] cf. Carlini et al. (2016), pp. 4–5; Contrail-Consortium (2014b), p. n/a, URL see Bibliography; Contrail Consortium (2011), p. n/a

[235] cf. Carlini et al. (2016), pp. 1–3; Carlini et al. (2012), p. 159

[236] Ibid. Contrail, as well as many other research and industrial projects, inspired including in the scope of this dissertation the vendor lock-in problems across the entire cloud stack as Section 6.6 in Chapter 6 explains and evaluating vendor lock-in in the context of the Platform-as-a-Service scenario (as Chapter 7, describes in Section 7.2)

[237] cf. García et al. (2016), pp. 19–22; Harsh et al. (2012), p. 435

[238] cf. García and Blanquer (2015), p. 35

[239] cf. Wang et al. (2015), pp. 321–323; Blasi et al. (2013), p. 146; Coppola et al. (2012), p. 1–3; Contrail Consortium (2012), p. 16

[240] cf. Morin and Cascella (2014), pp. 33–36; Out of the scope of this dissertation, Contrail implemented the map/reduce programming model for large data sets processing and allowed for autonomic workflow execution using a Hadoop-based genome assembler, which belongs to the future work of this dissertation as it escapes the scope of this dissertation: Li et al. (2013),

disconnecting the application system. Hence, live migration, which exceeds the scope of this dissertation as it focuses on supporting the cloud migration at design time. Nevertheless, live migration mechanisms could use the proposed concept to plan the cloud migration.

A federation is a form of collaboration, some cloud market models composite services coming from cloud providers which dynamically collaborate[241]. The federation brokers application systems by using the registry of the services of the collaborating cloud environments based on the SLA contract, meta-brokering, cloud brokering, and on-demand service deployment[242].

Self-adaptable autonomous management of federated cloud management architectures optimises the resource utilisation while minimising the costs of respecting the SLAs[243].

4.5.2 B) Peer-to-peer inter-cloud federation

Peer-to-peer inter-cloud federation architectures, as Figure 4.8 B) shows, have cloud providers directly communicating and negotiating without any mediator. Peer-to-peer architectures allow cloud environments to directly negotiate with each other how they share resources; on occasion, with the help of an abstract layer working as a service manager on top of the cloud federation to execute federated services[244]. Cloud environments might use open standards to participate in

pp. 169–173; Contrail-Consortium (2014a), p. n/a, URL see Bibliography; Schmuck and Haskin (2002), p. 1. They addressed Hadoop architecture but not BigTable as initially planned Chang et al. (2008), p. 1; Likewise, they offered secured storage and security mechanisms to let the federation and end users authenticate and authorise them with the appropriate set of rights while keeping virtual machines isolation: Stender et al. (2012), p. 267; Lazouski et al. (2012a), p. 202; Lazouski et al. (2012b), p. 79

[241] cf. Truong-Huu and Tham (2014), pp. 251–256; Hassan and Huh (2010), pp. 432–433; Yamazaki (2004), pp. 13–14

[242] To minimise conflicts when negotiating how to collaborate, auction policies let cloud providers publish joint bids to agree on the resources and services other cloud providers provide and who covers the collaboration costs —such as transmissions, network establishment, or capital flow— cf. Na and Huh (2015), pp. 1321–1327; ; Nepal et al. (2007), pp. 82–84; Nepal and Zic (2008), pp. 283–285; Hassan et al. (2009), pp. 9–11; Some research works map the service requests to the fitting virtual machines with the use of standard stateless web services described with the Web Services Description Language (WSDL) Klusch et al. (2016), pp. 139–147; Christensen et al. (2001), p. n/a, URL see Bibliography

[243] cf. Salama and Bahsoon (2015), pp. 844–848; Kecskemeti et al. (2012), p. 583

[244] cf. Cohn et al. (2018), pp. 2–7; Rodero-Merino et al. (2010), p. 1226

the federation and make them interchangeable[245]. Open standards[246] like the Open Cloud Computing Interface (OCCI) or the Open Virtualisation Format (OVF), which help the federation of cloud providers avoid vendor lock-in and scale out or up according to trigger-action rules for transparent and location-agnostic elastic application brokering[247].

The definition of service manifests for peer-to-peer inter-cloud federations in the Reservoir project used the Essential Meta-Object Facility (EMOF) to describe the structure of metadata specifying service application provision and OCL directives (Object Constraint Language) to capture the constraints and semantics[248]. OCL-based constraints are interesting to this dissertation as they can be used to model the target application system prior to migration and its requirements. Existing research suggest checking in the cloud-enabled architecture against the applicable SLA defining performance objectives (such as response time or number of needed virtual machines)[249].

Open research test beds[250], let research on peer-to-peer inter-cloud federations to design, provision, and manage federated services while also letting cloud consumers directly use a cloud provider[251]. Some publications by the Optimis project

[245] The Reservoir project —Resources and Services Virtualisation without Barriers— proposed peer-to-peer inter-cloud federations with the assumption that no matter how large it is, no cloud provider meets the demand for cloud services: cf. Silva et al. (2018), pp. 409–412; Rochwerger et al. (2009), pp. 1–5

[246] Survey: García et al. (2016), pp. 19–22; OCCI: Metsch et al. (2010), p. 3; OVF: Crosby et al. (2010), pp. 9–11; Wang et al. (2012), p. 4; Certificate Authorities: Chokhani et al. (2003), pp. 1–94, URL see Bibliography; ISOC: ISoc (2023), p. n/a, URL see Bibliography; ICANN: ICANN (2021), p. n/a, URL see Bibliography; SASL standard Melnikov and Zeilenga (2006), pp. 1–33, URL see Bibliography; SAML standard: Cantor et al. (2005), pp. 6–27; Bernstein and Vij (2010c), p. 18; The IEEE Cloud Computing Initiative: IEEE TCCLD (2020), p. n/a, URL see Bibliography

[247] cf. Hadas et al. (2014), p. 8; Rochwerger et al. (2011), p. 44; 49–50

[248] EMOF is the OMG (Object Management Group) standard for model-driven engineering and OCL a declarative language part of the UML standard to describe rules applying to UML (Unified Modelling Language) models

[249] cf. Jennings and Stadler (2015), pp. 567–573; Toffetti et al. (2010), p. 66

[250] cf. Sinha and Shekhar (2015), pp. 171–174; The IEEE Intercloud Testbed: Bernstein and Demchenko (2013), p. 45; Bernstein et al. (2009), p. 328; Bernstein et al. (2011), p. 293; Bernstein and Vij (2010a), p. 431; Bernstein and Vij (2010b), pp. 35–36; Open Cirrus: Avetisyan et al. (2010), p. 35

[251] cf. Sinha and Shekhar (2015), pp. 174–176; Avetisyan et al. (2010), p. 35; Campbell et al. (2009), p. 1–2. Open Cirrus uses the Tashi cluster management software Kozuch et al. (2009), p. 43, the Hadoop framework Apache Software Foundation (2022b), p. n/a, URL see Bibliography, and the Zoni custom physical resources management Avetisyan et al. (2010), p. 38. The research test bed scheduled computing tasks to the nodes with the needed data persisted in them with the use of Tashi and the location metadata available through the use of the Hadoop Distributed File System Avetisyan et al. (2010), pp. 41–42

explained the market-related challenges and legislative obstacles that peer-to-peer inter-cloud federations face as well as the requirements of these federations[252]. Requirements for service life-cycle optimisation and adaptive self-preservation together with considerations on trust, risk, ecology, and cost for peer-to-peer federations. The project offered a brokering toolkit[253], which similarly to some of the cloud migration support described in Section 4.2, used a provider-agnostic abstraction (quite abstract without a fixed software architecture) to combine cloud providers in arbitrary and hierarchical architectures, which then become provider-dependent in later stages. The use of these two levels of abstraction is a lesson learnt for this dissertation. The toolkit supports covering the multi-disciplinary challenges that hinder the wider adoption of cloud environments; and, in that sense, resembles this dissertation's contribution but applied to peer-to-peer inter-cloud federations instead.

Their toolkit checks whether the infrastructure respects the agreed SLAs when delivering the service or else it migrate it somewhere else[254]. While this addresses on-line migration and goes beyond the scope of this dissertation, it could belong to the future work of this dissertation with the proposed concept supporting this decision on design time and another component enacting the live migration.

Commercial frameworks also implement the peer-to-peer federation concept with data brokers or stand-alone solutions. The framework manages dynamic service agreements and policies specifying which resources to share and how according to the agreed SLAs for the federated cloud architecture[255]. Mechanisms for accountability and control that could trigger a cloud migration assisted by the concept of this dissertation. The proposed could support experiments with cloud bursting —which is out of the scope of this dissertation although described ads a potential future work venture in Section 8.4 describes in Chapter 8— to accommodate

[252] cf. Gouvas et al. (2015), pp. 217–221; OPTIMIS Consortium (2013), p. n/a, URL see Bibliography; Ferrer et al. (2012), p. 66

[253] cf. OPTIMIS-Consortium (2022), p. n/a, URL see Bibliography

[254] cf. Jennings and Stadler (2015), pp. 599–606; Macias and Guitart (2012), p. 156; Li et al. (2008), p. 139; Ziegler (2012), p. 37; Gogouvitis et al. (2012), p. 35

[255] Existing frameworks manage the federation by focusing on the technical aspects of the cloud migration, while this dissertation also considers other cloud migration criteria such as those related to the cloud environments provider, the business aspects, or the levels of security the cloud providers enforce. Cloud migration criteria adapted to peer-to-peer inter-cloud federations, in addition to their interplay: cf. Xu and Palanisamy (2018), p. 4, 542–543; Gerrard et al. (2011), p. 364; McGough et al. (2010), p. 88; Arjuna (2010), pp. 7–9

the demand when it overpasses the in-house and designed cloud capabilities by utilising other cloud resources[256].

4.5.3 C) Multi-cloud service

Multi-cloud services let cloud consumers access cloud services at the levels of the infrastructure, platform, or software working as a service. Application brokers, such as the one the mOSAIC project proposes, are the part of the multi-cloud architecture that mediate between cloud consumers and providers as a brokering service to access the infrastructure of cloud providers[257]. The Platform-as-a-Service scenario, which Section 7.2 explains in Chapter 7, studies the effects of using application brokering in the cloud migration to PaaS deployments supported by the proposed concept.

The research done in the mOSAIC project proposes postponing the cloud selection decisions using their cloud ontology to define vendor-independent service requirements informs this dissertation[258]. A lesson learnt from their work is how do the project assists the cloud migration on the potential cloud migration alternatives that fulfil the organisation requirements[259]. Their cloud agency component register cloud providers that cloud consumers can discover through their API. After a final analysis on all viable cloud migration alternatives, the mOSAIC multi-agent brokering mechanisms search services matching the application requirements to find the best-fitting cloud services candidates to outsource computation to them[260]. The project consortium arguments that this could foster the competition between providers and lower entry barriers to the PaaS market for small cloud providers.

The Optimis project contributes to peer-to-peer cloud-based federations and also supports provisioning cloud services in interdependent inter-cloud federations. That is, multi-cloud deployments using their deployment engine and service optimiser to let cloud consumers launch and monitor services on multiple cloud

[256] cf. Arjuna (2010), p. 5

[257] cf. Moscato et al. (2017), pp. 467–471; mOSAIC Consortium (2013), p. n/a, URL see Bibliography; Private and public cloud environments with focus on avoiding vendor lock-in issues: Sandru et al. (2012), p. 333; AMICAS with automatic scaling in a vendor-agnostic ecosystem: Petcu et al. (2013b), p. 2443

[258] cf. Di Nitto et al. (2016), pp. 61–67; Petcu et al. (2013a), p. 1417

[259] cf. Moscato et al. (2017), pp. 482–485; Petcu et al. (2011), p. 1,8; Petcu et al. (2013b), p. 2443

[260] Ibid.

environments; if they incorporate the Optimis agents to their architecture[261]. An strategy to overcome this limitation, as most cloud providers would probably not adapt their environments to include an agent tailored to a not-widely adopted platform, one could reasonably argue that it would be beneficial to let external developers create independent adapters to each cloud provider. Nevertheless, it would also be necessary (and difficult) to give incentives to develop adapters for less popular environments. The Contrail project also faced this conundrum[262].

Research in cloud broker services proposes provisioning target application systems to multiple cloud environments in accordance to a set of application requirements and topology[263]. Pawluk et al. drew inspiration from the service measurement index and cloud metrics[264] to implement mechanisms to let cloud consumers specify their needed topology, requirements, constraints, and objectives. Their contribution informs this dissertation and could be extended with further cloud migration criteria, such as cost or vendor-specific cloud migration criteria, and with automatic and pseudo-automatic assessment of cloud migration criteria to assist the cloud migration. Some commercial cloud management systems help in configuring and deploying to multi-cloud service architectures[265]. The authors of some of these solutions claim they provide similar features to RightScale and also realise trigger-based up- and down-scaling as well as in- and out-scaling actions using alert-action mechanisms[266]. Cloud management systems usually differ in price, supported cloud providers, used technologies, and terminology[267].

[261] cf. Gouvas et al. (2015), pp. 217–221; OPTIMIS Consortium (2013), p. n/a, URL see Bibliography; Ferrer et al. (2012), p. 66

[262] cf. Carlini et al. (2016), pp. 4–5; Contrail-Consortium (2014b), p. n/a, URL see Bibliography

[263] cf. Anastasi et al. (2014), pp. 304–307; Pawluk et al. (2012), p. 891

[264] cf. Ortiz et al. (2018), pp. 555–557; The service measurement index Siegel and Perdue (2012), pp. 411–415 inspired Garg et al. and Zachos et al. to separately define software metrics: Garg et al. (2011), pp. 210–211; Zachos et al. (2011), p. 16

[265] RightScale (now Flexera): cf. Flexera (2023), p. n/a, URL see Bibliography; Enstratius (formerly enStratus): EnStratius, Inc. (2013), p. n/a, URL see Bibliography; and Scalr (open source): SCALR (2022), p. n/a, URL see Bibliography

[266] cf. Flexera (2017), p. n/a, URL see Bibliography

[267] Cloud management systems usually let organisations deploy virtual machines to public and private clouds through their console or management features: cf. Varghese and Buyya (2018), p. 851,861–865; Grozev and Buyya (2014), pp. 375–378; Tusa et al. (2010), p. 477

4.5.4 D) Multi-cloud library

Cloud consumers use multi-cloud (or cross-cloud) libraries to build the application brokers tailored to how they want to provision their application systems to multiple cloud environments. Inter-cloud libraries allow for scheduling workloads to cross-provider application components. Unified cloud APIs help cloud consumers uniformly interact with multiple cloud providers; accordingly, the cloud computing community have come up with several multi-cloud libraries for different languages[268].

Multi-cloud libraries follow the same principle of letting cloud consumers manage the selected cloud environments without knowing their actual implementation details. These libraries abstract from the particularities of managing each cloud environment while offering an API to let application systems provision themselves with cloud resources across different cloud environments providers and locations. To state their potential to be effectively used, one could use the argument that several projects and software solutions have used them[269] to build the cloud adapters they can use to develop application- and system-specific brokers to manage the underlying cloud infrastructures. Using those libraries is an effort-intense endeavour that does not relieve those platforms and projects from additional tasks; for example, replicating the broker to deliver better availability.

In a similar manner, the Stackato polyglot PaaS[270] intends to automatic configure the language runtime, web server, application dependencies, databases, and other services so as to select the appropriate hypervisor and public or private cloud environment.

4.6 Cloud migration criteria

Different criteria drive the decision organisations make on how to migrate their application systems to both cloud environments and their local premises. In this

[268] JClouds for Java: cf. JClouds Apache (2023), p. n/a, URL see Bibliography; SimpleCloud for PHP: IETF (2015), p. n/a, URL see Bibliography; the DeltaCloud for ruby: Apache Software Foundation (2015), p. n/a, URL see Bibliography; LibCloud for Python: Apache Software Foundation (2022c), p. n/a, URL see Bibliography; and Nuvem and Brooklyn: Apache Software Foundation (2021), p. n/a, URL see Bibliography; Apache Software Foundation (2022a), p. n/a, URL see Bibliography

[269] The already mentioned projects Optimis or mOSAIC as well as the upcoming Stackato

[270] cf. Stackato (2021), p. n/a, URL see Bibliography

dissertation, these criteria are decision-making factors, which are defined as cloud migration criteria and classified in the four groups below: the business and economic, technical, provider-related, and security and privacy cloud migration criteria.

The process for the literature review on cloud migration criteria explored concepts related to cloud migration criteria by searching specific keywords from 2012 to 2023 in ACM digital library, IEEE Xplore, ScienceDirect, SpringerLink, and Google scholar: cloud computing, cloud migration criteria, cloud adoption, cloud migration, and criteria for cloud selection. Those specific keywords were selected based on own experience and literature knowledge. The process involved moving down from the broad cloud migration criteria to specific instances of sub-criteria within the business and economic, technical, provider-related, and security and privacy cloud migration criteria. The increasing level of detail come from exploring the specific literature per category and then searching based on the keywords found in that literature.

Given that the use of cloud computing entails an important change in how service providers offer those services so that consumers use them[271]. Practitioners usually consider both the potential benefits to technically improve modern data centres and the risks of migrating to cloud-based deployment environments. The literature analysed in this chapter shows how researchers illustrate the real-life implications and challenges of the migration of existing application systems to cloud environments. Researchers often include case studies or scenarios with different approaches focusing either on the business- and economics-related issues, the technical challenges, the provider-related implications of migrating to cloud environments, and security and privacy. Sometimes, they even mix and address, with varying depth, a subset of those concerns and how they affect one another. This dissertation works with the reasonable assumption that the understanding of these concerns and the approaches to cope with them might help organisations and companies to complete the efficient migration of their application systems to cloud environments. One can reasonably assume that the cloud migration criteria could be appropriate drivers of the decision making

[271] cf. Marinescu (2017), pp. 92–101; Xia et al. (2015), pp. 342–348; Fox et al. (2009), p. 19

of these organisations but with some sort of assistance due to the fact that they might depend or have an effect on each other. Organisations might need to understand and define their trade-offs between overlapping cloud migration criteria. Thus, the cloud migration decision is a multi-dimensional decision with intertwined decision-making criteria.

The reasonable assumption of this dissertation is using the cloud migration criteria detailed below to support the cloud migration decision. The analysis of this collection of concepts and knowledge is fundamental to abstract from these particular research works and to elicit the requirements that Chapter 5 lists after the analysis of the state of the art in Section 5.1.

4.6.1 Business- and economics-related cloud migration criteria

Industrial organisations are of course very concerned with the business implications of adopting cloud computing as their computational environment including aspects related to economics. Table 4.2 summarises the business-related cloud migration criteria. The rest of this section explains the most relevant works across these cloud migration criteria starting off with some considerations about conflicting criteria.

The explanation of conflicting criteria for different stakeholders comes in addition to the proposed targeted cloud consumers and on how to model interdependent business and economic cloud migration criteria.

Further, a description of research works and business practices follow to describe the implications in terms of cost of outsourcing more of the IT services of an organisation and the costs of running cloud-based architectures. This study includes how to compare these costs to the costs of running application systems in in-house computing infrastructures versus a partial migration architecture while considering the effects on cost of selecting one cloud environment over another one. Finally, this section describes the effects of cloud migration for the innovation as well as how it helps accommodate demand behaviour following contracts aiming for business agility according to different pricing schemes. Pricing schemes like the soft reservations researched in this dissertation[272].

[272] See Section 6.8

Conflicting criteria. Organisations identify the potential technical benefits of adopting cloud environments as the environment to run their application systems but they put these aspects into its business-related context to shape their decision of whether to migrate their application systems and how to do it[273]. Marston et al. focused on the business-related issues associated with migrating existing application systems to cloud environments opposed to analysing the technology itself. They analysed cloud providers from the practitioner's viewpoint and identified three different stakeholders —the cloud consumer, provider, and regulator— and what their concerns are. The focus of this dissertation still remains on the cloud consumer and leaves focusing on the other two stakeholders for the future work of this dissertation, hence out of the current scope. Thereby, it would be interesting studying conflicting criteria that are beneficial for one stakeholder and a risk for another one. That is the case of vendor lock-in that while it is a risk for a cloud consumer it might be more interesting for a cloud provider. Section 1.3 in Chapter 1 explains to more detail a particular flavour of these stakeholders that include organisations that decide to migrate their application systems to cloud environments acting as cloud consumers or actually consumers or users of a cloudified application system. The study of the economics of migrating application systems to cloud environments sometimes focuses on the effects on the strategy of enterprises, their policies, and the regulations that affect application systems. Their recommendations promted the inclusion of the business aspects they mention in the cloud migration criteria —in addition to the other cloud migration criteria that the next sub-sections explain— to reduce the uncertainty on how the cloud computing business environment will behave after the cloud migration.

Interdependent criteria. A lesson learnt from the study of how criteria are interdependent is that cloud migration criteria of organisations cannot be considered in isolation. Accordingly, organisations could translate their business objectives into modelled interdependent cloud migration criteria that accommodate their requirements and that help assess costs and business-related criteria in cloud-migrated application systems in a similar manner as the conceptual framework of Klems as Figure 4.9 shows. The framework inspires the proposition of potential taxonomies

[273]cf. Chang et al. (2016a), pp. 24–27; Marston et al. (2011), p. 180

of business- and economics-related criteria in this dissertation that inform the proposed cloud migration concept. Additionally, it can support organisations to model themselves a taxonomy of the cloud migration criteria interesting for them from a business-related and economic standpoint.

Cost. Along similar lines, Khajeh-Hosseini et al. presented the *Cloud Adoption Toolkit* as a collection of tools to support the decision-making process involved in the adoption of cloud computing as the execution environment for application systems[274]. These research works highlighted the potential economic challenges and business risks or implications that industrial professionals usually take into account before the migration of an existing software application system to cloud environments. The scope of this dissertation includes incorporating business-related cloud migration, which are derived from the analysis of the related work in this section, to assist the cloud migration decision.

Cost and outsourcing. Cloud computing environments represent a significantly cheaper alternative to purchasing and maintaining in-house application system infrastructures in addition to the potential to eliminate many support-related issues since there is no physical infrastructure to maintain and organisations outsource the infrastructure support. This dissertation pays attention to findings of their case study; that is, that despite the advantages of outsourcing some of their functions, organisations could also consider the socio-technical implications of migrating their IT application systems to cloud environments. This is one of the components of this dissertation, which understands the migration as a multidimensional decision that depends on different interdependent aspects and that includes business- and provider-related concerns. For example, this dissertation learns from their case with Amazon[275] and generalise their approach to assess the costs of migrating an application system to cloud environments and how this reduces the application system infrastructure cost.

The following table compares the list of business-related cloud migration criteria with its different sub-criteria including considerations about its conflicting criteria.

[274] cf. Allan et al. (2016), pp. 14–17; Jennings and Stadler (2015), pp. 567–573; Khajeh-Hosseini et al. (2010b), p. 453

[275] cf. Chang et al. (2016a), pp. 24–27; Khajeh-Hosseini et al. (2010b), p. 451; Koehler et al. (2010b), p. 16

Table 4.2: Overview of business- and economics-related cloud migration criteria

Criteria	Description
Cost	Costs of running computation on the organisation premises, migrating costs, running costs on cloud environments, compensation costs or income, software adaptation or re-design costs. Cost efficiency and opex versus capex[276].
Innovation	Potential to disruptively innovate as organisations shift resources as they optimise how they operate and focus on their core business[277].
Demand behaviour	Seasonal or expected demand and unexpected demand or demand resulting of success. Control and predictability of IT costs[278].
Contract	Contract commitment duration and flexibility to change or break the contract. Including the contract clauses that intend payments upon outages or SLA violations. Which sort of application systems do organisations provide including B2B, B2C, enterprise[279].
Business agility	Short time to market software solutions, ability to accommodate increases in demand, reducing complexity, fostering innovation, increased collaboration, flexibility, and business continuity[280].
Pricing scheme	Flexible alternatives such as tiered, per-unit, per-usage, and subscription-based pricing schemes in a more simple and transparent manner in addition to flat rates, one-time purchases, and inner solutions. Including soft reservations or AWS Spot instances[281].
Payment method	Flexibility to pay with different methods[282].
Core business	Ability of organisations to focus on their key competences rather than maintaining data centres and IT infrastructures[283].

Cost compared to in-house computing. Important to this dissertation is that existing research found that many interdependent dimensions and factors affect the

[276] cf. Allan et al. (2016), pp. 14–17; Karunakaran et al. (2015), pp. 582–604; Klems et al. (2008), pp. 114–116; Hajjat et al. (2010), p. 247

[277] cf. Sirianni et al. (2019), pp. 84–87; Chang et al. (2016b), pp. 158–160

[278] cf. Morgan and Conboy (2018), pp. 297–313; Motahari-Nezhad et al. (2009), p. 7

[279] cf. Chang et al. (2016a), pp. 24–27; Janssen and Joha (2011), p. 85; Marston et al. (2011), p. 180; Khajeh-Hosseini et al. (2010b), pp. 451–453; Motahari-Nezhad et al. (2009), p. 7

[280] cf. Gholami et al. (2017), pp. 105–113; Phaphoom et al. (2015), pp. 167–172; Khajeh-Hosseini et al. (2010b), pp. 451–453, 456–457; Hauck et al. (2010), p. 5

[281] cf. Allan et al. (2016), pp. 14–17; Klems et al. (2008), pp. 114–116; Koehler et al. (2010b), p. 16; Armbrust et al. (2010), p. 56; Armbrust et al. (2010), p. 55; cf. Mohammadi et al. (2013), pp. 228–129

[282] cf. Haug et al. (2016), pp. 292–303; Phaphoom et al. (2015), pp. 169–173; Koehler et al. (2010b), p. 16

[283] cf. Gholami et al. (2017), pp. 110–112; Zatonatska and Dluhopolskyi (2019), pp. 45–49; Carr (2008), p. 67

migration decision depending on the requirements of the organisation migrating an application system. Markus Klems et al. estimate the value of running an application system on a cloud-based architecture opposed to a in-house computing. Decision makers might be assisted in their business-related decision to incorporate how Markus Klems et al. estimated the costs of migrating a business-to-business (B2B) and a business-to-customer (B2C) solution to cloud environments as well as to help compare these costs to the cost of conventional IT solutions[284].

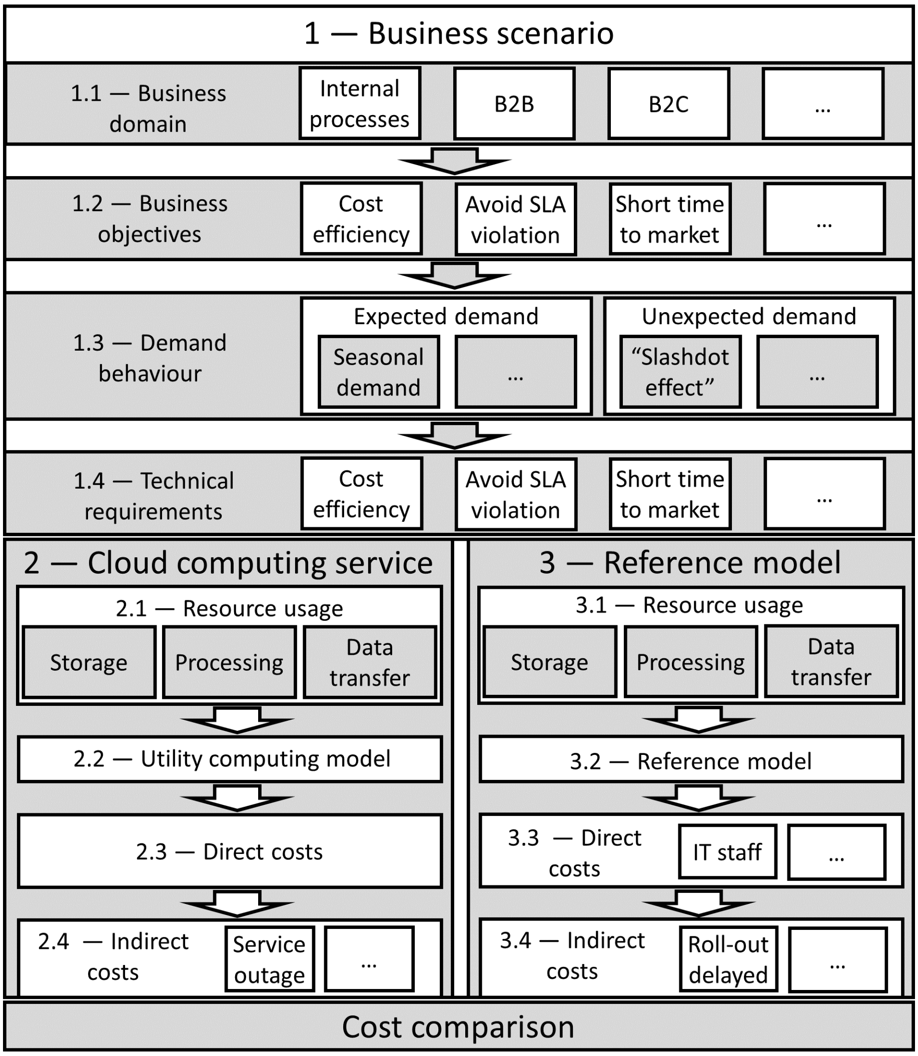

Figure 4.9: The framework by Klems et al. informs the proposed concept for the creation of multi-level taxonomies of business cloud migration criteria[285]

[284] cf. Haug et al. (2016), pp. 292–303; Phaphoom et al. (2015), pp. 169–173; Klems et al. (2008), pp. 114–116; Koehler et al. (2010b), p. 16; Similarly, Thanos et al. studied the business purposes of adopting grid computing Thanos et al. (2007), pp. 14–15

[285] Own illustration based on Klems et al. (2008), p. 112. The framework by Klems informed this dissertation on how to estimate cost in cloud-based architectures and how to include business-related cloud migration criteria in the proposed cloud migration DS concept and their interdependencies with the rest of cloud migration criteria

Cost and cloud selection. It can be reasonably assumed that weighing the technical implications (for an organisation) of adopting cloud environments as their computing infrastructure go deeper than just understanding the technical effects of the migration but it includes considering how organisations might trade some requirements off with others. Such is the case for the business-related cloud migration criteria. This dissertation takes the reasonable assumption stating that interdependent requirements affect each other in the cloud migration case as confirmed by other research[286]. Technical decisions are mutually reliant and very much affect the overall cost to, for example, re-design the target application so as to adapt it to the target computing environment. The selected cloud environment determines the adaptation effort in software engineering terms and hence affects the migrating costs. According to the mentioned research, assisting the cloud migration might require to a lower level of abstraction to estimate the technical characteristics of the migrated application based on the target application requirements and particularities. This dissertation learns from the above-mentioned research works that the cloud selection greatly affects the cost and migration decision, as this chapter explains. It could be reasonably assumed that both the cost estimation and migration decision depend on the accurate modelling of the potential target cloud environments; thus, this dissertation argues including cloud environment models into the decision-making process.

Cost and partial cloud migration. Similarly to the scope of this dissertation, Cloudward bound took a multidimensional approach than the mentioned research works with a technical focus including costs in the evaluation of the cloud migration decision and concepts that brought about new computing paradigms[287]. Cloudward bound included the economic viewpoint and studied how the overall migration cost is the result of a complex combined effect of the application characteristics in terms of workload intensity, storage capacity, growth rate, and the software licensing costs. Hajjat et al. argued for the horizontal partitioning between in-house and cloud-based deployments for a broad range of software application systems in order to cope with sensitivity concerns. Given the fitness of their research work

[286] cf. Phaphoom et al. (2015), pp. 167–172; Hauck et al. (2010), p. 5; Juan-Verdejo et al. (2014b), pp. 467–468
[287] cf. Villari et al. (2016), pp. 76–80; Hajjat et al. (2010), p. 247

for the purpose of this dissertation, this dissertation assumes it might be beneficial to design deployment models like this, which have software components partially running on the cloud environment and on the local premises, and add them into the cloud migration decision. Based on them, organisations could be assisted to consider the technical implications on other application system requirements, business objectives, and cloud migration criteria. Additionally, a lesson learnt is that it could be interesting to extend the application of cloud migration support to partially migrated cloud migration application systems, which partially run on private cloud offerings.

Innovation. Using cloud environments as the new enterprise data centres has the potential of promoting disruptive innovations as organisations overcome the minutia of daily operations and drive real business innovation instead[288].

Demand behaviour. It influences how to architect an application system if the demand workflow depends on the application system behaviour[289]. This includes considering whether there are unexpected spikes of demand with temporary effects or on a seasonal demand fluctuations —for example on Christmas or at the end of the year— and whether organisations offer their services for a specific period of time or the existence of batch processing tasks.

Contract. This dissertation is informed by the study of the business objectives, the demand behaviour of the target application system, and the specification of the technical requirements to be designed to let organisations specify their key business objectives in the proposed cloud migration criteria according to their priorities whether it is cost efficiency, decisions around violations of the SLA they signed with their customers, or reducing the time-to-market of their products.

Business agility. Businesses and other organisations can quickly react to the demand of their clients and users in a quick and cost-efficient manner. Within the scope of this dissertation is studying how different cloud migration criteria affect and interact with each other such as shorter time-to-market with the deployment times and unpredictable demand with scalability or the effects of introducing high availability in the cloud-migrated application system to cope with spikes in the

[288] cf. Linnenluecke (2017), pp. 5–12; Chang et al. (2016b), pp. 158–160
[289] cf. Villari et al. (2016), pp. 78–82

data traffic. Some of the existing research[290] compared the costs associated to migrating to a cloud-based deployment and running an application system in a cloud environment to the alternative that uses IT infrastructure with in-house computing capabilities privately owned and managed. This comparison could be included into the decision-making process to take a more informed decision. Khajed-Hosseini et al. designed application systems running on cloud computing environments such as EC2 and S3 from Amazon and Rackspace in addition to purchased hardware or rented data centre computing, networking, and storage resources. Based on their work, one could argue that adopting cloud computing might shorten the time enterprises usually need to market their software solutions due to the accumulated organisational inertia and low-entry barriers for start-up companies.

Pricing scheme. Running an application on a cloud environment usually entails a change in the pricing model for a cloud consumer and some such as Khajed-Hosseini argue that provides cost effectiveness in contrast to an in-house data centre[291]. The more common models are usually tiered, per-unit, and subscription-based pricing. More intricate pricing models already exist within the grid and utility computing research field[292] but some of the existing research argue that simpler pricing models will dominate the market as cloud consumers will increasingly understand a more transparent pricing model better and be more open to it[293]. Even if cloud environments have already started to implement new pricing models based on market mechanisms, it is yet to see how this models will evolve and how enterprises will effectively use them[294]. As an example, Amazon offers Spot Instances so that cloud consumers bid on their spare Amazon EC2 instances and run them whenever their bid exceeds the Spot Instance Price. The price varies in real-time depending on supply and demand.

Pricing scheme with soft reservations. Within the scope of this dissertation, the enacted collaboration with Mohammadi and Kounev strives to apply the concept of

[290] cf. Gholami et al. (2017), pp. 105–113; Khajeh-Hosseini et al. (2010a), pp. 456–457
[291] cf. Gholami et al. (2017), pp. 105–113; Khajeh-Hosseini et al. (2010a), pp. 456–457
[292] cf. Gholami et al. (2017), p. 107; Buyya et al. (2002), p. 1510; Clearwater and Huberman (2005), pp. 14–18; Thanos et al. (2007), p. 7; Wu et al. (2008), pp. 65–67
[293] cf. Baranwal et al. (2018), pp. 86–88; Armbrust et al. (2010), p. 56
[294] cf. Baranwal et al. (2018), pp. 83–86; Armbrust et al. (2010), p. 55

soft reservations to the proposed concept and provide uncertainty-aware reservations in IaaS environments as a contribution of this dissertation[295] and inclusion of soft reservations in the dissertation and the proposed concept. The motivation behind this work is to offer a situation whereby win-win situations for both the cloud consumer and provider stem from the accurate estimation of the resources the cloud consumer needs on the one hand and the transparency from the cloud provider side on the other hand. Although this research work touches upon the business perspective around the adoption of cloud computing environments, it is Motahari-Nezhad et al. who take on this research direction[296]. This research work highlighted the lack of environments for helping businesses migrate their legacy application systems to cloud environments. Additionally, they pointed out the difficulties organisations experience to find and integrate different cloud services for a given set of business requirements. Motahari-Nezhad et al. proposed a conceptual architecture for a virtual business environment whereby individuals and small and medium enterprises (SMEs) can start and operate a virtual business using cloud-based services. Their conceptual architecture include four layers, which include the business context and processes as well as the IT and business services. The lesson learnt about these four layers is that they might be put to practical use into the cloud migration process to improve how organisations migrate their application systems to cloud environments.

4.6.2 Technical cloud migration criteria

The following research works, which come from the computer science field, either focus on the technical side of the migration or identify the technical challenges and the value proposition of using cloud computing environments as Table 4.3 summarises.

From a technological and engineering standpoint, cloud computing infrastructures help cloud consumers and potential end users realise or improve their application systems and systems especially in terms of availability, scalability, and reliability. This section and Table 4.3 explain some of the examples of technical cloud migration criteria but does not intend to be complete. Additionally, cloud architectures

[295] cf. Mohammadi et al. (2013), pp. 228–229
[296] cf. Morgan and Conboy (2018), pp. 297–313; Motahari-Nezhad et al. (2009), p. 7

enable improving some other non-functional properties of the application systems including disaster recovery, sustainability, ease and agility of deployments, and ease of testing.

Interdependencies of technical criteria with other criteria. The list of challenges and opportunities in cloud computing research for software service architectures include topics on performance at runtime, power management, availability aspects, and distributed data usage[297]. When migrating an application system to cloud environments, the criteria driving the cloud migration can oppose one another with organisations trading some of them off with others. For example, improving one of the following three cloud migration mentioned by Hauck et al. could be detrimental to the rest: increasing availability and reliability, improving the level of security and ensuring privacy, and increasing performance in terms of latency and throughput. One can study this research work to analyse which metrics capture the nature of these decisions related to how to trade some criteria off with others upon cloud migration and whether engineers consider these trade-offs at design or run time.

The election of a run time strategy triggers new questions regarding how often to trigger adaptations of an architecture as a result of the runtime analysis and how to adapt a running system. Considering how to trade criteria off is fundamental to this dissertation in conjuction to how to help the description of the criteria interdependencies. In fact, Chapter 6 formalises metrics for the criteria that organisations might have to trade off for others. The initial concern of this research work is to assist the cloud migration at design time to then move forward from there, for example with future work activities out of the scope of this dissertation, towards supporting cloud migration on runtime reacting to changes in the computing environment and the cloud migration criteria. This dissertation would benefit of extending research works focusing on the technical aspects such as availability and performance of software running in cloud environments to add other concerns related to the security-related, provider-related, and business- and economics-related cloud migration criteria.

[297]cf. Phaphoom et al. (2015), pp. 167–172; Hauck et al. (2010), pp. 5–15

Table 4.3: Overview of the technical cloud migration criteria

Criteria	Description
Performance	Different VMs with different computing, networking, and storage, which perform differently for a large set of sub-criteria defined in this section[298].
Interoperability and portability	The ability to let an application system compose or decompose itself using interoperable services and the ability to move data, computation, networking to a different provider[299].
Elasticity and scalability	The level to which an application system provisions and deprovisions resources to automatically adapt itself to workload variations and the ability of an application system to use additional resources to cope with increasing workloads without considering time[300].
Usability	Ease of use, deployment, testing. Automation and cloud management[301].
Availability	The degree to which the target application system is operable or capable of responding to the requests submitted to it[302].
Failover	Disaster recovery, back-ups, failover, failure recovery, replication, geoplex, and multiple geolocations[303].
Worldwide access	Accessibility and collaboration across the internet[304].

Performance. Performance in virtualised environments is a huge and challenging research area on its own in the cloud computing field with virtualisation as a key component of the foundations of the cloud computing concepts (as Section 2.4 shows)[305]. Many organisations and cloud providers have been using virtualisation in their data centres to implement low cost solutions that deliver elastic behaviour. As an example, some research works[306] implement virtual clusters as computing

[298]cf. Bergmayr et al. (2018), pp. 497–512; Andrikopoulos et al. (2013a), p. 498; Fox and Brewer (1999), p. 174; Gilbert and Lynch (2002), pp. 54–56; Klems et al. (2010), p. 68

[299]cf. Štefanič et al. (2018), pp. 3–7; Yangui and Tata (2016), pp. 312–317; Zalila et al. (2019), pp. 266–271; OASIS (2014), pp. 29–39, URL see Bibliography; Cloud Management Initiative (2021), p. n/a; Binz et al. (2013), pp. 692–695; Juan-Verdejo et al. (2014a), p. 43; Metsch et al. (2010), p. 3;

[300]cf. Fazli et al. (2018), pp. 5154–5159; Hu et al. (2016), pp. 1–4; Herbst et al. (2015), pp. 47–50; Ahmad and Babar (2014a), p. 7; Devlin et al. (1999), pp. 1–3; Fox and Brewer (1999), p. 174; Gilbert and Lynch (2002), pp. 54–56;

[301]cf. Phaphoom et al. (2015), pp. 167–172; Hauck et al. (2010), pp. 5–15

[302]cf. Mogul et al. (2017), pp. 2–4; Birman et al. (2009), p. 72; Vogels (2009), pp. 40–44; Pritchett (2008), p. 52

[303]cf. Nabi et al. (2016), pp. 54–59; Zhang et al. (2015), p. 376; Patterson et al. (2002), pp. 5–6; Klems et al. (2008), p. 113; Lenk et al. (2009), p. 26; Nelson (1990), pp. 19–21

[304]cf. Rai et al. (2015), pp. 2–5; Babar and Chauhan (2011), pp. 50–56; Klems et al. (2008), p. 113; Lenk et al. (2009), p. 26

[305]cf. Huber et al. (2016), pp. 3–7; Becker et al. (2009b), pp. 7–12

[306]Such as the following research prototypes: Moha cf. Kim et al. (2016), pp. 95–98; My-Cluster

clusters whereby the researchers place an overlaying custom software stack on top of an existing middleware layer.

Sub-criteria of performance are many: throughput, latency. power management, availability, distributed data usage, service stability, sustainability, service response time. power management, transactional consistency, high availability, resiliency to network partitions, computing efficiency, reliability, or multi-tenancy.

Performance: resource usage computing efficiency. Raicu et al. researched on providing specialised resource management services for light-weight task execution to optimise resource usage computing efficiency. They studied how to enable lower latencies via multi-level scheduling to increase the computing efficiency of running short tasks in grid computing environments. The difference with its application to cloud computing is that grid computing does not natively support interactive application systems but focus on efficiently performing expensive scheduling decisions and coping with the long queue times and large amount of data coming in and out[307]. The proposed concept learns from the findings in grid computing and considers performance when the scalability shift from the application system to an additional software layer.

Performance: reliability. A particular component for the study of scalability in cloud-migrated application systems exist in the CloudSim simulation toolkit of scalable cloud-enabled application systems[308], which focuses on the reliability of the cloud-migrated application system and its automatic scaling capabilities. They implemented a component that abstracts the capacity of a data centre to scale and work in federated manner so that these components run in different cloud deployments and communicate with one another and with cloud brokers. The component monitors the internal state of a data centre with regard to the load balancing and the decision-making involved in application scaling. Their contribution periodically —in terms of simulation time— monitors cloud-migrated application systems and considers specific events triggering dynamic load migration. One could use simulations given the challenges to access the appropriate data to measure performance, which is really hard to evaluate. Integrating their simulations is

Walker et al. (2006), p. 101; and Falkon Raicu et al. (2007), pp. 12–15
[307]cf. Kim et al. (2016), pp. 95–98; Raicu et al. (2007), pp. 16–20
[308]cf. Gupta et al. (2017), pp. 23–29; Calheiros et al. (2011), pp. 32–36

out of the scope of this dissertation, but providing the building blocks to allow for this integration in future work after the completion of this dissertation, hence out of the scope of this dissertation, is reasonable. The potential for this integration is that the simulations might add insight to the decisions that organisations make to migrate their application systems to cloud environments. It might, upon integration, allow for a more informed decision making that uses an overarching set of cloud migration criteria involving different state-of-the-art metrics and estimations in the cloud migration decision.

Performance: multi-tenancy. NIST highlighted pooling common resources to serve multiple cloud consumers as one of the quintessential characteristics of the cloud computing architectures[309]. Section 2.4 explains in Chapter 2 multi-tenancy and how multiple tenants and users share pooled resources including storage space, processing power, memory, and network bandwidth. Andrikopoulos et al. distinguished between tenants and users with regard to consumers in multi-tenant architectures[310]. Tenants separate the consumers that use a multi-tenant service or application into groups. That is, departments, companies, or organisations. These groups are not necessarily disjoint as a consumer may pertain to multiple tenants at the same time.

Performance: multi-tenancy in cloud-based environments. Starting from the different definitions of multi-tenancy in literature[311], one could safely assume that the impact of this dissertation would increase if it takes into account the whole cloud stack of the most common cloud service models —IaaS, PaaS, and SaaS. The same research group accounted for sharing the whole stack of cloud-based service models from hardware to application instances including the operating system and middleware levels[312]. Software engineers let application systems be multi-tenant aware and able to identify and manage multiple tenants with the use of tenant-based identification and hierarchical access control. Multi-tenancy awareness requires supporting message exchanges isolated per tenant and letting tenants manage, administer, and configure their communication endpoints at

[309] cf. US-NIST (2009), p. 2

[310] cf. Bergmayr et al. (2018), pp. 497–512; Andrikopoulos et al. (2013a), p. 498

[311] cf. Rimal and Maier (2016), pp. 290–304; Guo et al. (2007), p. 551; Krebs et al. (2012), p. 428; Mietzner et al. (2009), p. 131; Walraven et al. (2011), p. 382

[312] cf. Rittinger et al. (2018), pp. 2–9; Strauch et al. (2012), p. 212

different abstraction levels[313]. A cloud-based application system decomposes tenant isolation into the isolation of the data and performance of different tenants of that application system. IBM offered a research framework to support isolation in multi-tenant application systems for the SaaS cloud service model[314]. They offered common services to facilitate designing and implementing native multi-tenant application systems.

Performance: multi-tenancy for IaaS, PaaS, and SaaS. On a lower level of abstraction of the cloud service model —the IaaS and PaaS service models— some cloud environments relieve the user from making application systems multi-tenancy aware[315]. Depending on the level of the cloud service model chosen to migrate an application the effort to enable multi-tenancy varies. In principle, IaaS providers support multi-tenancy only at the level of the resources the providers provision. Therefore, software engineers take the burden of adapting the target application and implement multi-tenancy on top of the IaaS level for their platform and application. At the SaaS level, the service provider offers multi-tenancy and ideally makes transparent the resource pooling to the SaaS consumer[316].

The lesson learnt for this dissertation is the importance of taking into account the different degrees of effort involved in adapting the target application depending on the selected cloud service model.

Performance: sub-criteria of multi-tenancy. Cloud providers maximise the utilisation of their resources and profit by transparently providing multi-tenancy to the cloud consumer who perceives to be the only one using the service or application systems without the interference of the use of other cloud consumers. Nevertheless, implementing multi-tenancy in an application system affects the performance and other Quality of Service aspects[317]. These works try to evaluate the isolation regardless the limited visibility some cloud providers offer of their application sub-systems. The optimal degree of isolation for the data architecture of a cloud-based application may vary significantly depending on the technical and

[313] Ibid.
[314] cf. Kim et al. (2016), pp. 551–555; Guo et al. (2007), p. 551
[315] cf. Bello and Mahadevan (2019), pp. 370–378; Walraven et al. (2011), p. 374; Piiroinen (2020), pp. 62–73
[316] cf. Rittinger et al. (2018), pp. 2–9; Strauch et al. (2012), p. 209
[317] cf. Hwang et al. (2015), pp. 126–131; Krebs et al. (2014b), p. 116; Folkerts et al. (2013), p. 173

business considerations together with the effort needed. Therefore, one could argue about the necessity of intertwining the cloud migration choice with other aspects of the decision-making process. Aspects such as the data architecture perspective and the patters for service-oriented application systems that Ralph Mietzner et al. captured as several options for delivering multi-tenant aware application systems[318].

Scalability and elasticity: definitions. Cloud architectures have got the potential to improve on scalability in cloud-based application systems as well as on other related concepts such as elasticity and computing efficiency[319]. Scalability is a prerequisite for elasticity and increasing the elasticity of an application system can improve its computing efficiency. Many definitions exist for elasticity and they do not exactly describe the same concept[320]. Unlike elasticity, scalability does not take into account the temporal aspect regarding how fast an application system scales in addition to how often this happens and at what level of granularity. The capacity of an application system to accommodate increasing workloads by using additional resources is known as scalability. In contrast, elasticity does also assess the extent to which the resources provisioned by the infrastructure cope with resource demands at a any point in time[321]. Scalability is the ability of a cloud-based application system to increase its capacity at a particular layer by expanding its quantity of consumed services at lower layers. Elasticity is the degree to which a cloud-based application system autonomously adapts capacity to workload over time[322].

Scalability: effects in computing efficiency. Babar et al. shared the experiences and observations they gathered through cloud migrating an open source software application system for the collection, analysis, visualization, interpretation, annotation, and dissemination of the software development process and product

[318]cf. Aleem et al. (2019), pp. 8–11; Mietzner et al. (2009), p. 135

[319]cf. Aleem et al. (2019), p. 1; Herbst et al. (2013), p. 25; Kuperberg et al. (2011), p. 5; Herbst (2011), pp. 7–20

[320]cf. Al-Dhuraibi et al. (2017), pp. 430–442; OCDA (2012), pp. 9–11, URL see Bibliography; Liu et al. (2011), p. 45; Shukla and Simmhan (2018), p. 1101; Pham (2016), pp. 39–47; Coutinho et al. (2015), pp. 299–303

[321]cf. Herbst et al. (2018), pp. 375–381; Duboc et al. (2007), p. 378; De Cerqueira Leite Duboc (2010), pp. 381–382; Jogalekar and Woodside (2000), p. 592

[322]cf. Herbst et al. (2018), pp. 375–381; Lehrig et al. (2015), p. 84

data[323]. First, Babar et al. gathered the application system details to then identify the different requirements of the target application system to offer their software application system as a SaaS solution to globally distributed teams deployed on an IaaS environment. Then, they performed the analysis of the requirements to identify architectural decisions driving adaptations that software engineers might perform to cloud-enable their existing application. The aforementioned research paper claims that prior to their implementation, architecture evaluation methods did not effectively assess architectural decisions nor cloud-related quality features such as scalability, accessibility, and computing efficiency. The state of the art suggests that scalability is one of the key technical promises of cloud computing[324]. In fact, cloud-based computing infrastructures offer flexibility for scaling up and down, or in and out, both software and hardware infrastructure without huge investments up front. This research work recommends that cloud-based infrastructures publish their resources through a single provider, let cloud consumers acquire transactional resources on demand, and bill them depending on the resources they used given the increased scalability of the system[325].

Scalability. The scalability of a cloud-based application system is a technical solution with organisational implications as it provides a great opportunity for the growth of the particular organisation. Third-party cloud infrastructures present an opportunity for an organisation to create new product offerings or services. Sales and marketing staff can target a different customer segment or tune the product to achieve product differentiation and grow the market-share of their organisation. In this manner, the scalability properties of cloud-based computing environments enable sales staff to target market segments previously ignored because of limitations to serve a huge set of users[326].

Elasticity. The computing efficiency of an application system expresses the amount of resources consumed to process a specific amount of computing work[327]. Increased levels of elasticity can improve the computing efficiency of an application

[323] cf. Rai et al. (2015), pp. 6–9; Babar and Chauhan (2011), pp. 50–56. Babar et al. migrated the Hackystat open source framework
[324] cf. Mao et al. (2017), pp. 7–12; Zhang et al. (2010), p. 12
[325] cf. Mao et al. (2017), pp. 7–12; Zhang et al. (2010), p. 18
[326] cf. Ferri et al. (2017), pp. 74–78; Khajeh-Hosseini et al. (2010a), pp. 454–456
[327] cf. Al-Dhuraibi et al. (2017), pp. 432–437; Lehrig et al. (2015), pp. 87–89

system as much as other factors other than scaling as a result of the execution of adaptation mechanisms. As an example, implementing an operation in a different manner might improve a system's computing efficiency and that would be totally independent of the elasticity of the application system[328]. Autili et al. defined the user computing efficiency as the ratio of used resources versus their accuracy and completeness achieving their goals[329]. Additionally, research works refer to power consumption efficiency over time[330]. Elasticity captures the level to which an application system provisions and de-provisions resources to automatically adapt itself to workload variations. As a result of these adaptations, at any point in time the available resources meet as closely as possible the demand. Elasticity is one of the cloud migration criteria that affect the overall cloud migration decision based on some metrics proposed in the literature[331].

Elasticity in cloud architectures. Ahmad et al. considered elasticity in the requirement analysis phase of their horseshoe-based architecture migration planning process of their cloud migration framework[332]. They discussed the use of scalability and interoperability to let an application system compose or decompose itself using interoperable services. Furthermore, during the architecture transformation process, the framework dynamically composes interoperable services to support the elasticity of target applications. Benefiting from the elasticity offered by cloud environments requires carefully designing an architecture to utilise the capabilities of the selected cloud environment and let cloud consumers run elastic application systems. This dissertation assumes that, given its dificulty, organisations benefit from being assisted to undertake migration decisions that facilitate autonomic scalability of the cloud-migrated application system according to some elasticity dimensions that take into account resource scaling units, and scalability bounds. The elasticity dimensions fix the resource types scaled with an adaptation process. A lesson for this dissertation is that for each of these resource types, one could set the units along which the amount of allocated resources varies up to the limit imposed by the cloud environment and the cloud migration decision maker.

[328] cf. Al-Dhuraibi et al. (2017), pp. 432–435; DOD (2013), pp. 77–85
[329] cf. Zhang et al. (2019), pp. 672–681; Autili et al. (2011), pp. 63–71
[330] cf. Zhang et al. (2019), pp. 672–681; Chen (2011), pp. 10–11; Chen et al. (2011), p. 772
[331] cf. Hu et al. (2016), pp. 1–4; Herbst et al. (2015), pp. 47–50
[332] cf. Althani et al. (2016), pp. 634–637; Ahmad and Babar (2014a), p. 7

Availability. The availability criterion describes the degree to which the target application system is operable and able to respond to the requests submitted to it. Some stringent definitions of availability define the time window allowed for the response to arrive, or for the application system not to be operable. Having an application system with a high degree of availability is usually a key requirement as financial losses can appear due to outages. Either a company can miss the chance to finalise a monetary transaction due to not finalising a purchase or a customer could lose trust in the company and return less often or even never to that web platform. Application systems usually replicate servers and storage to provide the necessary level of availability. Often cloud providers offer their cloud environment configurations and apparently offer high degree of availability, while they might have to match a high level of reliability too. In that respect the application system is always on and does not go off line.

Sub-criteria of availability. Replication is key to achieving high availability; at the same time, recovery-oriented computing also increases availability through failure recovery instead of protection against it[333]. A data centre replicating data to other servers might alter how those application systems perform with respect to other cloud migration criteria. Software application systems can usually only supply two of the tree properties —transactional consistency, high availability, and resiliency to network partitions— at the same time so organisations might have to decide how they trade these properties off with the other ones[334]. The trade-offs happen as a continuum rather than as binary choices and might impel them to relaxing consistency requirements in pursuit of higher availability[335]. Following this line of thought, this dissertation assumes that organisations trade availability off with other application system properties as a continuum. This way, organisations consider that migrating on-premises application systems and services to cloud environments can improve availability along the properties of worldwide access, workload elasticity, fault-tolerance, and disaster-tolerance[336]. These availability sub-criteria are relevant to this dissertation as cloud architectures introduce new

[333] cf. Mogul et al. (2017), pp. 5–12; Patterson et al. (2002), pp. 5–6

[334] cf. Mogul et al. (2017), p. 7; Fox and Brewer (1999), p. 174; Gilbert and Lynch (2002), pp. 54–56

[335] cf. Mogul et al. (2017), pp. 2–4; Birman et al. (2009), p. 72; Vogels (2009), pp. 40–44; Pritchett (2008), p. 52

[336] cf. Haug et al. (2016), pp. 292–303; Klems et al. (2008), p. 113; Lenk et al. (2009), p. 26

mechanisms and concepts to improve upon these criteria. The take-home message for this dissertation is to allow for classifying complex criteria like this following a rational and then move to experimental work that assess the different criteria from a broader to a finer-grain taxonomy.

Failover. Cloud offerings usually use a fail-over cluster of servers and storage in their data centre, with multiple redundant energy sources and, ideally, replication between multiple geographical locations. That is something that usually large, multinational companies could afford[337]. Server farms are distributed application systems with large-scale server clusters running on a single data centre. Server farms geographically replicated across different sites and connected using the Internet form what the state of the art names a geoplex[338].

Sub-criteria for failover: fault tolerance Basic characteristics for fault tolerance include not having a single point of failure so that the application systems continue working without interruption during the repair process upon an application system failure[339]. In the proposed deployments using resources from public and private cloud environments in addition to on-premises deployments, it is indispensable to construct these infrastructures avoiding the potential for a single point of failure through the implementation of redundancy for critical application system components across all levels. This might entail having redundancy in the access to the Internet with many access points, different providers coming from different locations to avoid a construction work impeding accessing the Internet, using internal or cloudled-based resources for computation, networking, and storage. With the use of cloud environments, cloud consumers can build cost-efficient replication strategies and thereby a recovery cloud. This approach delivers, from the perspective of application migration and configuration management, a disaster recovery strategy with computing running in cloud environments as an emergency environment available on demand[340].

Worldwide access. This dissertation assumes that the proposed concept might follow a modular approach so that researchers, such as those involved in CloudSim,

[337]cf. Nabi et al. (2016), pp. 54–59; Zhang et al. (2015), p. 376
[338]cf. Nabi et al. (2016), pp. 55–61; Devlin et al. (1999), pp. 1–3
[339]cf. Phaphoom et al. (2015), pp. 169–173; Nelson (1990), pp. 19–21
[340]cf. Phaphoom et al. (2015), pp. 172–176; Klems et al. (2010), p. 68

can attach their additional modules to base the decision making in the result of their simulations or estimations. As a result, researchers might, upon integration, use simulation tools to evaluate their hypotheses prior to the actual migration and even reproduce their tests with CloudSim or the proposed cloud migration concept. One could reasonably assume that organisations benefit from the assistance to capture the requirements of various application systems scaling across multiple geographically distributed data centres that different cloud providers offer.

4.6.3 Provider-related cloud migration criteria

Prior to cloud infrastructures, the majority of enterprises and organisations built and maintained their own data centres regardless their core expertise, which created an inefficient ecosystem of data centres[341]. The adoption of cloud infrastructures has the potential to entail provider-related criteria that Table 4.4 lists with consumers paying per the use they do of resources as an utility[342]. Organisations usually adapt their business processes to the new computing paradigm while avoiding the implementation of local optimisations at the cost of organisation-wide performance.

IT cloud governance: sourcing. Organisations migrating their application systems to cloud environments usually adapt their sourcing strategies and policies to the specific case of IT sourcing in cloud-based architectures planning. This dissertation argues for a software evolution helping organisations balance the on-premises, shared, and cloud-based services that fits the organisational goals and resources while keeping an eye on a future with a more and more reduced role of the on-premises solutions. Adopting cloud computing for IT sourcing entails significant planning of change management to undertake an smooth transition to using cloud environments minimising risk with organisations and their IT departments evolving from awareness of cloud services to commitment with understanding and transition (or migration) phases in between. This dissertation assumes, based on the literature analysis[343], that organisations usually move from one to the next stage increasing the business value of adopting the cloud computing paradigm as they learn, experiment, and implement IT sourcing strategies and cloud ser-

[341] cf. Zhu et al. (2017), pp. 66–72; Zatonatska and Dluhopolskyi (2019), pp. 45–49
[342] cf. Carr (2008), pp. 36–37; Carr (2008), p. 67
[343] see Section 4.2 and cf. Juan-Verdejo and Baars (2013), pp. 35–36

vice management. The criteria to be trade-off here would include the degree of cloudification of the proposed architecture, the conformance to previous norms in the on-premises application system, or organisational readiness to undergo the transformation.

IT cloud governance: sourcing to cloud environments which use on-premises services. Some organisations might prefer to steadily migrate their services to cloud-based architectures along the entire product life cycle of their target application system considering the reasons to migrate to cloud environments and aligning them to their business objectives. This dissertation assumes therefore that some organisations might benefit from support to decide which IT infrastructure best suits them. Organisations move from internal data centres, clients, and legacy computing environments to target cloud-enabled application systems that still might use some on-premises services. Organisations would benefit from the support to the transformation phase including the transformation roadmap design, the cloud migration, and the IT service management.

IT cloud governance: ITSM and ITIL in cloud computing. Organisations that conform to the practices of IT service management, or ITSM, observe a set of processes and activities to deliver end-to-end IT services[351]. ITSM supports the design and creation of services as well as their delivery, operation, and support activities. IT service providers undertake these activities and implement IT services following policies that organisations structure in processes and supporting processes while defining clear roles and responsibilities for the appropriate mix of

[344] cf. Mahy et al. (2016), pp. 12–24; Palos-Sanchez et al. (2019), p. 8–12; Suicimezov and Georgescu (2014), pp. 830–835; Petruch et al. (2011), pp. 268–270; Fellows (2008), pp. 13–15; Chang et al. (2016b), pp. 158–160; Agutter (2012), pp. 1–12, 14; Brewster et al. (2012), pp. 11–33

[345] cf. European Union (2016), p. 1; Kaisler and Money (2011), p. 1; Khan and Hamlen (2012), p. 175; Elson and Howell (2009), pp. 2–5; Khajeh-Hosseini et al. (2010c), pp. 8–9; Catteddu (2009), pp. 25–32; Chang et al. (2015), pp. 2–5; Khajeh-Hosseini et al. (2010c), pp. 2–4; Yanosky (2008), p. 32

[346] cf. Chang et al. (2015), pp. 1–4

[347] cf. Zafar et al. (2017), pp. 52–56; Khan and Hamlen (2012), p. 175; Youseff et al. (2008), pp. 7–8; Kaisler and Money (2011), p. 1; Hilley (2009), pp. 7–8

[348] cf. Ongowarsito et al. (2017), pp. 1–5; Youseff et al. (2008), pp. 7–8

[349] cf. Duncan (2018), pp. 1–6; Cheung and Weber (2015), pp. 32–38; Koehler et al. (2010a), pp. 3–4; Hilley (2009), pp. 28–30; Anandasivam (2010), p. 72; Abdulateef et al. (2020), pp. 2–4

[350] cf. Brandenburger et al. (2017), pp. 4–6; Noor et al. (2016), pp. 34–38

[351] Services here refers to customer needs and organisational processes and not to more technology-oriented services orchestrated in IT application systems. Services here deal with continuously improved processes fulfilling customer needs rather than a software functionality

Table 4.4: Overview of the provider-related cloud migration criteria

Criteria	Description
IT cloud governance	Criteria describing different approaches to IT risk management, sourcing strategies, availability or quality of the selected cloud provider, out-of-the-box cloud provider compatibility with ITSM best practices such as ITIL, selection based on different service catalogues[344].
Regulatory concerns	Regulations might limit cloud adoption if the resulting architecture does not fulfil organisation criteria of compliance to the applicable regulatory framework or does not provide mechanisms to audit that compliance[345].
Training and support	Different cloud providers provide better (helpdesk) support or training such as those aiming at decreasing the start-up time or time passed from booking to having a fully set up infrastructure[346].
Criteria related to organisation type and power	Large and small organisations adapt to cloud-enabling their application systems or they might even be born into cloud environments. Different kind of organisations might differ in their evaluation of the different criteria and the criteria they might consider including assessing the effects of losing power to influence or control the infrastructure in their organisations[347].
Data governance	Criteria to compare data management policies of the cloud migration alternatives including data location and the enforcement of data ownership and standards[348].
Vendor lock-in	Criteria to measure the readiness for cloud consumers to easily take their data to another cloud provider[349].
Trustworthiness	Reputation and trust that the organisation associates to the brand name[350].

people, processes, and IT with the goal of providing services to their customers[352]. Among the several standards and framework for IT service management, ITIL stands out as the de facto standard; that is why this section focuses on ITIL for ITSM provision in cloud infrastructures. ITIL details practices for IT service management, or ITSM, to align IT services with business needs. It takes a market-driven approach to propose best practices to service providers based on practical experience of multiple organisations. That way, ITIL guides organisations in clarifying and prioritising their customer-oriented investments in services over the long term[353]. Organisations that migrate their application systems to cloud environments outsource certain elements, tasks, and responsibilities considered

[352]cf. Jothimani (2022), pp. 11–13; cf. Sousa et al. (2017), pp. 21–25; Agutter (2012), pp. 1–12; Brewster et al. (2012), pp. 11–33
[353]cf. Sousa et al. (2017), pp. 12–21; Agutter (2012), p. 14

in ITIL such as the management of IT processes, risk, and finances. Likewise, the use of cloud infrastructures alter the ITIL practices used to design cloud-based IT architectures and to ensure application systems interoperability and an acceptable level of security[354]. Organisations usually make choices to best utilise cloud environments while addressing the challenges for service management; therefore, this dissertation advocates for assessing this decision making comparing the risk, implemented processes, and financial prospects of each cloud migration alternative as well as with respect to how each of them manages ITSM for the cloud consumer.

IT cloud governance: service catalogue. Using ITIL in cloud-based deployments helps to, in addition to considering the fundamentals of ITSM, focus the attention of organisations in adapting their processes and structures to improve the scalability and business agility in the design, development, deployment, management, and continual improvement of their IT infrastructures. It might be beneficial for organisations to adapt their service catalogue to use change management and even cloud services marketplaces. Matching the service catalogue to the cloud migration criteria could help organisations select the most appropriate services that fulfil their cloud migration criteria while striking a balance to deliver the rest of their requirements to an extent that is acceptable to the assisted organisation. A cloud marketplace is an example of a service catalogue offering clearly defined services over a central portal, which uses ITIL best practices to pre-approve the services in the service catalogue and to manage the service catalogue and portfolio. Section 7.2 explains, in Chapter 7, the scenario addressing the evaluation of the proposed concept within the field of migration to Platform-as-a-Service offerings, which even considers and reflects on the potential to use the proposed concept to offer a marketplace of PaaS offerings in order to boost the adoption of the proposed concept.

IT cloud governance: service disintermediation. Service disintermediation in the context of cloud computing happens when consumers buy services directly from a cloud provider or cloud broker rather than through the standard processes for IT provisioning in their organisation. Governance helps overcome this problem that

[354]cf. O'Loughlin (2019), pp. 5–7, URL see Bibliography

could stem from that fact that the organisation fails to provide value to its customers, does not address their requirements in a flexible manner, or when consumers use better and more cost-efficient services running on Service-as-a-Service offerings. One could assume that organisations might need assistance to design and deliver services that the service consumers require through incorporating best practices to alleviate service disintermediation and facilitate governance. Similarly to the risk of service intermediation, shadow IT appears when some organisations build and use IT application systems without explicit organisational approval. Shadow IT application systems escape control of the IT department as they are specified and deployed by other departments. The conflicting aspect resides in that these application systems also offer the potential to quickly innovate and provide software at a faster pace.

IT cloud governance: IT cloud risk mitigation strategies. In the context of cloud computing, IT risk management include risk mitigation efforts, data protection, and business continuity issues. Mitigation efforts include implementing portability measurements to avoid vendor lock-in measures for business continuity such as plans for disaster recovery, backups, restores, and data archival policies[355]. The lesson learnt is that each cloud migration alternative could be evaluated with regards to which of the strategies or tools for risk mitigation the selected cloud environment configuration incorporate.

IT cloud governance: contract management. IT governance in contract management entails verifying whether the cloud environment configuration can respect the SLAs between the cloud consumer and provider. In that respect, cloud migration criteria relate to SLA and could be based on the analysis of the different aspects of the SLA by which they abide. This way, this dissertation assumes that one could rank the different cloud migration alternatives with respect to the extent to which they respect the policies in the SLA[356].

IT cloud governance: organisational change[357]. IT governance aims at mitigating and managing the impact of organisational change when transforming to use cloud resources. Existing frameworks for IT governance account for the core prin-

[355] cf. Mahy et al. (2016), pp. 12–24; Palos-Sanchez et al. (2019), p. 8–12
[356] cf. Suicimezov and Georgescu (2014), pp. 830–835
[357] cf. Prieto-González et al. (2015), pp. 377–381; Dzombeta (2016), pp. 32–51

ciples of the changes upon cloud adoption with timely and effective communication about them. Changes with organisations adopting new business models entail that new organisational roles might appear with newly needed skills while other roles and resources become redundant as tasks become more automatic. For example, new roles like DevOps engineers, cloud architects, or Docker developers might appear at a similar rate as others become less needed or redundant. Such is the case of the need of an IT department or for owned data centres and its operators. Cloud consumers might increase their influence in organisations as the role of the IT department shifts from provider to certifier, consultant, and arbitrator[358]. The increase in the complexity of IT application systems could couple the lack of resources on the cloud provider side to make it challenging for them to offer customisation support to tailor assistance to the large amount of cloud consumers. Additionally, people with the newly required skills might not work in an organisation, which might instead look for them in the market or train people in the organisation to take on the new roles and to learn the new skills. Organisations might need assistance to consider the different criteria of their readiness to go for the different alternatives to cloud migration according to the organisational changes they might undergo and the ones they already underwent. This dissertation works on the assumption of the value of concepts for easily weighing the different options to evaluate which cloud migration architecture favours the cloud migration project depending on whether the organisation structures will effectively drive the cloud migration across the entire organisation[359].

IT cloud governance: IT resources control. Members of an organisation access shared resources in cloud infrastructures that escape the control of the IT department or of the entire organisation. Relinquishing from this control as computing and data moves to cloud environments entails sometimes changes in the business processes of organisations prior to them migrating their application systems[360]. One could argue based on literature that freeing up technological and human resources from typical IT functions can trigger innovation as these people work in other projects[361].

[358]cf. Schneider and Sunyaev (2016), pp. 5–9; Yanosky (2008), p. 32
[359]cf. Petruch et al. (2011), pp. 268–270
[360]cf. Wadhwa et al. (2015), pp. 29–34; Fellows (2008), pp. 13–15
[361]cf. Linnenluecke (2017), pp. 4–12; Chang et al. (2016b), pp. 158–160

All in all, these issues call for a contribution assessing organisations on the consequences in their structures of the cloud migration.

Regulatory concerns. Organisations can compare different alternatives to cloud migration with regards to their conformance to regulation and possibilities for auditing. Some cloud environments might offer out-of-the-box compliance to some regulation. Likewise, criteria within the regulatory concerns includes whether the cloud provider offers the possibility to seamlessly connect monitoring processes and procedures to audit and verify compliance. Newly passed laws in that respect include the European data protection laws (GDPR) [362], the Directive on Copyright in the Digital Single Market[363], and the US Clarifying Lawful Overseas Use of Data Act or CLOUD Act[364]. Non-repudiation falls under this category as the wrongful refusal to follow a transaction —hence, not fulfilling the obligations to a contract—, which also implies that the two parties of a transaction cannot deny having received or sent that transaction[365].

Training and support. Different cloud providers provide better (helpdesk) support or training such as those aiming at decreasing the start-up time or time passed from booking to having a fully set up infrastructure.

Criteria related to organisation type and power. The structure and business processes in an organisation usually impact how organisations can influence or control the behaviour of people working directly with the computing infrastructure they use. Moving computation and data to a different organisation, the cloud computing provider, might entail that organisations might not control the organisation of the cloud provider at the same level as they do within the boundaries of their organisation. Some research works study how support personnel and application system administrators could resist the migration to cloud-based environments out of fear of losing their jobs or not having complete control of the application system whereas application system administrators cloud be in favour of relying on cloud service providers to support their end users in return of the control on their application systems they give up[366]. Under the assumption of this dissertation that

[362]cf. European Union (2016), p. n/a, URL see Bibliography
[363]cf. European Union (2019), p. n/a, URL see Bibliography
[364]cf. Congress-US (2018), p. n/a, URL see Bibliography
[365]cf. McCarthy and Bidgoli (2006), pp. 65–66
[366]cf. Chang et al. (2015), pp. 8–12; Catteddu (2009), pp. 25–32

an organisation drives the cloud migration of its application systems according to its cloud migration criteria, one could reasonably assume that organisations would benefit from support to analyse the different cloud migration alternatives and the risks they bring about for their specific case. For example, cost might be more important to an start-up while at the same time they could benefit from being ready to scale up.

Criteria related to organisation type and power: power shift. The changes in the computing infrastructure to which organisations do not exert complete access, as when they used to run the application system in their premises, entail new challenges to the compliance departments. Organisations might interact with cloud providers through an organisational interface, to which management usually do not belong. This could degrade the authority of executives as they cannot demand special priority from the support staff of an external business entity such as Amazon or Microsoft[367]. Therefore, this dissertation calls for including the ripple effects that picking one cloud architecture over another one has on the decision-making organisation.

Criteria related to organisation type and power: large and small organisations. Large and small organisations adapt their structure to adapt to the cloud computing model and cloud business models to benefit from the potential improvements in scalability and business agility that cloud computing offers. Start-ups belong to a special kind of small organisations or enterprises that can seamlessly and highly beneficially adopt cloud-empowered software solutions as they can shape their organisations around the technology given their initial small size or that they are actually born delivering cloud services. They might deem elasticity (and scalability) as more important than, for example, a feature to interface with a Customer Relationship Management (CRM) application system offered by a Platform-as-a-Service provider. The lesson learnt is that organisations usually tailor the cloud migration of their application systems to their needs and preferences, which may depend on their size and structure, while arguably considering how their cloud migration criteria affect each other.

[367]cf. Chang et al. (2015), pp. 2–5; Khajeh-Hosseini et al. (2010c), pp. 2–4

Organisations that adopt cloud computing impact their enterprise and consumer market. This dissertation advocates for incorporating organisational concerns and provider-related criteria in addition to the technical considerations. In that respect, it distinguishes itself from the state of the art and intends to design a concept to assist organisations and enterprises willing to migrate their application systems to cloud environments but that want to assess how they must change their structure and their processes to accommodate the technological change[368]. This dissertation assumes based on these research works that a concept for cloud migration might offer pre-filled cloud migration criteria (sort of templates) depending on the selected organisation type, which might range from a bank that might be more interested in governance and business processes to a start-up optimising in terms of costs and potential to scale. Their different organisation type and legal environment affect them differently. Likewise, this dissertation assumes that organisations, and start-ups in particular, benefit form assistance in the design of scalable cloud architectures to let them benefit from increased business agility that lets them keep their costs low as they develop a minimum viable product with few users but at the same time making them ready to scale up if they acquire multiple clients[369]. The rating of the importance of different criteria may differ depending on the size of the organisation and patterns might arise that could, in time, bring about size-dependent criteria classifications or ratings of the importance of criteria such as elasticity and cost for smaller organisations and compliance to ITIL and application systems interoperability for large organisations.

Data governance. This includes criteria including to what degree do cloud consumers and end users have control over their own data to do things that the European data protection laws (GDPR) guarantees as the right to access, erase, and port their own data[370]. This also entails questions related to the level of data control and ownership if the user did not pay the bills[371]. These concerns hinder the cloud migration, especially previously to the 2016 EU GDPR law or in absence of agreement of which standard to use to guarantee a correct data management policy that enforces data ownership.

[368] cf. Chang et al. (2015), pp. 6–9; Khajeh-Hosseini et al. (2010c), pp. 8–9
[369] cf. Gangwar et al. (2015), pp. 112–116; Elson and Howell (2009), pp. 2–5
[370] cf. European Union (2016), p. 1
[371] cf. Osanaiye et al. (2016), pp. 147–149, 152–154; Kaisler and Money (2011), p. 1

Data governance: data location. Data location impacts the ability to ensure regulatory compliance as cloud services usually store data redundantly in multiple physical locations. Details about the location of the data might be uncertain or not even disclosed as they vary over time. This make it difficult to ascertain whether the used safeguards in place are enough and whether the application system meets standards, legal, and regulatory compliance requirements.

Data governance: standards. Standards for data governance in cloud environments address issues of data lock-in. As cloud environments do not adopt any predominant standard across the entire cloud stack, it is complicated to support cloud computing users that decide to move data off a particular cloud provider. One could intertwine this issue with the privacy issues and use anonymous authentication based on the use of public keys and cryptography to disassociate data ownership and metadata from the private data they label[372]. This dissertation learns from these data governance standards and advocates from addressing concerns regarding data ownership either disregarding those cloud computing environments not complying to relevant data ownership standards or rating them lower for adoption in accordance to which standards they respect.

Vendor lock-in. IT governance include IT risk management actions and mitigation strategies that tackle, among others, the vendor lock-in problem. That is, whether cloud consumers and end users can take their data with them as they move their application systems to a competing service provider and what do they need to do this[373]. These are criteria to measure the readiness for cloud consumers to easily take their data to another cloud provider; see Sections 4.4 (under "Different approaches to modelling cloud environments") and 4.5.

Trustworthiness. Reputation and trust that the organisation associates to the brand name. The following sub-section explains trustworthiness in more detail in the context of the security and privacy cloud migration criteria.

[372]cf. Zafar et al. (2017), pp. 52–56; Khan and Hamlen (2012), p. 175
[373]cf. Ranjan et al. (2015), pp. 49–53; Youseff et al. (2008), pp. 7–8

4.6.4 Security and privacy cloud migration criteria

According to NIST, security is one of the most important hindrances to cloud adoption. Table 4.5 sums up the security and privacy cloud migration criteria[374].

Data integrity. Data integrity means that only authorised parties can modify assets to the degrees of freedom that their authorisation allows them. Assets refers to data, software, and hardware. Cloud providers try to protect data form unauthorised fabrication, modification, theft, and deletion to maintain data integrity and accuracy[375]. Data integrity is the concern of organisations that want to maintain or increase the integrity, accuracy, and consistency of their application systems as they lose control over the physical security. Some cloud providers offer secured development tools with an embedded security model to assist developers, access control, encryption. Migrating a business application system to cloud environments removes enterprise boundaries and changes the security landscape of application systems hosted in cloud infrastructures[376]. Although cloud environments are neither more nor less secure than other computing environments, running application systems in cloud infrastructures imposes new risks, threats, challenges, and opportunities to those deployments. In that respect, the adoption of cloud computing to run application systems, similarly to the use of any new computing technology, might require (or offers the chance) to re-design or re-architect the target application or the underlying infrastructure to meet or even exceed the state-of-the-art security requirements[377]. Organisations and individuals want to maintain (or even increase) the integrity of the new computing environments they intend to use[378]. Were they to migrate their data and application systems to public cloud environments, they would lose control over the physical security because cloud providers share their computing resources with other companies.

Data integrity: data accuracy. Data accuracy as one of the aspects of data quality is a metric into data reliability but relates to data security in cloud environment with cloud providers that keep it high and store the correct data values of an object.

[374] cf. Ali et al. (2015), pp. 357–366; US-NIST (2009), p. 2
[375] cf. Gheyas and Abdallah (2016), pp. 1–6; Claycomb and Nicoll (2012), pp. 392–393
[376] cf. Shaikh and Sasikumar (2015), pp. 493–498; Hu and Klein (2009), p. 738
[377] cf. Ali et al. (2015), pp. 367–372; Cloud-Security-Alliance (2009), p. 11
[378] cf. Chang and Ramachandran (2015), pp. 142–149; Rittinghouse and Ransome (2009), pp. 153–182

Cloud providers will usually ensure the right value in a consistent and unambiguous form including protecting their data from insider attacks. Insider attacks do not happen too often but have got greater impact than external attacks and therefore cloud service providers try to minimise these threats to reinforce cloud consumers confidence in them[379]. Insiders that administer the software application systems at the provider ultimately possess the technical means to access the customers' virtual machines[380]. The paper includes approaches to define, identify, classify, and assess insider threats in the context of cloud computing environments as well as the analysis of malicious providers —or Acid Clouds[381]— with the potential of gaining large amounts of sensitive data from numerous sources.

Data integrity: data consistency (validity). Some works propose a protocol to verify the consistency of cloud object storage to let a group of mutually trusting clients detect consistency violations in cloud storage[382]. Additionally, cloud providers restrict and control the access to the facilities where they have their hardware and minimise the number of staff who access critical infrastructure components[383].

Data security. Cloud providers do not offer enough information about how things run in their premises. A cloud provider with low security standards could decrease

---

[379] cf. Singh and Chatterjee (2017), pp. 89–93; Duncan et al. (2012), p. 857

[380] cf. Singh and Chatterjee (2017), pp. 91–95; Duncan et al. (2012), pp. 858–860

[381] cf. Akinrolabu et al. (2017), pp. 5–7; Creese et al. (2013), pp. 660–665

[382] cf. Brandenburger et al. (2015), p. 7

[383] cf. Coppolino et al. (2017), pp. 136–142; Cachin et al. (2009), pp. 81–83; Someswar and Hemalatha (2012), pp. 4–9

[384] cf. Ali et al. (2015), pp. 357–366; US-NIST (2009), p. 2; Hu and Klein (2009), p. 738; Cloud-Security-Alliance (2009), p. 11; Rittinghouse and Ransome (2009), pp. 153–182; Rittinghouse and Ransome (2009), p. 159; Santos et al. (2009), p. 3; Subashini and Kavitha (2011), pp. 10–11

[385] cf. Chang and Ramachandran (2015), pp. 145–151; Subashini and Kavitha (2011), p. 1; Hu and Klein (2009), p. 735, 737–738; Kooper et al. (2011), pp. 195–196; IBM (2010), pp. 5–16; Claycomb and Nicoll (2012), pp. 392–393; Duncan et al. (2012), p. 857; Duncan et al. (2012), pp. 858–860; Creese et al. (2013), pp. 660–665; Juan-Verdejo et al. (2014b), p. 467; Kooper et al. (2011), p. 195; IBM (2010), p. 1

[386] cf. European Union (2016), p. 1; Kaisler and Money (2011), p. 1; Khan and Hamlen (2012), p. 175

[387] cf. Bauer et al. (2017), pp. 147–153; Oliveira Albuquerque et al. (2016), pp. 3741–3753; Peltier (2013), pp. 219–238; Kandias et al. (2011), pp. 93–95; Chun et al. (2007), p. 192; Yumerefendi and Chase (2007), p. 9; Yumerefendi and Chase (2007), pp. 1–2; Brandenburger et al. (2015), p. 7; Cachin et al. (2009), pp. 81–83; Someswar and Hemalatha (2012), pp. 4–9; Herlihy and Wing (1990), pp. 463–466; Lamport (1979), pp. 690–691; Cachin et al. (2007), p. 137; Robinson and Park (2010), p. 191; Dias Canedo et al. (2011), p. 44; Santos et al. (2009), pp. 1–3; Subashini and Kavitha (2011), pp. 5–6; Hu and Klein (2009), p. 735, 737–738; Juan-Verdejo (2012), pp. 15–19; Juan-Verdejo et al. (2014b), p. 467; Kooper et al. (2011), p. 195

Table 4.5: Overview of the security and privacy cloud migration criteria

Criteria	Description
Data integrity	Organisations want to maintain or increase the integrity of the data as they lose control over them and the physical security after the migration. This includes considerations about data accuracy and data consistency as well as security-aware tools by the cloud providers[384].
Data security	Criteria about which mechanisms against insider breaches and its communication that the cloud provider offers and the same for the respect to standards of data privacy and security. This includes criteria about whether the selected cloud provider provides the necessary tools for secured development to assist developers as well as for access control, encryption, confidentiality, reliability, and information governance [385].
Data privacy	Cloud consumers exert control over their data in accordance to existing legislation; thus, cloud providers might put appropriate privacy-aware and secured mechanisms at their disposal to do it[386].
Trustworthiness	Cloud providers supporting accountability and auditing procedures as well as data privacy and security standards[387].

the availability to their cloud consumers and cause them data and money losses. These concerns affect the likelihood that organisations would be willing to cloud migrate their application systems. However, software engineers can address these concerns with successful and effective application and data migration mechanisms. The cloud security alliance recommended specific controls and technologies to enforce information governance. That is, the set of multi-disciplinary structures, policies, procedures, processes, and controls organisations implement to manage their data with their immediate and future regulatory, legal, risk, environmental, and operational requirements in mind[388]. Cloud consumers worry about insider breaches because it is difficult to assess the protection mechanisms a cloud provider puts in place to protect the data of organisations using their infrastructures.

Data security: confidentiality. Cloud providers intend to provide a technical solution that guarantees the confidentiality of computation and make it easy for cloud consumers to verify[389]. For these reasons, cloud providers make their

[388]cf. Rahimi et al. (2016), pp. 150–152; Subashini and Kavitha (2011), p. 1; Kooper et al. (2011), pp. 195–196; IBM (2010), pp. 5–16

[389]cf. Coppolino et al. (2017), pp. 126–135; Peltier (2013), pp. 219–238; Kandias et al. (2011), pp. 93–95

accountability and auditing procedures more stringent[390]. Some other research works supply the audit interface to let an auditor verify confidentiality over arbitrary intervals of use history so that faulty server risk detection as they revert writes[391]. Other participants may verify audit results without trusting the auditor.

Data security: secured development tools. The use of development tools with an embedded security model might be extremely important for some cloud migration projects and irrelevant for others. These tools may assist developers to restrict access of end users of a production environment to the data they are authorised to access[392]. According to a survey[393], many organisations have not adopted these security-aware development tools and rush into running their application systems on cloud computing to benefit from the potential for increasing costs saving and scalability. One of the basis of this dissertation is to start with a partial deployment whereby organisations keep their sensitive data that they are not ready to migrate in their premises. Putting security mechanisms in place, such as encryption, could allow for the secured migration of sensitive data and computation to cloud environments although 100% secured does not really exist. The lesson here is that the provision of some privacy-enhanced tools (or not) to support development might be criteria on which to base the decision of which cloud migration alternative to select.

Data security: encryption. With a clear focus on the privacy requirements of migrated application systems to cloud environments, some benchmarks address the migration to cloud environments focusing on different data encryption strategies to overcome privacy-related issues. Encrypting middleware layers to securely migrate legacy application systems to cloud infrastructures with the goal of improving data confidentiality and integrity[394]. Encryption usually protects against internal and external leaks. That way, migration transparency and cloud provider independence increase while enabling higher degrees of cloud-based scalability. Likewise, this dissertation assumes adopting state-of-the-art encryption mechanisms to deliver

[390] cf. Matetic et al. (2017), pp. 1289–1290; Chun et al. (2007), p. 192; Yumerefendi and Chase (2007), p. 9
[391] cf. Brandenburger et al. (2017), pp. 1–4; Yumerefendi and Chase (2007), pp. 1–2
[392] cf. Chang and Ramachandran (2015), pp. 145–151; Rittinghouse and Ransome (2009), p. 159; Santos et al. (2009), p. 3
[393] cf. Subashini and Kavitha (2011), pp. 10–11
[394] cf. Sciancalepore et al. (2018), pp. 5190–5192; Hu and Klein (2009), p. 735

the required privacy and security cloud migration criteria. The encrypted user reference data might be non-transaction data about service customers and providers: authentication data —password, certificates—, personal information including payment information, and personalized data —like preferences, purchase history, and recommendations. Organisations usually weigh the advantages and disadvantages of encrypting data at different levels such as storage, database, middleware, or application layers.

Data security: availability. Properly designed information systems and application systems make information available when needed through computing application systems that store and process that information in combination with security controls that protect it and communication channels that transport it. In the context of data security, this entails protecting the computing application system against denial-of-service attacks (when a flood of incoming messages forces the application system to shut down), flooding attacks (when a cloud application system scales up to the point of overloading itself to respond to an enormous number of requests for data processing), insecure APIs[395], account, service, or traffic hijacking[396].

Data privacy. Data privacy refers to the collection of data and its dissemination. In addition, data privacy refers to political and legal matters. Some of the newly appeared laws try to improve upon it[397]. Cryptography alone cannot meet the requirements of cloud privacy so one might turn to trusted computing by enforcing privacy and security via tamper-resistant hardware. Although this approach present several limitations[398]. Trusted computing cloud architectures presumes trusting a single provider wheras distributed trust envisions distributing data across a collection of cloud providers and delegating privacy enforcement[399].

Trustworthiness. Software engineers migrating application systems to cloud environments usually need to understand and document the user requirements

[395] Application programming interfaces

[396] cf. Bennasar et al. (2017), pp. 72–74; Rabai et al. (2013), pp. 72–74; Paquette et al. (2010), pp. 69–70; Hanna (2009), pp. 6–11; Subashini and Kavitha (2011), pp. 57–61; Krutz et al. (2010), pp. 43–46

[397] cf. European Union (2016), p. n/a, URL see Bibliography

[398] cf. Bukowski (2016), pp. 81–86; Anderson (2008), pp. 239–252; Parno (2008), pp. 1–6

[399] cf. BPR4GDPR Consortium (2021), p. n/a; Ateniese et al. (2007), pp. 598–602; Bowers et al. (2011), pp. 501–505; Juels and Kaliski Jr. (2007), pp. 584–591

of their application systems to design an appropriate solution, but they might also need to come to grips with the fundamental security requirements and the needs for data protection and information security[400]. It all together shapes the design of the information system to offer a multi-user distributed environment that addresses the security challenges posed at different cloud deployment levels. The threats to address stem from the security requirements and depend on the cloud deployment model of the migrated application. In addition to these cloud migration criteria, the subjective trust security sub-criteria depend on the devised architecture. In a cloud migration initiative whereby only non-sensitive data leaves the organisation premises of the organisation, the trust the cloud consumer puts into the cloud environment plays a different role than in a cloud migration project moving sensitive data[401]. The Cloud Security Alliance (CSA) recommends multi-disciplinary structures, policies, procedures, processes, and controls for organisations to manage their data according to their regulatory, legal, risk, environmental, and operational requirements[402]. The lesson learnt for this dissertation is the reasonable assumption that trustworthiness is an aggregate of its sub-criteria that might include compliance to some security standards, security-aware tools provision, and subjective elements related to the cloud provider brand.

Trustworthiness: partial migration. Potential to architect mixed architectures using on-premises and cloud environments with data and computation running on cloud environments and locally to the organisation to cope with data sensitivity, privacy, and trust issues. A trusted cloud computing platform can help in the use of IaaS environments with node and virtual machine management. In cloud environments, describing trust-related cloud migration criteria entails considering multiple cloud computing models[403]. Trust is a subjective criteria that affects each of the alternative cloud deployment models according to the devised architecture of the target application system. For example, trust plays a different role when partially migrating an application by only moving non-sensitive data out of the

[400] cf. Tang et al. (2017), pp. 302–307; Robinson and Park (2010), p. 191; Dias Canedo et al. (2011), p. 44; Herlihy and Wing (1990), pp. 463–466; Lamport (1979), pp. 690–691; Cachin et al. (2007), p. 137

[401] cf. Juan-Verdejo et al. (2014b), p. 467

[402] cf. Rahimi et al. (2016), pp. 142–145; Kooper et al. (2011), p. 195; IBM (2010), p. 1

[403] cf. Rahimi et al. (2016), pp. 150–156; Subashini and Kavitha (2011), pp. 5–6; Santos et al. (2009), pp. 1–3

computing premises of the organisation than when undertaking a full-fledged data migration that moves sensitive data. Some works target insider breaches in the context of SaaS security together with other thirteen issues at the SaaS level, while continuing with the IaaS and PaaS security issues.

4.7 Research design

The research design narrows down the scope of this dissertation by pinpointing the research method followed and by defining its research frame of reference[404]. Additionally, this section describes the research processes and methods used to analyse the research challenges of cloud migration assistance and finally answer the four research questions of this dissertation (explained in Chapter 1). The research design of this dissertation acts like the thread connecting the state of the art of cloud migration assistance presented in Chapter 4 with the requirements of the proposed cloud migration DS concept. Chapter 6 will present the concept design according to the requirements specified in Chapter 5.

4.7.1 Research scope

Figure 4.10 shows the research scope of this dissertation: designing the proposed concept. A decision support concept for the migration of application systems to cloud environments. The analysis of academic and industrial contributions to the state of the art of cloud migration[405] (already shown in this chapter) is the first step to start defining the research scope to then focus on answering the four research questions of this dissertation[406]. Limiting the research scope of the dissertation helps breaking down the research design into its components: the research method, research frame of reference, and research processes and methods.

[404] See Section 1.5 in Chapter 1 for the research methodology of this dissertation, which helps limiting the research design

[405] The state of the art is represented by the five input components to the cloud migration DS concept (the grey elements) in Figure 4.10, which is a refinement of Figure 1.3 shown in Chapter 1

[406] Section 1.2 in Chapter 1 explains the four research questions of this dissertation: *Research question 1:* On how a decision support concept assists organisations in the decision-making involved in cloud migrating their application systems. *Research question 2:* To investigate which criteria driving the cloud migration decision. *Research question 3:* What are the criteria or requirements of organisations to select to which cloud environments to migrate their application systems? *Research question 4:* Which requirements and characteristics of the application system prior to the cloud migration do organisations consider in the decision-making process?

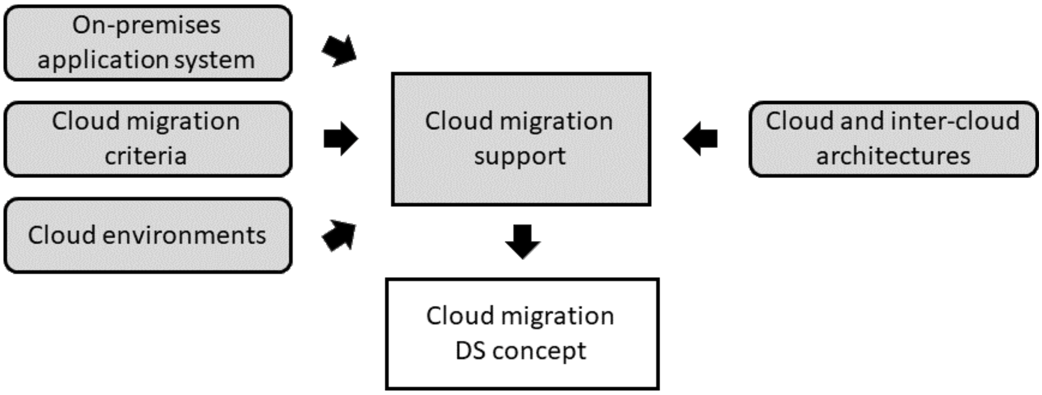

Figure 4.10: Scope of the contribution of this dissertation[407]

Section 4.2 informed the answer to *research question 1* (How can a decision support concept assist organisations in deciding how to migrate their application systems to cloud environments) by defining what cloud migration support (at the centre of Figure 4.10) and its fundamental input components on the left- and right-hand side. Answering the other three research questions started with Section 4.3 describing the application system prior to the cloud migration, Section 4.4 showing different alternatives to modelling existing cloud environments, Section 4.5 describing different deployments to cloud environments and the local premises, and Section 4.6 detailing the cloud migration criteria that drive the cloud migration.

The analysis of the state of the art helps specifying the scope and potential solutions that might deliver the main contribution of this dissertation: the cloud migration concept. Potential solutions that might use model-driven engineering or the Analytical Hierarchy Process (AHP). Some other research works describe different criteria affecting the cloud migration project. Criteria that capture the requirements of organisations for cloud adoption and the particularities of the application systems they migrate. The research scope includes selecting cloud environments and particular cloud environment configurations (following cloud and inter-cloud architectures) to migrate the application system after assessing how that would fulfil the cloud migration criteria.

The research design shown in Figure 4.11 points out the current phase of this dissertation, which works between Goal 1 —analysis and structuring of the state of the art— and Goal 2 —requirements identification— and as an overall methodology

[407] Own illustration

for the entire dissertation and research process. The research design frames the methodology followed by this research according to the design-oriented information systems research principles[408]. Following this principles helps in delivering the four goals of this dissertation, shows on the right-hand side of Figure 4.11[409]. The research design envisions iteratively delving into the research questions to increasingly detail answers to them.

The elicitation of the research method, frame of reference, and processes is the first step towards identifying the requirements of this dissertation in Chapter 5. Based on these requirements, the design of the proposed concept will take place. Finally, with the aim of evaluating the proposed concept in real-life settings, the concept will potentially be implemented as a prototype. The prototype is designed so that the concept can be evaluated via three real-life scenarios for: cloud migration to PaaS offerings, cloud migration of business intelligence, and cloud migration of collaborative networks systems.

4.7.2 Research method

Design-oriented information systems research focuses on the outcomes of the performed research. It offers guidelines to evaluate and incrementally iterate towards the development of artifacts that achieve the research objectives of the dissertation[410] —in this case the developed artifact is the proposed concept. Figure 4.12 shows the first outcome of this dissertation after fulfilling the first goal of this dissertation: the elicitation of the research frame of reference and methods. The result of the first goal is finding the combination of methods for prototyping and their application to the three scenarios selected to evaluate the proposed concept.

A combination of qualitative and quantitative research methods are used in this dissertation to examine the research domain and improve the re-scattering of components to cloud environments and the local premises[411].

[408]Section 1.5 explains the design-oriented information systems research principles
[409]The goals (see Figure 4.12) are based on the research questions of this dissertation
[410]cf. Dresch et al. (2015), pp. 67–74; Hevner and Chatterjee (2010), pp. 8–11; Design science research strives for improving the functional performance of the design artifacts: Witell et al. (2016), pp. 2863–2865; Ostrom et al. (2015), pp. 127–130; Peffers et al. (2007), pp. 45–55
[411]cf. Sjoberg et al. (2007), p. 358
[412]Own illustration

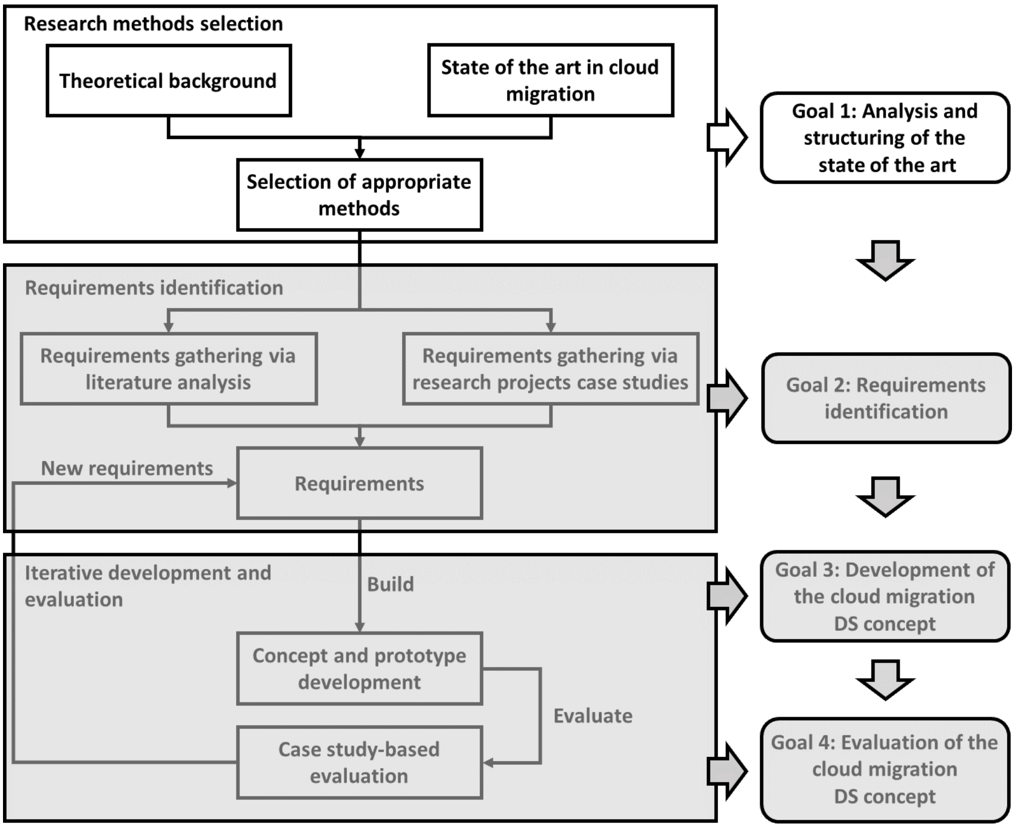

Figure 4.11: Research design (part 1)[412]

Research contributions that mix research methods[413] combine qualitative and quantitative research methods to balance out their different limitations[414]. Mixed-method research copes with these limitations via using an overarching research methods to take into account multiple insights and factual data to provide a lifted-up viewpoint. The use of several research methods with their limitations helps correcting the evaluation of the cloud migration concept according to the weaknesses and strengths of each research method used. These are research methods ranging from conducting a literature review or performing action research and exploratory research to applying empirical methods and observation methods to cases studies or scenarios, as well using method engineering and meta-modelling methods.

[413]cf. Creswell (2013), pp. 203–214
[414]cf. Stol et al. (2016), pp. 120–124; Easterbrook et al. (2008), p. 285
[415]Own illustration

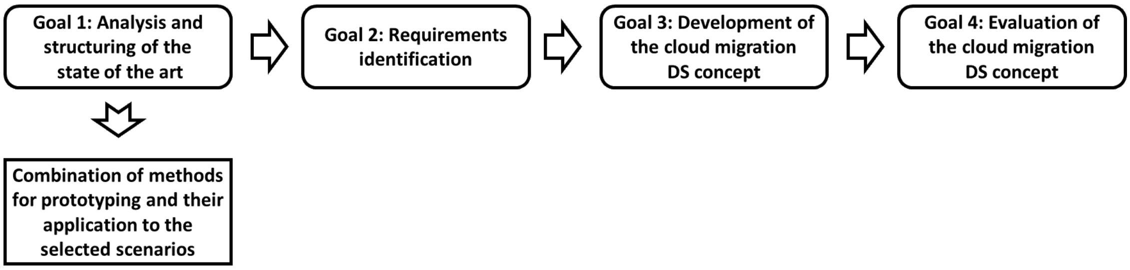

Figure 4.12: Result of the first goal[415]

Literature reviews were continually conducted during the research towards this dissertation to update the cloud migration foundational research work and especially at the beginning[416]. Action research methods involve undertaking representative actions in practical phases such as the iterative conception and development of practical approaches to improve the theories proposed in the cloud migration concept[417]. Exploratory research with real-life settings facilitates testing the hypotheses of this dissertation and represents the combination of confirmatory and exploratory research[418].

Empirical methods for observing real-world phenomena drive this research to build around the hypotheses of this dissertation to test them. These methods pertain to the body of observation methods[419]. Within this dissertation, empirical methods helped building the research questions to be answered with the hypotheses assumed for the design of the proposed concept. The construction of new methods based on existing ones belongs to method engineering, which advocates for the design, construction, and evaluation of these methods, techniques, and support tools for the development of information systems. With method engineering, this dissertation uses a single-purpose method with domain-specific processes and knowledge to develop the proposed concept[420].

[416] The literature review research method is of special relevance at the beginning of the research process: cf. Machi and McEvoy (2016), pp. 5–7; Cooper (1998), pp. 40–42; Jesson et al. (2011), pp. 103–127

[417] cf. Garousi et al. (2019), p. 101,117–119 ; Potts (1993), p. 19; Avison et al. (1999), p. 94

[418] cf. Stol et al. (2016), pp. 125–129; Easterbrook et al. (2008), pp. 309–310; Denzin and Lincoln (2011), pp. 278–279

[419] cf. Stray et al. (2016), pp. 117–122; Seaman (1999), pp. 557–558; Sjoberg et al. (2007), pp. 375–376

[420] cf. Brinkkemper (2016), pp. 235–240; Mayer et al. (1995), pp. 7–10; Brinkkemper et al. (1996), pp. 1–7

As for meta-modelling, this dissertation might use meta models to facilitate the cloud migration decision making and to model how cloud migration criteria interact with each other according to: the description of the target application system and the selected cloud computing environments[421].

The selected research methods are tools to cope with the problem statement of this dissertation. These methods are combined in a research plan to follow the described research design to answer the four research questions of this dissertation[422]. This research plan focuses first on the process to come up with the proposed concept, which addresses the challenges of cloud migrating application systems, to then evaluate its application to real-life scenarios[423].

4.7.3 Research frame of reference

The business administration and communications fields call for using empirical research to decompose the research problem by devising a reference framework that pinpoints the factors influencing this dissertation as well as its context[424]. The research frame of reference delimits the research design context of cloud migration support to limit the field of research of this dissertation.

The reference framework shown in Figure 4.13 helps with a top-down description of the main research challenges. It states the research scope and connect each element of the research frame of reference with its related section of the state of the art: Sections 4.2 to 4.6.

Figure 4.13 is not complete as it describes the frame at a high level of abstraction to make it understandable regardless the field of expertise of the reader. The research frame of reference can be broken down by including further finer-grained elements[425]. The research frame of reference shapes the overall logical consistency of the planned research while going into more details about what this dissertation should achieve to fulfil its goals[426].

[421] cf. Rossi (2015), pp. 1–3; Gonzalez-Perez and Henderson-Sellers (2008), pp. 53–72; Jeusfeld et al. (2009), pp. 22–31

[422] For reference, see the problem statement described in Section 1.1 and the corresponding research questions in Section 1.2

[423] Chapter 6 explains the proposed concept and Chapter 7 its application to real-life scenarios

[424] cf. Kubicek (1977), pp. 9–12

[425] cf. Atteslander (2003), pp. 44–57

[426] cf. Kubicek (1977), pp. 9–12

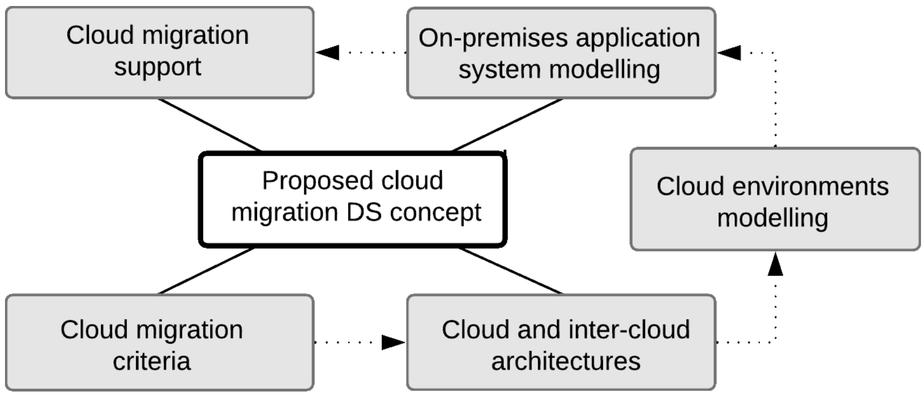

Figure 4.13: Research frame of reference[427]

The research frame of reference (Figure 4.13) includes, on the top right-hand side, the cloud migration support element of the proposed cloud migration DS concept. Directly to its right, the on-premises application system modelling refers to the basic modelling elements needed to describe the architectural model of the target application. An architectural model that includes its constraints, its functional and non-functional requirements, and specification of the component-to-component communication. Further, the cloud environments modelling element captures the knowledge about the cloud environments as well as the effects the cloud selection has got on the portability of cloud architectures. The cloud and inter-cloud architectures element focus on the particularities of multi-cloud deployments. Finally, on the bottom left-hand side of Figure 4.13, the cloud migration criteria element refers to the decision criteria with which organisations can describe the goals they want to achieve with the cloud migration.

4.7.4 Research processes and methods

From an Information Systems (IS) standpoint, finding the appropriate *combination of methods for prototyping and their application to the selected scenarios*[428] entails analysing existing methods. Methods for both assessing the information needs of cloud migration decision support concepts and developing the proposed concept. After the analysis of the methods summarised in isolation in Tables 4.6 and 4.7, the most suited for this dissertation are selected. These research methods are

[427]Own illustration

[428]That combination is the artifact produced, as Figure 4.12, by this dissertation as its Goal 1: *elicitation of the research frame of reference and methods*

combined at different stages of this dissertation to benefit from the advantages of each one of them while mitigating their disadvantages[429].

The selection of the research processes and methods happens in parallel with the review of existing literature and state of the art on cloud computing, software migration, and cloud migration. This is because both phases affect and feed each other[430].

Based on that parallel execution of research methods, the requirements emerge in an incremental and iterative manner that lead to the incremental and iterative development of the proposed cloud migration concept[431]. Thus, frequent versions of the cloud migration DS concept have been available since the beginning of this dissertation to quickly scrutinise it, put it to test, and show it experts. This approach fosters quick and early feedback from the research community, which is iteratively incorporated into the development and evaluation of the proposed concept. The decision-making process involved in the cloud migration of application systems is quite unstructured with multiple dimensions affecting how organisations migrate them to cloud environments.

Additionally, these multiple dimensions affect the cloud migration to a different degree depending on the particular application system, cloud migration criteria, and selected cloud environments. Therefore, methods that are good for structured tasks and processes —such as tasks analysis— are not the perfect fit for the cloud migration support research domain.

Multiple **brainstorming sessions** (see Table 4.6) helped getting started with ideas on cloud migration support and to explore potential research paths. Most of those ideas were discarded through experimentation and **literature analysis** (see Table 4.7) to keep the best candidate ideas that could make an impact in the research and industrial communities working on cloud migration support.

[429] The analysis of literature about research processes and methods was needed to select and combine them appropriately: cf. Lemke and Brenner (2015), pp. 193–201; Heinrich and Lehner (2005), p. 421; Herzwurm and Pietsch (2009), p. 159; Goeken (2007), p. 104; Stickel (2001), p. 120

[430] Chapters 2 to 4

[431] See Chapter 5 for the requirements

Table 4.6: Methods to assess this dissertation information needs (1/2)

Methods[432]	**Applicability and limitations**
Interviews	Interviewing is a method widely used as it is, in principle, applicable to the majority of research works. However, it requires some experience to distinguish the relevant information acquired. The interviewer should possess an acceptable level of knowledge about the topic to lead the interview.
Questionnaires	Questionnaires in a written form are an effective mechanism to collect large quantities of information about the different aspects of a topic from many interviewees. It is, though, not the best mechanism to go in the deep in the topics addressed.
Focus groups	Organising a focus group helps bridging the communication difficulties with several interviewees in parallel. The method allows to gather information from multiple respondents at the same time but, like in the case of a regular interview, the interviewer should be knowledgeable on the discussed topic.
Brainstorming sessions	Brainstorming sessions bring about lots of ideas and hypotheses to quickly reject them or to design the next experiment to try to deepen the knowledge towards proving the selected hypotheses. Nevertheless, brainstorming does not usually help to come to more concrete decisions.
Methods based on introspection	In cases whereby the researcher does not know the application domain or system very well, methods based on introspection help increasing this knowledge (if the analysts exercise quite a deep knowledge on the field of research). The method is based on the researcher (sometimes together with the subject) acting as the introspector who uses the own relevant emotions, thoughts, imaginations, sensations, or memories as data for analysis.

The literature analysis research method served well to investigate the fields of cloud computing, software migration, and cloud migration. This analysis allowed for going deeper into contributions related to on-premises application system modelling, multiple cloud migration criteria, cloud migration support, cloud environments modelling, and inter-cloud architectures. The identification of research gaps and existing challenges emerged from this study of the state of the art and helped narrowing down the research domain and scope of this dissertation. The

[432] Research processes and methods

challenges identified in this study, such as the complexity of the cloud migration decision, drive the decision of offering cloud migration support through the proposed concept and sub-concepts. Some of which might have not been been previously used by organisations when cloud migrating their application systems.

Research **workshops** (see Table 4.7) worked well to present to the research community the iterative and incremental improved versions of the cloud migration DS concept. This helps collecting expert feedback and pivot the research direction.

The **prototyping** and **scenarios** methods in Table 4.7 are the selected research methods to evaluate the cloud migration concept. In addition, other research methods will play a role for different parts of this dissertation in combination with these basic methods that conform the core of the applied research methods. Such is the case of **interviews**, **brainstorming**, **questionnaires**, and **workshops**, see Table 4.6 & 4.7.

Prototyping is the method selected to realise the different IT-related aspects of the proposed concept with the goal of evaluating it in real-life scenarios. A prototype would help evaluating the cloud migration given its complexity and the multiple criteria taken into account. A prototype helps organisations to design the cloud migration themselves by being assisted by the proposed concept[434].

The scenarios method puts together system analysts (the author of this dissertation in this case) and users (cloud migrating organisations) to refine the solutions to which to apply the cloud migration DS concept. The three main scenarios —cloud migration to PaaS offerings, cloud migration of business intelligence, and cloud migration of collaborative networks systems— facilitate discovering the extent to which the contribution of this dissertation can be applied to different fields of application. Scenario development is the perfect match of research method as the involved decision-making tasks of the cloud migration in this field of applicability are complex and the scenarios make it lighter to put the proposed concept to use.

In the span of the execution of the three evaluation scenarios, different dissemination **workshops**, **interviews**, and **questionnaires** (see Table 4.6 and 4.7) are used to gather feedback from the users to derive conclusions and evaluate the

[433]Research processes and methods
[434]cf. Korsgaard et al. (2016), pp. 71–75; Eder et al. (2002), p. 10

Table 4.7: Methods to assess this dissertation information needs (2/2)

Methods[433]	Applicability and limitations
Tasks analysis	Task analysis refers to observing participants in action as they perform their tasks. It usually helps driving the structured decision making involved in a research process.
Documentary analysis	This method gathers information on the status quo of the touched topic from documents —public records, personal documents, and physical evidence— but does not help identifying which data the research will need from the interviewees or which information is relevant.
Literature analysis	This method usually refers to academic-oriented literature on a particular topic. It includes the knowledge with substantive findings and contributions that are theoretical and methodological. Literature analyses are secondary sources, which do not contain new or original experimental work.
Workshops	A sort of small conference for researchers (academic and non-academic) to present and discuss their contributions and research work.
Analysis of technical data	Like documentary analysis, this analysis gathers the status quo of the research object but, unlike it, it extracts the structure of the data through the evaluation of the data gathering activities.
Case studies	The interactions between system analysts and users helps refining future solutions starting off with case studies. Real-life case studies help when the decision-making tasks are complex and facilitate discovering new solutions and research directions towards them.
Scenarios	Scenarios capture system use contexts to offer specific circumstances to review the attributes of a proposed solution. Scenario analyst can assess how well or easily a scenario satisfies its constraints. Scenarios are less structured than case studies and facilitate analysing complex decision-making tasks and discovering how to research towards them.
Prototyping	Prototyping a solution is very helpful when the user does not know the functionalities that the researcher will develop. Researchers usually combine prototyping with other methods such as interviews and workshops.

contribution of this dissertation. Questionnaires are used in the evaluation scenario for cloud migration to Platform-as-a-Service offerings. Dissemination events help gather feedback of the application of the proposed concept to the cloud migration of BI application systems and are complemented with workshops and informal feedback. Finally, the last evaluation scenario —cloud migration of collaborative networks systems— gathers feedback through questionnaires and conference presentations.

5. Requirements gathering

The requirements gathering phase documented in this chapter employs two research methods[435]: literature analysis (see Section 5.1) and case studies (see Section 5.2), as Figure 5.2 highlights. Both research methods converge into the result of the requirements identification: thirteen requirements (see Figure 5.1). The analysis of literature on cloud migration groups the requirements around five groups of literature (Sections 4.2 to 4.6) to then be scrutinised in the context of three potential categories of application systems to apply the proposed cloud migration concept to cloud migrating: personal services marketplaces, collaborative application systems, and knowledge management platforms. Analysing these three case studies help identify further corroborating or weakening arguments related to the gathered requirements from the praxis of cloud migration.

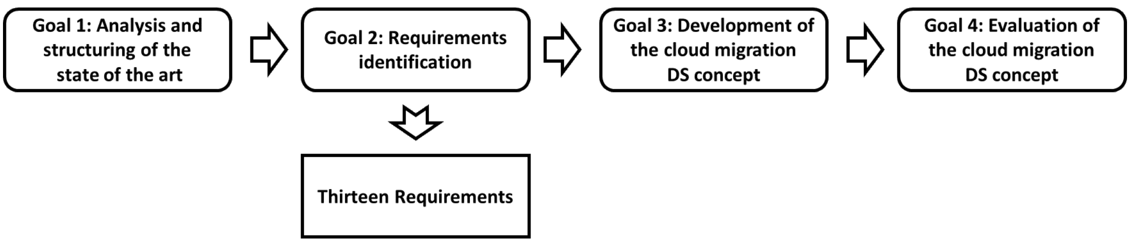

Figure 5.1: Result of the second goal[436]

Based on the requirements gathered here, the proposed cloud migration concept will be designed and built (See Chapter 6). To evaluate the concept in further

[435]Table 4.7 describes the used research methods and processes in Section 4.7 of Chapter 4
[436]Own illustration

real-life scenarios (different from the case studies used for requirements gathering in this chapter), the concept will be prototypically implemented[437] as Figure 5.2 shows in grey.

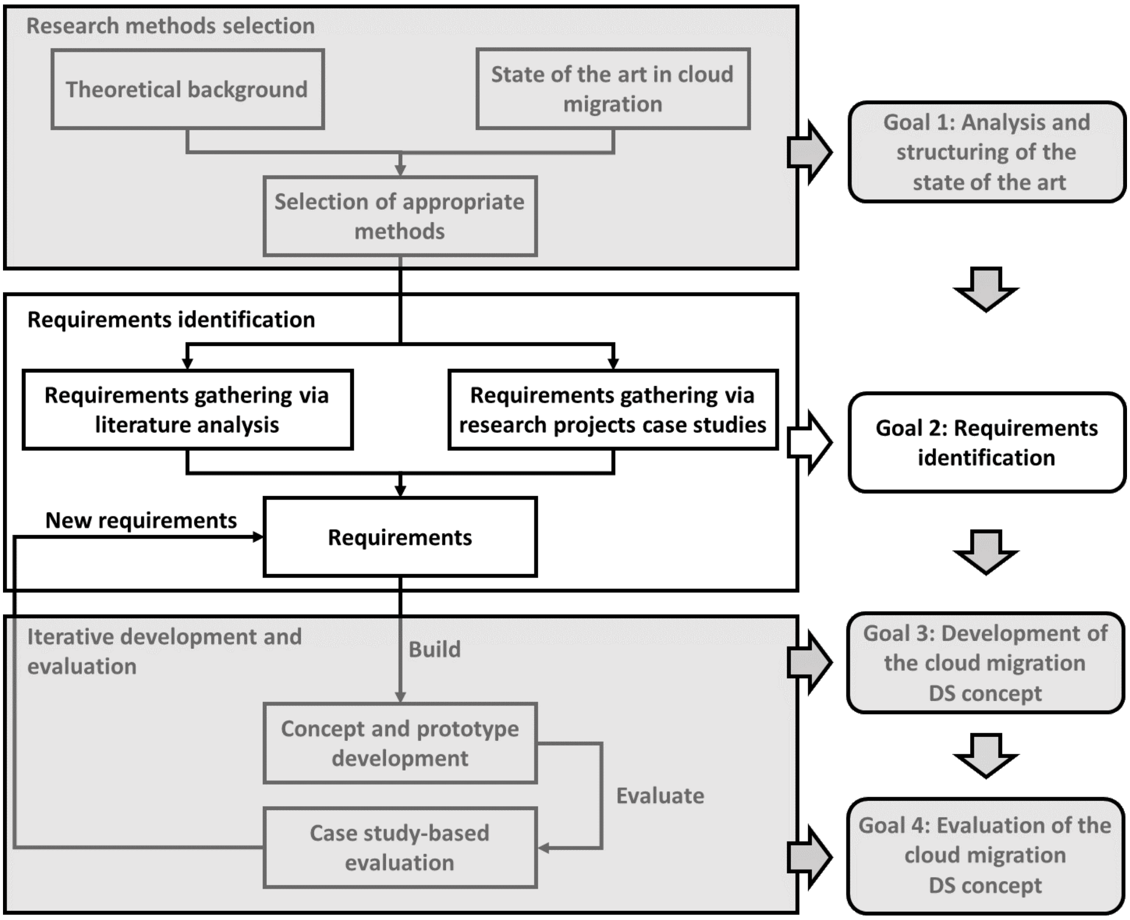

Figure 5.2: Research design (part 2)[438]

5.1 Requirements gathered by analysing the state of the art

Based on the analysis of the state of the art of cloud migration (industrial and research contributions alike) performed in Chapter 4, thirteen requirements are formulated for the design of the cloud migration concept. This section elaborates on the rationale of how these requirements emerge. Each sub-section below refers to the literature field that drove gathering each requirement: from the state of the art of cloud migration support, to research on modelling on-premises application

[437]See Chapter 7 to read the scenario-based evaluation of the proposed concept
[438]Own illustration

systems, cloud migration criteria, or cloud environments. Because the line between requirements emerging from the literature on cloud migration support and cloud and inter-cloud architectures is fuzzy, the requirements stemming from the literature in Sections 4.2 and 4.5 are grouped in one here.

5.1.1 Requirements for cloud migration support: cloud and inter-cloud architectures

Two requirements stem from the analysis of the state of the art related to cloud migration support for cloud and inter-cloud architectures[439]. The state of the art of cloud migration and cloud migration support use, at times, domain-specific modelling languages to specify the provisioning and deployment of multi-cloud systems[440] because using models and model transformation as their primary assets help them improve the productivity and quality of their contributions. This is applicable to the proposed concept to, in addition, empower domain experts[441] in a cost-effective manner while reducing errors, bridging the gap between business and IT, and flexibly adapting to changes in business requirements. More importantly, domain-specific modelling languages allow for capturing the variability within cloud environments at the different abstraction levels of the cloud stack.

R1 — Requirement 1

The proposed cloud migration DS concept should model the cloud deployments at two levels, in both provider-independent and -specific manners including the particularities of the Infrastructure-, Platform-, and Software-as-a-Service offerings and configurations in multi-cloud environments.

Vendor lock-in problems vary in severity and frequency depending on the service model used: IaaS, PaaS, or SaaS. Accordingly, the efforts to port an application system or software components to a cloud environment service model vary; and so

[439] This is the part of the requirements gathering phase that focuses on research question 1 based on Sections 4.2 and 4.5 of Chapter 4: how can a decision support concept assist organisations in how to migrate their application systems to cloud environments?

[440] The research works addressing this challenges stem from the multiple model-driven research projects (such as PaaSage, MODAClouds and Remics) and research works (Sections 4.2 and 4.5 in Chapter 4)

[441] Domain-specific modelling languages let domain experts focus on their expertise given that they enforce separation of concerns and skills: cf. Brambilla et al. (2017), pp. 7–9, 57–58;

does cost. Standardisation reduces both. In the case of IaaS offerings, the maturity and usefulness of existing standards have facilitated solving the vendor lock-in by working at the level of VMs to use the Open Virtualization Format (OVF) to port software runtime environments, configurations, and APIs to cloud environments[442].

Standardisation activities for the PaaS service model were not that advanced at the time of starting this dissertation and many standardisation actions took place contemporaneously to this dissertation[443]. As for the SaaS abstraction level, it usually entails moving data between providers.

R2 — Requirement 2

The proposed cloud migration DS concept should assess the effects of migrating to IaaS, PaaS, SaaS, or multi-cloud in terms of vendor lock-in issues and the potential to use the cloud services they offer[444].

The efforts and costs to overcome vendor lock-in through standardisation and migration depend on the selected cloud service model. Likewise, the cloud services offered depend on the selected cloud environment configurations. The proposed concept might consider ranking higher those cloud migration alternatives assessed to present less vendor lock-in issues, needed off-the-shelf cloud services (for example: e-mail, calendar, cloud storage) or cloud management and deployment services (such as Rightscale).

Cloud migration projects are affected by the effects, in performance and architecture, of using brokering techniques for multi-cloud or other inter-cloud options: peer-to-peer federated inter-cloud architectures[445], independent inter-cloud architectures, or multi-cloud deployments. Multi-cloud deployments use multiple cloud providers independently and do not necessarily interconnect nor do they share their infrastructure. Instead they use a broker and multi-cloud libraries to mediate[446].

[442] The Distributed Management Task Force (DMTF) proposed the Open Virtualisation Format (OVF) as the open format to package and distribute virtual appliances or software running on VMs. OVF files explain how to deploy, manage, and run a virtual appliance on a VM

[443] See the mOSAIC and Contrail projects as Sections 4.2 and 4.5 explain in Chapter 4

[444] The first case study in Section 5.2 explains *R2* from the perspective of cloud-based personal services

[445] Additionally to Contrail (see Section 4.5 in Chapter 4), other projects address these challenges including Optimis, Arjuna, Stackato, mOSAIC, CloudScale and its own tools, or Artist. Reservoir, Open Cirrus, or Optimis project proposed the peer-to-peer federations

[446] Section 4.5 in Chapter 4 explains how to use brokers and rely on multi-cloud libraries and cloud management tools

Table 5.1: Literature from which the requirements for cloud migration support for cloud and inter-cloud architectures emerge

Reqs.	Literature involved in gathering the requirement
R1	**Cloud stack**: cf. Gholami et al. (2016), pp. 31–40; Ahmad and Babar (2014a), p. 7; Ahmad and Babar (2014b), p. 1; **Multi-cloud**: cf. Di Nitto et al. (2016), pp. 61–67; Petcu et al. (2013a), p. 1417; Moscato et al. (2017), pp. 467–471; Gouvas et al. (2015), pp. 217–221; **Software evolution**: cf. Bruneliere (2018), pp. 71–79; Menychtas et al. (2013), p. 424; Alonso et al. (2017), pp. 397–404
R2	**IaaS, PaaS, SaaS**: cf. Alfraihi and Lano (2017), p. 451, 453–456; Juan-Verdejo and Baars (2013), pp. 35–36; Mohagheghi and Sæther (2011), pp. 507–510; Khusidman and Ulrich (2007), pp. 1–7; Krasteva et al. (2013), p. 1; Mohagheghi et al. (2010), pp. 3–6; **Metrics**: cf. Wang et al. (2015), pp. 321–323; Contrail Consortium (2012), p. 16; **Lock-in**: cf. Duncan (2018), pp. 1–6; Cheung and Weber (2015), pp. 32–38; Koehler et al. (2010a), pp. 3–4; Hilley (2009), pp. 28–30; Anandasivam (2010), p. 72; Abdulateef et al. (2020), pp. 2–4

5.1.2 Requirements for on-premises application system modelling

The analysis of the state of the art suggests that considering the architecture of application systems prior to the cloud migration (the status-quo of the application system target for migration) informs the cloud migration and is an input of the decision-making process. The application system's design, requirements, and constraints inform the cloud migration[447].

R3 — Requirement 3[448]

The proposed cloud migration DS concept should let organisations model the application systems to be cloud migrated. They should be allowed to specify their technical requirements prior to the cloud migration including the application systems architecture (both application system components and component-to-component dependencies) and the application systems requirements and constraints.

Once the proposed concept fulfils *R3*, it might use the information about the on-premises application system to generate partial cloud migration alternatives

[447] cf. Frey et al. (2018), pp. 12–19; Frey (2014), pp. 141–145; Babar and Chauhan (2011), p. 50; Kaisler and Money (2011), pp. 1–4; Gustavo et al. (2004), pp. 91–92

[448] Requirement *R3* stems from the literature analysis of on-premises application system modelling to address research question 4: which requirements and characteristics of the system prior to the cloud migration do organisations consider in the decision-making process?

whereby the premises local to the organisation can still be used to run computation or store data. How the functionalities of an application system are distributed among different software components and how those components interact is, according to the state of the art, fundamental in the decision of how to distribute (software and data) components and the links between them. Likewise, the application systems requirements and design constraints will probably have to be respected after the cloud migration.

Table 5.2: Literature from which the requirement related to on-premises application system modelling emerge

Req.	Literature involved in gathering the requirement
R3	**Cloud reconfiguration**: cf. Frey et al. (2018), pp. 12–19; Frey (2014), pp. 141–145; Rai et al. (2015), pp. 6–9; Babar and Chauhan (2011), p. 50; **UML**: cf. Platt and Thompson (2019), pp. 1452–1456; Torre (2016), pp. 53–54; OMG (2015), pp. 21–35, URL see Bibliography; **Model-driven**: cf. Bruneliere (2018), pp. 42–55; Bruneliere et al. (2014), p. 1012; Madiot (2010), pp. 12–18; Arcelli Fontana et al. (2017), p. 1122; Pérez-Castillo et al. (2011), pp. 525–529; Koltun et al. (2019), pp. 687–692; Bruneliere (2011), pp. 21–24

5.1.3 Requirements for cloud environments modelling

Based on the lessons learnt after analysing the state of the art, this dissertation reasonably assumes that the proposed concept uses cloud environment models to rank the cloud migration alternatives based on the information about the selected cloud environment, the (previously mentioned) requirements and constraints specified in the on-premises application system model, and the set of cloud migration criteria defined by the organisation (see next sub-section)[449].

R4 — Requirement 4

The proposed cloud migration DS concept should allow for modelling the cloud environments and its configurations. This includes allowing for specifying the technical details of the cloud offerings that the proposed concept should use to assess the score of a cloud migration alternative for a cloud migration criterion based on metrics defined for that criterion.

[449] The analysis shown in Section 4.4 of Chapter 4 informs the requirements gathering phase addressing research question 3: what are the criteria or requirements organisations consider to select cloud environments to migrate their application systems?

Table 5.3: Literature from which the requirement related to cloud environments modelling emerge

Req.	Literature involved in gathering the requirement
R4	**Cloud modelling**: cf. Bergmayr et al. (2018), p. 1; Ferry et al. (2018), pp. 20–21; Sáez et al. (2018), pp. 19–20; Woitsch and Utz (2015), p. 128; Brandtzæg et al. (2012), pp. 213–214; Ferry et al. (2013b), p. 892; Ardagna et al. (2018), pp. 11–24; MODAClouds Consortium (2012), p. n/a, URL see Bibliography; **Feature models**: cf. Wittern and Zirpins (2016), p. 560; Wittern et al. (2012), p. 127; Quinton et al. (2013), pp. 21–23; Ferry et al. (2013a), pp. 38–40; Benavides et al. (2005), pp. 138–140; Leitner et al. (2019), pp. 340–344;

5.1.4 Requirements for cloud migration criteria

The analysis of cloud migration criteria includes the state of the art of the factors driving organisations in their decision-making processes to cloud-enable their software application systems[450]. The requirements in this section capture the fact that cloud migration criteria are interrelated factors affecting each other and, accordingly, the need to assist organisations to trade some of these criteria off with others to achieve their goals related to the cloud migration.

This dissertation classifies the cloud migration criteria in four categories[451] that affect the cloud migration projects of organisations differently over time according to their self-defined hierarchy of cloud migration criteria. Cloud migration criteria are quite volatile as they might even vary over time for a single organisation, which calls for flexibility.

R5 — Requirement 5

The proposed cloud migration DS concept should allow organisations to model their cloud migration criteria in a flexible, extensible, and customisable manner. Likewise, it should let organisations describe cloud migration criteria in a easy-to-use and semi-assisted fashion[452].

Part of the cloud migration research tries to classify and address the decision-making aspects and challenges guiding how to adapt an application system to the

[450] The requirements under this classification target research question 2 based on Section 4.6 of Chapter 4: what are the decision-making criteria driving organisations to decide how to migrate their application systems to cloud environments?

[451] Business- and economics-related, technical, provider-related, and security and privacy cloud migration criteria

[452] In order to minimise the errors and save modelling costs the proposed concept.

new cloud computing environment. The following four sub-sections elaborate on the different cloud migration criteria mentioned in this dissertation to assist the decision of how to migrate application systems to both cloud environments and the computing premises local to organisations.

R6 — Requirement 6

The proposed cloud migration DS concept should offer a standardised method to let organisations measure and compare the potential cloud migration alternatives without requiring too much effort from them. It should allow organisations to model how they prioritise the cloud migration criteria, how they trade some criteria off with others, and how these criteria depend on one another.

To fulfill *R6*, the proposed concept might have to let organisations flexibly specify metrics to measure some of those criteria.

Table 5.4: Literature from which the requirements related to cloud migration criteria emerge

Req.	Literature involved in gathering the requirement
R5	**Multi-dimensional criteria**: cf. Villari et al. (2016), pp. 76–80; cf. Hajjat et al. (2010), p. 247; **Model-driven**: cf. Bruneliere (2018), pp. 71–79; cf. Menychtas et al. (2013), p. 424
R6	**Metrics for criteria**: cf. Ortiz et al. (2018), pp. 555–557; cf. Siegel and Perdue (2012), pp. 411–415; cf. Garg et al. (2011), pp. 212–214; cf. Tran et al. (2009), p. 1378; **AHP**: cf. Tzeng and Huang (2011), pp. 143–156; cf. Saaty (1977), pp. 245–252

To lower the effort organisations need to put into using the proposed concept, the concept might offer a "default" way of using it. This could mean delivering sort of a pre-filled cloud migration concept that organisations can tailor to their needs. The concept might come with a pre-configured set of metrics and a prioritised list of cloud migration criteria.

Business- and economics-related cloud migration criteria

Cloud migrating an application system affects how organisations do business and the cost of offering their services. According to literature, the economics-related criteria shape the cloud migration decision in particular according to the pricing models offered by IaaS, PaaS, or SaaS cloud offering providers. The pricing model of the selected cloud migration affects the cost of each cloud migration alternative. Although cost is usually an important factor in cloud migration decision making, it affects organisations differently depending on their particularities. For example, a financially stable large organisation might have more cash at its disposal and hence be less restricted to investments than a struggling start-up. The importance of cost to cloud migrate an organisation's application system depend on the organisation and the cloud migration project.

R7 — Requirement 7

The proposed cloud migration DS concept should offer decision makers mechanisms to compare cloud migration alternatives in terms of how well these alternatives do in terms of business and economics-related cloud migration criteria (such as cost and business agility) by allowing the translation of their key business objectives to cloud migration criteria.

By letting organisations translate their key business objectives to cloud migration criteria. The proposed concept could let them model and compare the business agility and cost they could achieve with the potential cloud migration alternatives. Allowing for modelling the business- and economics-related aspects in cloud environments includes modelling the pricing strategy of the different configurations in IaaS, PaaS, or SaaS service models.

Economics-related cloud migration criteria, such as cost, are multi-dimensional and interdependent. Cost might emerge from a combination of different criteria and not as a direct result of how much cloud environment providers charge for their services. For example, cost might emerge from efforts to adapt the target application system, to migrate of data, or to train the organisation's workforce. Additionally, other criteria might restrict the search space of cloud migration alternatives; for example, the staff of an organisation might be well trained in particular technologies that restrict the cloud environments to which that organisation is ready to migrate its application systems. While to some organisations pricing might be the most fundamental factor driving the cloud migration. Others might trade cost off with some other cloud migration criteria; for example, organisations would have to consider cloud migration criteria assessing the potential risks of not timely delivering their services if they had to devote staff to the cloud migration project rather than to its daily development activities.

Cloud migration criteria change over time as the needs of organisations evolve. Especially, for the business- and economics-related criteria: demand behaviour and key business objectives such cost efficiency, shorter time-to-market, or minimising the number of violations to SLAs[453].

Table 5.5: Literature from which the requirements related to business-related and economic cloud migration criteria emerge

Req.	Literature involved in gathering the requirement
R7	**Conflicting criteria**: cf. Chang et al. (2016a), pp. 24–27; Marston et al. (2011), p. 180 **Cost**: cf. Allan et al. (2016), pp. 14–17; Jennings and Stadler (2015), pp. 567–573; Khajeh-Hosseini et al. (2010b), p. 453; Koehler et al. (2010b), p. 16; Haug et al. (2016), pp. 292–303; Phaphoom et al. (2015), pp. 169–173; Hauck et al. (2010), p. 5; Juan-Verdejo et al. (2014b), pp. 467–468; Villari et al. (2016), pp. 76–80; Hajjat et al. (2010), p. 247; **Other criteria**: cf. Gholami et al. (2017), pp. 105–113; Linnenluecke (2017), pp. 5–12; Chang et al. (2016b), pp. 158–160; Mohammadi et al. (2013), pp. 228–229

Technical cloud migration criteria

From a technical standpoint, organisations running their application systems in cloud computing infrastructures can achieve improvements in terms of, among others[454], availability, performance, scalability, elasticity, accessibility, computing efficiency, power consumption, response time, throughput, utilization, latency, or reliability with respect to their previous on-premises deployments. However, as stated by *R6*, configuring application systems to gain in some of these technical cloud migration criteria might come with losing with respect to another one. Hence, organisations trade some criteria off with others. An activity in which they can be assisted.

According to the state of the art, the quantitative assessment of cloud migration criteria can facilitate comparing cloud migration alternatives based on measurable technical metrics; sort of Quality-of-Service attributes. Additionally, this qualitative assessment allow for assessing non-functional cloud migration criteria and

[453] See Section 4.6 of Chapter 4
[454] Section 4.6 in Chapter 4 explains the technical cloud migration criteria

their effects: disaster recovery (with a recovery cloud), fault-tolerance, sustainability, ease and agility to deploy, and ease of testing together with performance at runtime, multi-tenancy, power management, availability aspects, replication (for higher availability), and distributed data usage. Nevertheless, one can not consider quantitative cloud migration criteria independently of qualitative criteria such as business-related cloud migration criteria. For example, scalability has got implications as to how an organisation grows or organises its workforce, which entails that organisations might trade off those interrelated criteria with each other. While technical key performance indicators (KPIs) might be relatively easy to compare due to their objective nature, subjective cloud migration criteria are more prone to represent reality differently depending on the stakeholder assessing the cloud migration criteria.

R8 — Requirement 8

The proposed cloud migration DS concept should allow for modelling both the technical quantitative cloud migration criteria of a cloud migration alternative as well as the subjective or qualitative ones. It should offer flexibility and customisability to let organisations model abstract qualitative cloud migration criteria and capacity to specify the qualitative cloud migration criteria to the detail.

The proposed concept might have to use the information described in the on-premises application system model, its constraints, and the particularities of the selected cloud environments to specify in a low level of detail the qualitative criteria.

The assessment of quantitative cloud migration criteria by the cloud migration concept depends on the on-premises and cloud-migrated architecture as well as on the cloud migration criteria in a non-orthogonal way. For example, while elasticity could improve computing efficiency so might applying adaptation mechanisms such as implementing an operation in a particular way. An adaptation of an operation might have more effects on computing efficiency than higher levels of elasticity.

Cloud migration at run time and performance assessment are entire dissertations on their own. Therefore, the scope of these dissertation restricts both to rough performance assessment and cloud migration of application systems at design time. However, given that fine-tuning performance assessment (with the potential future appearance of easier-to-use research and industrial contributions) would improve the cloud migration assistance; this dissertation paves the way to easily integrate those contributions. Future research related to this dissertation (but out of its scope) might intend to analyse the complexity and dynamicity of cloud deployments at run time; thus, performing runtime checks entailing deciding how often to trigger adaptations as a result of the analysis and how to actually adapt a running architecture. Both lines of research need a flexible, extensible, and customisable cloud migration concept; that is, requirement *R5*. Building the decision support concept in a modular manner would make it ready to integrate sophisticated performance prediction mechanisms. Additionally, the design of the proposed concept might influence the easiness to integrate in the future, and hence out of the scope of this dissertation, mechanisms to assess adaptations to the architecture of the cloud-enabled application system at run time.

R9 — Requirement 9

The proposed cloud migration DS concept should consider the design time cloud migration and will offer the architecture to incorporate in the future runtime aspects to the assessment of metrics.

Table 5.6: Literature from which the requirements related to technical cloud migration criteria emerge

Req.	Literature involved in gathering the requirement
R8	**Qualitative criteria**: cf. Juan-Verdejo et al. (2014b), p. 467; Garg et al. (2011), pp. 212–214; Herbst et al. (2013), pp. 23–24; **Technical criteria**: cf. Bergmayr et al. (2018), pp. 497–512; Štefanič et al. (2018), pp. 3–7; Fazli et al. (2018), pp. 5154–5159; Mogul et al. (2017), pp. 2–4; Nabi et al. (2016), pp. 54–59; Rai et al. (2015), pp. 6–8;
R9	**Design vs run time**: cf. Hauck et al. (2010), pp. 11–14; Juan-Verdejo et al. (2014b), pp. 472–473; Phaphoom et al. (2015), pp. 167–172;

Provider-related cloud migration criteria

The selected cloud provider might change how organisations organise themselves and adapt some of their business processes to refefine how they collaborate with other business entities and their organisational boundaries[455].

[455]See Section 4.6 in Chapter 4

This is a double-edged sword with negative consequences —were, for example, these changes not done smoothly— or positive effects —as, for example, innovation might increase as technological and human resources are freed from computing-related tasks (undertaken now by the cloud provider) and shifted to other tasks. Sharing or collaborative models associated with cloud-based processes imply that mistakes, achievements, and changes are immediately transparent to others. This might change how organisations and their individuals interact, collaborate, and behave. All in all, this relates to research question 2[456] and encompasses that organisations might benefit from using mechanisms to assess the organisational benefits and risks of cloud migrating their application systems. Organisations might benefit from assistance to plan the adoption of cloud-based infrastructures.

R10 — Requirement 10

The proposed cloud migration DS concept should allow organisations to assess the changes their organisations undergo depending on the cloud migration decision. Power and responsibility shifts, the impact of the collaborative model, and changes in organisational structures or processes.

R10 includes allowing organisations to weigh power and responsibility shifts from the IT department to other individuals, departments, or even organisations as well as letting them assess the impact of the collaborative model in their organisations, organisational structures, and business processes.

The vendor lock-in problem might be one of the most important provider-related cloud migration criteria. It happens when the cloud selection represents a single point of failure. Organisations need to develop strategies to cope with their cloud provider going down or out of business; in addition to coming to grips with how the IT department faces failures on the cloud provider side. Additional challenges related to the vendor lock-in problem include: how to react to outages, to low QoS that non-stringent SLAs allow, or to unilateral changes in pricing or technologies. Under such circumstances, cloud consumers might think whether they can take

[456] Explained in Section 1.2 of Chapter 1: What are the decision-making criteria driving organisations to decide how to migrate their application systems to cloud environments?

their data with them as they move to a competing service provider and what do they need to do this[457].

R11 — Requirement 11

The proposed cloud migration DS concept should pay special attention to the vendor lock-in problem depending on the abstraction level used: IaaS, PaaS, or SaaS.

Table 5.7: Literature from which the requirements within the provider-related cloud migration criteria emerge

Req.	Literature involved in gathering the requirement
R10	**Organisational changes**: cf. Mahy et al. (2016), pp. 12–24; cf. Palos-Sanchez et al. (2019), p. 8–12; cf. Suicimezov and Georgescu (2014), pp. 830–835; **Regulation**: cf. European Union (2016), p. 1; cf. Kaisler and Money (2011), p. 1;
R11	**Vendor lock-in**: cf. Duncan (2018), pp. 1–6; cf. Cheung and Weber (2015), pp. 32–38; cf. Koehler et al. (2010a), pp. 3–4; cf. Hilley (2009), pp. 28–30; cf. Anandasivam (2010), p. 72; cf. Abdulateef et al. (2020), pp. 2–4; **Trustworthiness**: cf. Brandenburger et al. (2017), pp. 4–6; cf. Noor et al. (2016), pp. 34–38; **Data governance**: cf. Ongowarsito et al. (2017), pp. 1–5; cf. Youseff et al. (2008), pp. 7–8

The proposed concept might take into account data ownership concerns to fulfil *R11*. This could be done by either disregarding or giving lower rating to cloud migration alternatives that use cloud environments which do not comply to standards of data ownership, data portability, and interoperability.

Security and privacy cloud migration criteria

Security and privacy concerns remain one of the fundamental obstacles to cloud migration[458]. Sometimes the risk organisations take to move sensitive data or computation to a different computing environment might exceed the stakeholders' risk tolerance or the achieved improvements. Organisations need to re-design or

[457] See the sub-section explaining the different provider-related cloud migration criteria under Section 4.6 within Chapter 4. In particular the works on organisational and provider-related aspects, vendor lock-in, and outages.

[458] Some of the security-related sub-criteria considered in this dissertation, together with their metrics in Chapter 6, stem from related research works analysed in the span of the dissertation requirements gathering phase and the analysis of the related work in Section 4.6 in Chapter 4 phase

re-architect their target application system to fulfil state-of-the-art security requirements in the new computing environment that maintain or increase the security levels they guarantee. As security and privacy are transversal aspects that affect the entire application system, cloud migration security critera relate to many other cloud migration criteria such as the design effort —hence cost— or other performance-related metrics[459] (providing a more secured environment might entail introducing overheads that lower performance).

R12 — Requirement 12

The proposed cloud migration DS concept should allow organisations to assess the security risks they face and their data sovereignty depending on the selected cloud environment and the cloud-enabled architecture. It should assess the privacy and security sub-criteria: trustworthiness of the cloud environment, data integrity, data privacy, data security, compliance to security standards.

Security provision might entail changes in the architecture of the cloud-enabled application system, which would require incurring into some effort in terms of cost and time. *R12* includes allowing organisations to assess to which extent the selected cloud environments (and their cloud-migrated application systems) comply to security standards.

The proposed concept considers options to cloud migration that keep sensitive data local to the organisation with data ownership. Previous research works motivated this dissertation to consider quantitatively comparing the performance (in terms of time and space) by having the proposed cloud migration concept weigh the different cloud migration alternatives that encrypt sensitive data before cloud migrating them. The proposed concept could accordingly estimate the performance loss to help adjusting the QoS within the SLA prior to the actual migration.

[459]cf. Sciancalepore et al. (2018), pp. 5190–5192; Hu and Klein (2009), pp. 735–736

R13 — Requirement 13

The proposed cloud migration DS concept should present partial (deployments to cloud environments and, if needed, the local premises) cloud migration alternatives that, at least, keep the security levels provided by the application system prior to the cloud migration.

The proposed concept could use the specified data sensitivity of the data to compute data locality restrictions depending on that and the security provided by the selected cloud environment (that could also be modelled with the proposed concept) to generate cloud, local, or partial (deployments to cloud environments and the local premises) cloud migration alternatives. Those alternatives could be ranked according to the cloud migration criteria. This refers to the generation of multiple cloud migration alternatives using cloud environments and the local premises to rank them according to how the organisation trades off performance and security. Security provisioned by using security mechanisms and abiding by security standards.

Requirement *R13* refers to modelling the security provided by the selected cloud environment. This means allowing for modelling the security mechanisms existing in the cloud environment configurations offered by cloud providers. This information could be used to generate cloud migration alternatives based on the available security provision, the component-to-component interactions (that indicates the data exchanges happening in an application system), and the sensitivity of those data.

5.2 Requirements analysis via case studies

The three case studies described below are part of the analysis to help identify further corroborating or weakening arguments related to the gathered requirements from the praxis of cloud migration. Accordingly, Table 5.9 lists all the requirements (gathered via literature analysis) that are relevant to each of the three case studies (marked with an X in the intersection of column *i* for requirement R*i* with the row with the three case studies). The case studies were selected because they are cloud migration projects to achieve improved collaboration and business

Table 5.8: Literature from which the requirements related to the security and privacy cloud migration criteria emerge

Req.	Literature involved in gathering the requirement
R12	**Data sovereignty**: cf. Ali et al. (2015), pp. 357–366; US-NIST (2009), p. 2; Hu and Klein (2009), p. 738; **Effort assessment**: cf. Chang and Ramachandran (2015), pp. 145–151; Subashini and Kavitha (2011), p. 1; Juan-Verdejo et al. (2014b), p. 467; **Other criteria**: cf. Bauer et al. (2017), pp. 147–153; Oliveira Albuquerque et al. (2016), pp. 3741–3753; Yumerefendi and Chase (2007), pp. 1–2; Brandenburger et al. (2015), p. 7; Cachin et al. (2009), pp. 81–83;
R13	**Data sensitivity**: cf. Xu and Zhao (2015), pp. 1344–1349; European Union (2016), p. 1; Kaisler and Money (2011), p. 1; Khan and Hamlen (2012), p. 175; **Partial migration**: cf. Rahimi et al. (2016), pp. 150–156; Subashini and Kavitha (2011), pp. 5–6; Santos et al. (2009), pp. 1–3;

agility via running marketplaces, collaborative application systems, and knowledge management application systems on cloud environments.

1) Marketplaces can be optimised by running them in cloud environments to optimise cost in addition to making marketplaces more attractive as they offered cloud providers in the marketplace access to a larger consumer. That, in turn, makes marketplaces more interesting as when the number of providers increase so does competition among them either in quality or price which attracts more consumers. Likewise, a larger number of marketplace consumers incentivise providers to join the marketplace and access a larger pool of consumers. Creating a win-win relationship between them.

2) As for migrating collaborative application systems to cloud environments, it is beneficial due to the potential to enhance communication and collaboration across entire supply and value chains. Consumers of cloud-based collaborative application systems can co-create in global networks that offer more value than smaller networks given the increased number of opportunities and synergies that may appear.

3) Knowledge management platforms running on cloud environments have also got the potential to adjust their cost to the consumption. In cloud-based architectures, they might offer interchangeable components in a centralised manner to different organisations according to their requirements.

Table 5.9: Requirements relevant to each of the three case studies

Case study	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	R11	R12	R13
Personal serv.[460]	X	X		X	X	X	X			X	X		
Collab.[461]	X		X	X	X				X	X	X	X	X
Know mgmt.[462]	X	X			X		X	X		X	X		

5.2.1 Cloud migrating personal services marketplaces

Personal services are commercial services addressing the personal and professional needs of customers. They include services such as: accounting and finances, business and legal consulting, graphic design, multimedia productions, communications and public relations, recruitment, technical authoring and translation, and training and certification.

As personal services have undergone an intense process of digitalisation, they have got the potential to scale and offer their services to a larger consumer base. The app paradigm brought about new business models to offer personal services over marketplaces as the intermediary. Consumers can discover the personal service apps published in the marketplace and install them.

Migrating personal services marketplaces to cloud environments entails a change in business models to enhances collaboration and scalability[463]. Cloud-based marketplaces have got the potential to provide value added by combining cross-platform apps offered through online marketplaces to provide personal services. Cloud-based marketplace consumers and providers profit from the role of an independent third party that markets apps and cloud services. An intermediary that distributes those services while offering extra functionalities like on-line discovery and multi-purpose service consumption. The following are four important aspects affecting how to cloud-enable marketplaces of personal services.

[460] Cloud-based personal services case study

[461] Cloud-based collaboration case study

[462] Cloud-based knowledge management case study

[463] Adrian Juan-Verdejo (the author) and Rustem Dautov presented the cloud migration of personal services marketplaces through a poster in 2015 —cf. Pantelopoulos et al. (2015), pp. 1— in the context of the Invent project — INVENT Consortium (2015), p. n/a, URL see Bibliography

Cross-device apps engineering. The complexity, maintenance effort, and economic challenges increase when developing solutions for multiple target computing devices and software platforms[464]. Cloud migration criteria of personal service marketplaces describe the need for scalable run-time environments to deploy these cross-device apps so that they become seamlessly available to any device connected to the Internet.

Design of personal services marketplaces. Software is not the actual commodity that these marketplaces offer but the means to an end. Consumers are interested in the service behind the app, which can be recommended based on the user's preferences[465].

App on-boarding environment. The on-boarding environment bridges the delivery platform and the distribution channel. It presents governance problems as the framework scales. Cloud-based application platforms and marketplaces face quality challenges as they let multiple developers on-board new apps and services. The marketplace requires storing and organising all app artifacts and personal service metadata to facilitate deploying apps to the run-time environment and marketplace.

Tailored business models. Intermediary business models to incentivise collaboration across the entire value chain through revenue sharing models and initiatives, tailored value propositions, services co-creation, new structure of partner ecosystems, and cross-organisational business processes.

Concept requirements for migrating personal services marketplaces

Cloud-based marketplaces of personal services motivate the analysis of arguments that support or weaken some of the requirements of the proposed cloud migration concept identified by analysing the state of the art (see Section 5.1). In particular those concept requirements marked with an X in Table 5.9

The research in personal services and marketplaces provided arguments supporting requirements of the proposed related to the need of assisting marketplace providers to deploy the marketplace to different cloud providers; a supporting argument for capturing *R1*. Likewise, a particular instance of *R1* apply to let personal

[464] cf. Messerschmitt and Szyperski (2000), pp. 31–37
[465] As mobile app marketplaces (Apple App Store, Google Play, or GetApp.com) or Amazon do.

service providers model their migration problem when selecting the appropriate services marketplace as a PaaS solution. The proposed concept allows personal service providers to describe this cloud selection requirement —according to *R5* and *R6*— by defining their cloud migration criteria. The services that the marketplace supports —such as distribution and service delivery— could be described per alternative marketplace to which personal services can be migrated to (which supports *R4*).

According to *R2* and *R4*, the cloud migration concept could let marketplace providers model their quality aspects and developers on-board apps and services. In addition, cloud-based personal services marketplaces and personal service providers need to take into account the horizontal *R5*, *R6*, *R7*, *R10*, and *R11* requirements for the cloud migration of their entire ecosystem.

5.2.2 Cloud migrating collaborative application systems

Cloud infrastructures enable modern collaboration as cloud consumers share and co-author data stored in cloud-based data storage. End-to-end collaboration between customers and local suppliers enables the co-creation of service-enhanced products by glocal organisations or enterprises. Enterprises are glocal when they act globally while collaborating with local (service) suppliers to operate and offer service-enhanced products. Thus, value creation stems from globally networked operations. Service-enhanced products are complex products in manufacturing and their associated services; for example, automated buildings or solar energy plants[466]. Mass customisation describes how with the customer co-designs the product features and services of service-enhanced products[467].

The distributed units in a global supply chain use software services to share and exchange information —such as catalogue of products, brochures, process descriptions, best practices, or company profiles— in a collaborative environment. Cloud migrating those services can improve how glocal enterprises collaborate via services available to anybody, at any time, and from anywhere. Cloud-based collaborative networks support co-creation, co-design, and negotiation among all parties to reach a solution by creating a workflow of the negotiated and accepted

[466]As used for the evaluation of the proposed concept explained in Chapter 7)
[467]cf. Juan-Verdejo and Surajbali (2016), pp. 11–23

service-enhanced product. Organisations negotiate to create long-term strategic networks to eventually start goal-oriented or short-term networks. Cloud-based workflows help monitor complex business processes, while including the stakeholders involved in their execution, and intend achieving higher levels of efficiency and collaboration to improve production and processes for product life-cycle support[468].

Concept requirements for migrating collaborative application systems

Cloud-based collaborative networks provide arguments supporting the relevance of the requirements of the proposed cloud migration concept, as Table 5.9 shows, to co-create and collaboratively manage service-enhanced products.

Given the heterogeneous stakeholders and application systems participating in collaborative networks systems, tackling cloud dependencies is fundamental to implement this cloud-based solutions by relying on standards across the entire cloud stack. This reality provides arguments corroborating requirements *R2* and *R9*. Hands-on experience with cloud-based collaborative application systems revealed requirements to minimise vendor lock-in problems as well as interoperability requirements. Both needs provide supporting arguments to *R2*. As cloud-based collaborative application systems do not impose a particular cloud environment, the cloud migration of collaborative networks systems can be based on matching the cloud migration criteria including the minimisation of vendor lock-in problems by selecting cloud environments which abide by particular standards. *R11* and *R1* capture the requirements of interoperability, vendor lock-in, and the use of standards of collaborative application systems.

Applying collaborative application systems to the realm of Internet of Things or Industry 4.0[469] motivates considering to partially migrate these kind of application systems to both cloud environments and the local premises; If geographically-near processing is needed. These two realms of application provide supporting arguments for requirement *R3* as using near computation can improve the performance of some deployments and architectures as well as the security and privacy provision.

[468]cf. GloNet Consortium (2016), p. n/a, URL see Bibliography, the GloNet project —*glocal* enterprise networks focusing on customer-centric collaboration— supported the cloud migration requirements gathering

[469]cf. Juan-Verdejo and Surajbali (2016), pp. 11–23

Collaborative Internet-of-Things application systems share data about complex products stored in the selected cloud environment. Therefore, its collaboration spaces enforce data sharing policies according to the the role of the organisation's members and the process they are involved in. For example, two organisations co-creating a solution might have sharing access to some data while the same two organisations would not have access to data of a business service if they are sequentially providing two of its sub-processes. This is a particular case of *R13* with role- and process-based access control; and of *R9* with security assessment. Although these requirements are related to *R5* and *R4*, Table 5.9 does not mark them as they do not directly emerge from the main needs of cloud migrating collaborative application systems with respect to the proposed concept.

Collaborative application system manage servicing phases across the product life cycle. Differences might exist between what was planned at design time and what actually happened when executing business processes of those phases at run time; this relates to *R9* as a corroborating argument.

The software components conforming the cloud-based collaborative networks system come from different providers. This calls for offering interfaces for data integration in cloud environments. Likewise, it calls for service integration to connect services with both service-enhanced products and for the composition of new services. These cloud environment-dependent decisions support considering *R5* in the design of the proposed concept with regards to which cloud environment to select for the integration and to how to migrate software components to them.

R3 describes the need to model the collaborative application system's architecture and its component-to-component dependencies. This allows for modelling which cloud migration alternatives allow running the cloud-based platform working on top of the Spring-based OSGi run-time environment[470].

Cloud-based collaboration facilitates sharing projects, products, and opportunities but also affects organisations, their structures, and their cross-organisational collaboration —as *R10* captures. Organisations and groups —such as partners,

[470]OSGi is the infrastructure offering the mechanisms to define, deploy, and run independent software modules. The Spring framework together with the OSGi run time environment structure and simplify developing and managing the different application system modules and their dependencies

resellers, providers, distributors, agents, brokers, or customers— have different views of the data according to each member's view or edit permissions and perform cross-organisational scheduling for collaboration across teams. These changes how organisations collaborate and how team members monitor action items, priorities, and member status. Likewise, the cloud migration affects the trust and security within collaborative networks, which corroborates the importance of *R13*.

5.2.3 Cloud migrating knowledge management application systems

Knowledge management application systems help organisations achieve their objectives by offering them processes to use the information they possess. That is, processes to create, share, use, and manage knowledge. This cloud migration project pursues migrating the existing knowledge management application system[471] to the Amazon EC2 cloud offering. The goal is offering a reusable PaaS offering for knowledge management applicable to social businesses. That is, businesses interested in using social knowledge management to generate knowledge based on the interactions between individuals captured via social media. The resulting PaaS offering might have to offer interchangeable components in a centralised manner so that organisations can use them to communicate and engage their target audiences and stakeholders using social networking behavioural standards in a customised manner.

By adopting the cloud-enabled knowledge management application system for their application systems, organisations might benefit from making their organisational support systems always available anywhere and at any time. Additionally, they can improve the agility with their businesses scaling to become economies of scale while also using the off-the-shelf ready-to-use technology (see Figure 5.3) to cut costs and speed up the implementation of their social businesses. The Amazon EC2 cloud environment offering was preselected as well as the Amazon Relational Database Service (RDS[472]), which narrowed down the solution space of cloud

[471] The case study touched upon the requirements to cloud migrate the OrganiK knowledge management application system with the support of the KnowHow project: KnowHow Consortium (2014), p. n/a, URL see Bibliography; cf. KnowHow Consortium (2015), pp. 1–77

[472] cf. Amazon.com, Inc. (2021b), p. n/a, URL see Bibliography

migration alternatives to those using EC2 and RDS. The cloud selection was a design constraint due to the previous expertise of the organisation developing the knowledge management application system with this cloud environment offering.

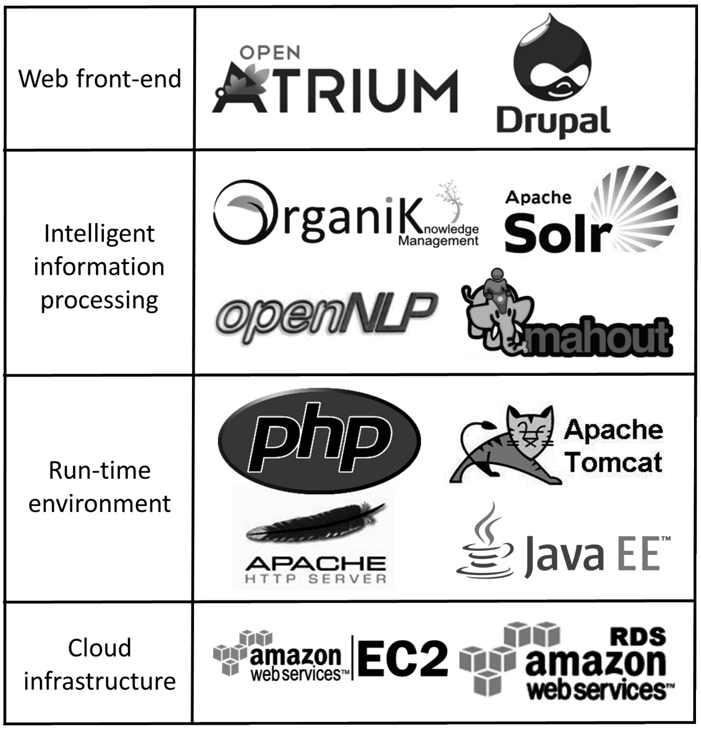

Figure 5.3: The technology stack of the knowledge management application system after cloud migration[473]

The cloud migration criteria for the knowledge management application system support its migration to Amazon to improve on aspects related to which cloud environments offer advantages. That is, collaborative innovation, support to social enterprises, supply chain networking, and social media marketing.

In addition, some cloud migration criteria favoured using off-the-shelf software components —such as Apache Mahout, Solr or openNLP[474]—, packaging them in easy-to-deploy virtual machine images[475], and using an offering with off-the-shelf elastic scalability and automatic load balancing.

[473]Illustration by the KnowHow project consortium: KnowHow Consortium (2014), p. n/a, URL see Bibliography

[474]See Figure 5.3: Mahout is an open source framework for scalable and distributed machine learning algorithms for recommendations, clustering, regression, preprocessing, and distributed linear algebra. OpenNLP are open source libraries for Natural Language Processing

[475]For example, the Amazon Machine Image (AMI) that can be hosted in their resources running on the Amazon EC2 IaaS offering

Concept requirements for migrating knowledge management app. systems

Table 5.9 shows all the requirements of the proposed concept which are important to the cloud migration of the knowledge management application system and the social business application systems running on top. In addition to the cloud migration of the knowledge management application system, four client organisations had to cloud migrate their customised software and therefore tailored the cloud migration to their needs, which indeed brought about corroborating arguments for *R5*, *R7*, and *R8*. The four organisations required assessing the effects of the migration in their organisations, a requirement corroborating *R10* as requirement of the proposed concept.

The cloud migration of the knowledge management application system to the Amazon IaaS offering was pre-designed as a PaaS solution for knowledge management. In that design, the four client organisations build on top of the PaaS offering for knowledge management to offer their SaaS solutions. Both facts corroborate the need of *R11*. One of the requirements of the knowledge management PaaS is to avoid vendor lock-in problems to keep the door open to migrate different components for Web front-ends, intelligent information processing, and the run time environment to different cloud providers. This entailed describing the knowledge management application system in both a provider-independent as well as depending on the cloud infrastructure; a requirement covered by *R1*.

From the SaaS level, the four pilot demonstrators also focused on minimising the effects of vendor lock-in were they to decide to migrate to a different platform by following standards directives and architectural best practices like *R2* describes. The organisations migrating their social products undertake the cloud migration based on the product integration and the overall objectives, which they can model with the cloud migration criteria and corroborates *R5*.

6. Cloud migration decision support concept

This chapter explains the main contribution of this dissertation. It details the core of the cloud migration DS concept and documents how it is designed and developed. Figure 6.1 frames the context of this phase of the dissertation, which builds the concept design.

Firstly, this chapter formalises the cloud migration problem and solution in order to use these concepts to describe the steps and workflow followed by the proposed concept. Next, the architecture of the concept design is detailed next to the description of how cloud migration criteria affect the migration of application systems to IaaS, PaaS, and SaaS offerings. Some cloud environments might offer advanced mechanisms such as soft reservations and cloudlets[476]. Accordingly, this chapter explain how to model these two mechanisms with the proposed concept to include them in the decision-making process. Likewise, the concept could be extended to weigh in the decision making the effects of potential new technologies that cloud environments might support in the future. Finally, the chapter specifies how to tailor the proposed concept to each of the three scenarios[477] used to evaluate it; as Chapter 7 documents.

[476] Sections 6.8 and 6.9 explain the concept tailored to consider soft reservations and cloudlets
[477] Sections 6.10 to 6.14 specify how to tailor the proposed concept to each of the three evaluation scenarios

6.1 Formalisation of the cloud migration problem and solution

Three input models of the proposed cloud migration concept come together to model the cloud migration problem: the on-premises application system model, the cloud migration criteria, and the cloud environments. The proposed concept combines these three modelling approaches because they express different aspects of the cloud migration problem in order to assess the different cloud migration alternatives as output of the proposed concept.

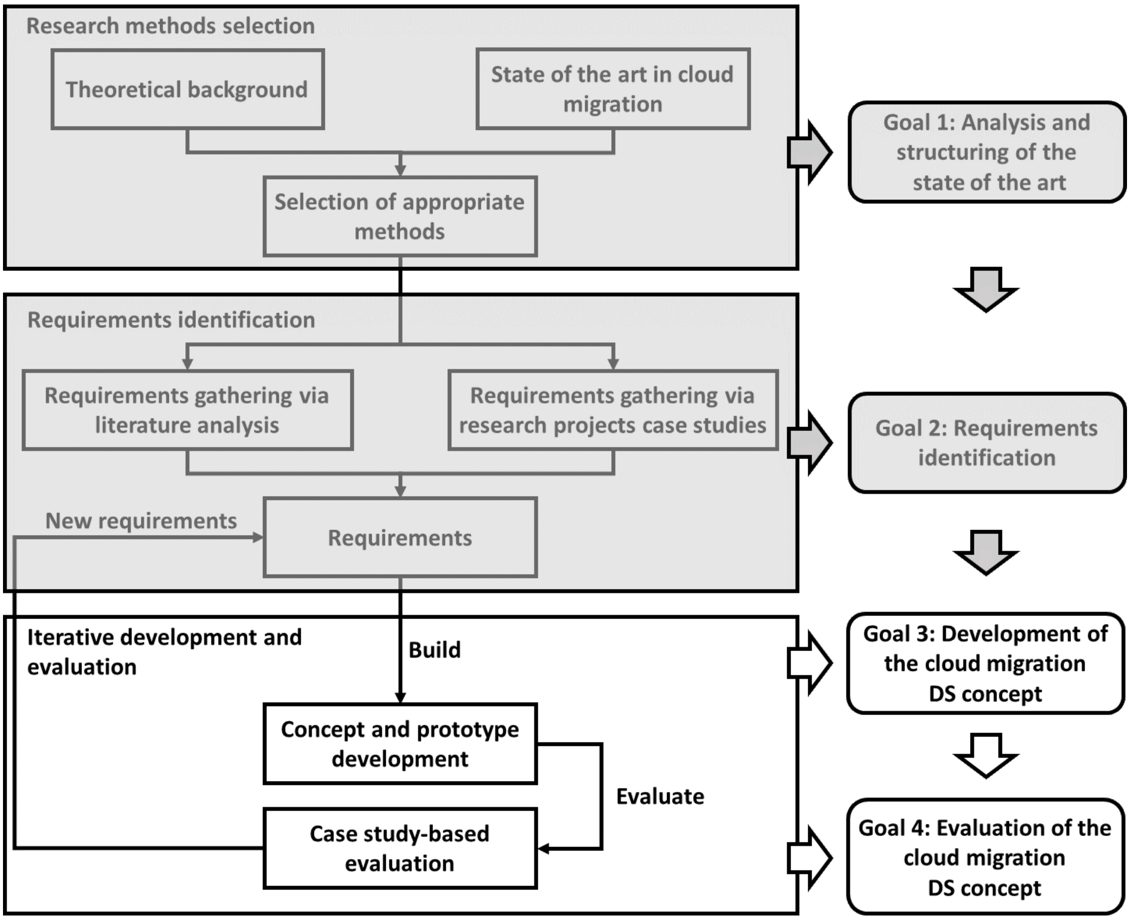

Figure 6.1: Research design (part 3)[478]

A previous paper[479] formalises the details about the input and ouput of the concept in addition to how the proposed research concept models the value of each cloud migration alternative. That is, the solution space. The computation explained in that paper ranks the alternatives to cloud migrate the application system according

[478] Own illustration

[479] cf. Juan-Verdejo et al. (2014b), pp. 467–474

to the three input models described below. The highest-ranked cloud migration alternative is recommended to the organisation responsible for the application system followed by the ordered remaining cloud migration alternatives. For simplicity, this dissertation refers to that paper if the reader needs further details about these formalisations.

6.1.1 Output: Assessment of cloud migration alternatives

The proposed concept calculates the utility of all the potential cloud migration alternatives that split the application system's components to the cloud and the local premises if they respect all the constraints of the application system and those imposed by the organisation. The concept assesses the gains in cloud migration criteria for the cloud-migrated application system by taking into account the objective and subjective cloud migration criteria of the organisation.

6.1.2 Input 1: On-premises application system model

The application system is a composition of several data and computing components that perform their tasks and communicate with one another through links that enable them to transfer data to each other. Application system's components possess several intrinsic properties as well as the relations (edges) between those components. The number of cores a component needs to operate or the memory allocation it needs are examples of components' properties. As for the relations' properties, an example is the data transfer rate two components need in order to provide a satisfactory quality of service.

6.1.3 Input 2: Cloud migration criteria

The cloud migration criteria are specified by the organisation that wants to migrate the application system to cloud environments according to its requirements to migrate it. Cloud migration criteria are aggregated according to a matrix that describes the importance of the cloud migration criteria relative to each other in order to rank the cloud migration alternatives which comply to the set of design constraints. The cloud migration criteria comprise either objective or subjective criteria: a criterion is objective if there is a metric to measure a cloud migration alternative for that specific cloud migration criterion, whereas a subjective criterion is assessed based on human intervention as there is no available metric. The cloud migration concept makes the organisation pairwise compare all cloud migration criterion to one another to obtain the aforementioned matrix describing the relative importance of all cloud migration criteria. The following Section, Section 6.2, shows a more detailed explanation of the steps of cloud migration decision support.

6.1.4 Input 3: Cloud environments

Different configurations of cloud environments are made available to cloud consumers by cloud providers. Cloud environment configurations offer great variability of service models (IaaS, PaaS, and SaaS), storage (relational or non-relational database management systems (RDMS), file systems), networking, resources allocation, orchestration (kubernetes), server-side technologies, deployment or continuous integration capabilities, security-aware mechanisms, and other usual services (e-mail, CRM, or CMS). Many other support capabilities depend on whether cloud providers offer services at the infrastructure, platform, or software level.

6.2 Steps for cloud migration decision support

The proposed concept ranks the generated cloud migration alternatives, present them in an orderly fashion to the organisation cloud migrating its application system, and recommends the highest-ranked cloud migration alternative.

The proposed concept firstly discards unsuitable cloud migration alternatives according to the specified constraints[480] to then calculate local rankings for every cloud migration alternative based on the cloud migration criteria. Local rankings that depend on many properties and sub-properties related to: the on-premises application system system, the cloud migration criteria, and the cloud environment configurations. Hence, this is a multi-criteria decision problem, for which it is arguably assumed that organisations need assistance given the complexity of evaluating the structured hierarchy of cloud migration criteria.

[480]This is formalised in cf. Juan-Verdejo et al. (2014b), pp. 467–474 and illustrated with examples in Section 6.3

Organisations can use the proposed concept to build the hierarchy of cloud migration criteria. The concept presents organisations with pairwise comparisons to weigh each criterion to one another, according to how determinant it is for the success of cloud migrating the target application system. A criterion can be further specialised by defining its sub-criteria and usually conflicts with other criteria. Weighting so many conflicting criteria while taking into account the interdependence between them burdens organisations and might disrupt their judgement. Additionally, some attributes cannot be easily measured and quantified to a numerical value, as even though the proposed concept uses metrics to quantify subjective criteria, sometimes it is not feasible to quantify all criteria and the process requires human intervention. That is, the organisation solve ambiguities around the decision-making process by manually assessing part of the cloud migration process. This manual assessment uses pairwise comparisons of cloud migration alternatives with regards to a subjective cloud migration criterion that the proposed concept uses to compute its global assessment.

In order to easily consider trade-offs and structured relationships between objective and subjective criteria, quantify them, and aggregate them; the proposed cloud migration concept is inspired by the Analytic Hierarchy Process (AHP)[481] to offer a ranking system for the decision-making problem using multiple criteria. The proposed concept uses the AHP to weigh cloud migration criteria expressed in different units/dimensions. Weighting the cost of a cloud migration alternative —expressed in Euros— against the respect to national regulations —expressed in Boolean rationales— is an example of the kind of challenges the Analytic Hierarchy Process helps with. The proposed concept applies five steps to orderly recommending cloud migration alternatives: Step 1, modelling the problem as a hierarchy with a goal, criteria, and cloud migration alternatives; Step 2, prioritising the criteria; Step 3, evaluating the different cloud migration alternatives for every criterion; Step 4, checking the consistency of the judgements; and finally Step 5, coming to a decision.

[481] Section 4.2 explains AHP in Chapter 4

Step 1

Step 1 describes the multi-criteria problem with a three-layered *hierarchy model* —suitable for the application of the Analytic Hierarchy Process— by specifying the goal, cloud migration criteria, and cloud migration alternatives of this decision problem. The goal is finding the highest-ranked cloud migration alternative which splits the application system components into both the organisation's local premises and the cloud environment configurations. Inspired by AHP, the cloud migration concept focuses on the hierarchically structured cloud migration criteria (including the cloud migration criteria and their weights) to calculate the decision matrices[482]. These matrices directly depend on a set of cloud migration criteria within a level of the structured hierarchy of criteria, these criteria relative weights, and the evaluation of each cloud migration alternative per each of these cloud migration criteria. Each matrix refers to a level within the structured hierarchy of criteria and consists of of multiple alternatives and migration criteria. The proposed concept calculates each criterion for each leaf and branch node of the hierarchy. For each criterion without sub-criteria (or leaf node), its value calculated by computing either an objective or a subjective criterion. For branch nodes —that is, criteria with sub-criteria— its value is calculated according to the value of a lower level of the cloud migration criteria hierarchy. Section 6.3 describes the taxonomy of cloud migration criteria and some exemplary criteria.

The proposed concept relies on cloud migration criteria templates as pre-defined criteria sets and off-the-shelf pairwise comparisons between each criterion. Organisations customise or create their own templates tailored to their specific cloud migration project; or otherwise re-use criteria templates applicable to their cloud migration project. A cloud migration criteria template saves organisations some effort when they have similar priorities and requirements in how to cloud migrate their application systems.

Finally, the bottommost level of the AHP hierarchy contains the cloud migration alternatives, which capture how to adapt the application system and which components to re-scatter to the organisation's local premises and to cloud environments

[482]For more details about the decision matrices, DM, refer to cf. Juan-Verdejo et al. (2014b), pp. 467–474

configurations. All cloud migration alternatives are generated subject to respecting all applicable constraints.

Step 2

The prioritisation of the cloud migration criteria happens in *Step 2*, with a criteria set containing the pairwise comparisons of each cloud migration criterion to one another. The pairwise comparisons are then used to calculate the eigen vector with all criteria weights inputed in this step.

Step 3

Section 6.3 explains with more detail *Step 3* that is, the *alternatives evaluation* for all cloud migration criteria. The cloud migration alternatives are aggregated according to the eigen vector of weights computed in Step 2 from the eigen values of the matrix with all criteria weighed against one another.

Step 4

Next, *Step 4 validates that the consistency ratio* does not trespass the acceptance level of 10% that Saaty and Vargas proposed and thoroughly explained[483]. This step checks that the pairwise comparisons derive from consistent or near-consistent matrices by dividing the consistency index by the random index and hence assess the uncertainty of the decision made.

Step 5

Finally, *Step 5* entails ranking all cloud migration alternatives to recommend a particular cloud migration *decision*. The organisation in charge of the migration could still pick another one by correcting the cloud migration criteria and their weights, the application system components model and properties, the cloud environment model, or the formula that the proposed concept uses to evaluate an alternative for a particular criterion. The organisation can re-define the latter by either changing how an alternative is automatically evaluated or manually evaluating that alternative.

[483]cf. Saaty and Gonzalez Vargas (2001), pp. 23–39

6.3 Semi-automatic cloud migration decision support

The semi-automation of the cloud migration decision relieves organisations cloud migrating an application system of the burden of considering all cloud migration alternatives and the multi-dimensional repercussions of picking one over another. The proposed concept facilitates such an error-prone decision by automatically generating the cloud migration alternatives based on three input models describing: the target application system, the cloud migration criteria, and the cloud environments models.

Given the entire set of cloud migration alternatives, the proposed concept rejects the non-viable one according to the applicable constraints to then automatically weigh the alternatives for every cloud migration criterion for which there is a metric according to each criterion's relative weight with respect to each other criterion. For those subjective or objective criteria without a metric to assess them, the cloud migration concept includes the human in the loop by letting organisations weigh those cloud migration criteria. Finally, the ranked cloud migration alternatives are presented to the organisations so that they analyse the assessment and fine-tune it to pick their selected cloud migration alternative.

The following sub-sections show examples of constraints to describe the automatic alternatives generation subject to them. Further, examples of cloud migration criteria explain how the proposed concept automatically weighs the remaining cloud migration alternatives. However, the actual formalisation of these constraints and objective cloud migration criteria is published in a paper and not included here for simplicity[484].

6.3.1 Cloud migration alternatives generation subject to constraints

The proposed cloud migration concept generates architectural design alternatives to migrate an application system to cloud environments subject to the constraints

[484] cf. Juan-Verdejo et al. (2014b), pp. 467–474

defined by an organisation. Every cloud migration alternative specifies which components run within the local premises of the organisation and which components to migrate to which particular cloud computing environment configuration. A constraint limits the acceptance of cloud migration alternatives subject to a specific property of the application system, a cloud migration criteria, or a design restriction established by the organisation[485]. It represents a requirement vital for an application system regardless of whether it runs within the local premises or in a selected cloud environment. An organisation adds, extends, or removes constraints with the proposed concept.

A constraint set represent a fundamental part of the requirements of a cloud migration project. A constraint limits the acceptable cloud migration alternatives' properties. All the cloud migration alternatives for an application system have to either comply with all constraints or be disregarded to not to be considered by the cloud migration concept. The proposed concept decides when to reject an alternative to cloud migrate the application system based on the cloud migration criteria described by the organisation, the characteristics of the application system such as its architecture and properties, and the cloud environment configurations. Its output is a *Boolean* value to show whether a constraint is respected or not. As an example, by using the kind of formalisations described in the aforementioned paper, an organisation can define a constraint stating that a Java Virtual Machine (JVM) has to be runnable for a particular re-engineered version of application system deployed into a specific configuration of a cloud environment. Section 6.3 shows examples of how the proposed concept generates the cloud migration alternatives subject to all constraints.

The following specify constraints related to security and privacy as well as to the target application system.

A security and privacy constraint

One of the suggestions issued by the proposed concept to cope with sensitivity and privacy issues is keeping components within the computing premises local to the organisation cloud migrating an application system. That is, a cloud migration alternative with the local premises storing or processing data that organisations

[485] An equation mathematically describes a constraint in cf. Juan-Verdejo et al. (2014b), pp. 467–474

do not want to move out of their premises to a particular cloud environment configuration. This might be due to the lack of trust in that cloud provider or in the communications channel used for the cloud migration. In this case, the cloud migration concept does not cloud migrate the components with high sensitivity levels; that is, *High sensitivity* and *National Security High Sensitivity*[486]. The proposed concept returns a sensitivity constraint value equal to *False* for cloud migration alternatives that do not comply with the sensitivity-related constraints; and are therefore rejected. An organisation which trusts the cloud migration mechanisms and a cloud environment would relax this constraint.

An application system constraint

The cloud migration concept generates alternatives that comply with the application $AppSys$ requirements according to the components model and the components' properties specified by the organisation. Additionally, the concept lets organisations explicitly define constraints disallowing the migration of a component to a cloud environment configuration or cloud environment that does not supply a vital feature. For example, an explicit constraint does not allow the generation of an alternative that cloud migrates a component to a cloud environment configuration that does not support a required storage technology or proprietary tool, say *MongoDB*. This detail is captured in a component property of the application system whose value 1 represents a component that requires *MongoDB*; and where a component property of the cloud environment configuration whose value 1 means that the configuration supports *MongoDB*.

6.3.2 Cloud migration alternatives semi-automatic weighing

The cloud migration alternatives automatic weighing pertains to Step 3 in Section 6.2. Provided that the organisation has defined a particular metric for a cloud migration criterion or used a metric provided by the proposed concept, the proposed concept automatically calculates the score of all cloud migration alternatives for that criterion. Next, it aggregates the cloud migration alternatives based on the cloud migration criteria eigen vector —as Step 2, criteria prioritisation describes— to obtain the global ranking for all cloud migration alternatives.

[486] cf. Deshpande et al. (2018), pp. 194–201; cf. Rosado et al. (2012), p. 474

The proposed concept considers two kinds of cloud migration *criteria* as relevant for the organisation that wants to move application system to cloud environments: either objective or subjective cloud migration criteria. From a mathematical point of view, an *objective cloud migration criterion* refines the criterion so that it incorporates knowledge related to the available cloud environment configurations, the needs and characteristics of the application system, and the organisation's cloud migration criteria to move the application system to cloud environments. Each objective criterion represents a dimension related to the cloud migration —e.g. cost, performance, or sensitivity— and the cloud migration concept assesses it for every non-rejected cloud migration alternative. That is, a migration alternative to re-engineer application system by re-scattering its components to both the organisation's premises and the cloud environment.

Non-rejected migration alternatives comply to the constraints defined. An objective criterion depends on the function to re-scatter application system components, the application system components as well as their properties, the relations between the components and the properties of those relations, and the properties of the cloud environment configurations. Section 6.3 shows several examples of cloud migration criteria.

The *re-scatter* function describes the deployment function fundamental for the definition of the organisation's cloud migration criteria and constraints. Based on application system's components, the function returns the application system components re-scattered into both the local premises and the cloud environments. Examples of cloud environment configurations are a *small Amazon Elastic Compute Cloud (EC2) instance*, a *big Amazon EC2 instance*, or an *extra small Azure instance* provided by cloud providers such as *Amazon*, or *Azure*. A cloud service provider specifies the properties of its multiple configurations. An example of such a property is the amount of memory provided by a cloud environment configuration such as a *small Amazon EC2 instance*.

For a cloud migration criterion without any specified metric, the decision support model considers a subjective migration criterion, which gathers the feedback from the organisations regarding the target application system; hence, with human intervention. Organisations pairwise compare all cloud migration alternatives for each of these criteria. An example of a subjective cloud migration criterion is the trust an organisation has in a specific cloud environment configuration if no metric has been defined for it[487].

The proposed concept uses the criteria below[488] and the metrics per criterion to measure each cloud migration alternative for each criterion.

Accountability

Accountability is a subjective attribute or cloud migration criteria that affects trust and might be built based on several other cloud migration criteria such as: auditability, compliance, contracting experience, data ownership, ease of doing business, provider ethics, or service sustainability[489]. For the cloud migration problem, accountability includes the commitment of a cloud provider to comply with the existing policies as well as the legal, ethical, and moral obligations. The degree of accountability of a cloud provider depends on the mechanisms it puts into action to measure, prevent, and act to take responsibility for the stewardship of personal and confidential data entrusted to them[490]. The proposed cloud migration concept considers the cloud migration alternative accountability equal to the accountability of the cloud environment if the alternative migrates a component with a considerable sensitivity level. That is, all levels except *Low*[491]

Agility

Agility refers to the use of cloud environments to run an application system so that the organisation can quickly and cost-efficiently expand, change, and integrate new IT capabilities to accommodate business needs. It is reasonably argued that agility might depend on sub-criteria related to elasticity, portability, adaptability, and flexibility. For matters of clarity, the proposed concept calculates some criteria, such as the degree of portability of a cloud migration alternative to a cloud environment,

[487] Saaty and Vargas described how to calculate subjective criteria and evaluate them to a number: cf. Saaty and Gonzalez Vargas (2001), pp. 23–40

[488] The extensible hierarchy of criteria is inspired by SLAOrchestrator: cf. Ortiz et al. (2018), pp. 555–557; the Service Measurement Index framework: Siegel and Perdue (2012), pp. 411–415; and SMICloud Garg et al. (2011), p. 210

[489] Ibid.

[490] Ibid.

[491] Hence, *Moderate*, *High*, and *National Security High*; The proposed cloud migration concept formalises this criterion using an equation described in cf. Juan-Verdejo et al. (2014b), pp. 467–474

with relation to other criteria. For example, the degree to which the effort of porting the application system to the new environment is less than the cost of redevelopment or $1 - \frac{\textit{cost to port}}{\textit{cost to redevelop}}$[492]. The proposed concept relies on human intervention to determine, with pairwise comparisons, the cost to port and the cost of redevelopment. This is explained below with the cost cloud migration criterion.

Assurance

Assurance defines the probability of the application system to comply to the SLA and one could assume it depends on the services' availability, maintainability, recoverability, reliability, resiliency or fault tolerance, service stability, or serviceability[493]. These cloud migration sub-criteria related to assurance are highly affected by the properties of the local premises and of the cloud environment configurations to run the application system's components. For example, in the proposed concept the average *availability* —hence, service uptime divided by the total service time— depends on the availability properties for each software component, the SLA of the cloud environment configuration, and the local infrastructure. The proposed concept considers the infrastructure properties that represent the uptime and downtime in milliseconds of a component running either on the local infrastructure or a cloud environment in order to assess the availability of a cloud alternative. The concept considers that the total downtime time of a component which is down if the infrastructure that hosts it is down, given that a component will be down as often (or more) as the cloud environment.

Cost

It is reasonably assumed in this dissertation, that organisations prefer to effortlessly adapt the application system to cloud environments, the on-going cost of running the application system, and the one-time migration costs[494]. The cost criteria groups them all according to its weight relative to each other. As an example of a cost criterion, refer to the formulation of the one-time migration cost criterion[495]. This formulation uses a sum that runs over the different component sizes, which the cloud migration concept defines from *Very Small* to *Very Large*.

[492]cf. Gonidis (2015), pp. 132–139; Mooney (1997), p. 2
[493]cf. Ortiz et al. (2018), pp. 555–557; Siegel and Perdue (2012), pp. 411–415
[494]cf. Gholami et al. (2017), p. 107; Armbrust et al. (2010), p. 50
[495]cf. Juan-Verdejo et al. (2014b), pp. 467–474

The proposed concept can measure the effort to adapt an application system to cloud environments by taking into account the cost of porting or redeveloping the application system. The proposed concept incorporates organisations into the cloud migration decision-making process by letting them pairwise compare cloud migration alternatives to calculate these cloud migration criteria. There is room for improvement in calculating the effort criterion without human intervention.

Security and privacy

Arguably, most organisations migrating their application systems to cloud environments consider security and privacy as major hindrances to the cloud migration because they fear losing control of parts of the application systems and data they host outside their premises[496]. The proposed concept defines a constraint that forces leaving *High* and *National Security High* Sensitivity components at the local computing premises of the organisation instead of migrating them to any cloud environment[497]. This approach considers acceptable, in terms of security and privacy, moving *Low Sensitivity* data out of an organisation's premises due to their lower sensitivity level[498]. This decision leaves the proposed concept with judging whether to migrate data and components marked with *Moderate Sensitivity* to a cloud environment configuration. The fewer elements with *Moderate Sensitivity* a cloud migration alternative migrates to a cloud environment, the higher the cloud migration concept ranks that particular alternative.

Usability

The ease of using a component depends on its accessibility, client requirements to use it, which might include sub-criteria such as installability, learnability, operability, suitability, transparency, and understandability[499]. These cloud migration sub-criteria to usability are rather imprecise and subjective. Therefore, the proposed concept allows organisations to subjectively estimate usability based on the application system's components model and some clodu environment configurations properties related to the ease of use.

[496]cf. Deshpande et al. (2018), pp. 194–201; Rosado et al. (2012), pp. 482–485
[497]As Section 6.3 shows
[498]cf. Deshpande et al. (2018), pp. 194–201; Rosado et al. (2012), p. 469
[499]cf. Ortiz et al. (2018), pp. 555–557; Siegel and Perdue (2012), pp. 411–415

6.4 Architecture of the cloud migration DS concept

The architecture of the proposed concept puts together the different software components (modules) and repositories shown in Figure 6.2. The cloud environments modelling module allows for maintaining the repository with the cloud environment configuration models consistent and synchronised with the existing cloud offerings. The cloud migration criteria and the on-premises application system modules model the priorities of an organisation to cloud migrate its application system. Finally, the cloud migration alternatives module generates the cloud migration alternatives based on the IaaS, PaaS, and SaaS cloud environments configurations and the organisation's requirements. The cloud migration alternatives module uses its cloud migration metrics sub-module to assess the cloud migration criteria per cloud migration alternative.

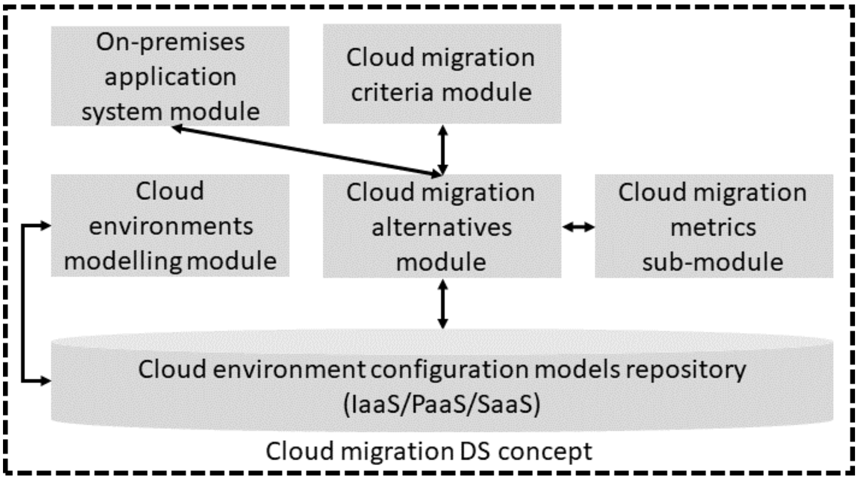

Figure 6.2: Bird's-eye view of the architecture of the cloud migration DS concept[500]

Cloud consumers find available cloud offerings as the proposed cloud migration concept implements core functionalities such as cloud offering searching, modelling, and proposition. It ranks the different cloud migration alternatives to adapt the target application system to the existing cloud environments after discarding unsuitable alternatives according to the specified constraints. The proposed concept ranks every cloud migration alternative based on the cloud portability criteria and the cloud migration criteria that depend on the cloud offerings' service level[501]. Finally, it proposes the highest-ranked cloud migration alternative.

[500] Own illustration.
[501] IaaS-, PaaS-, and SaaS-specific cloud migration criteria: see Section 6.6

The proposed concept supports the previously described process by automatically measuring and quantifying to a numerical value the cloud migration criteria. For that, it uses the objective cloud migration metrics sub-module. Even though the cloud migration concept proposes metrics to quantify subjective criteria. Sometimes it will not be feasible to quantify all criteria and organisations will give appropriate weights with the assistance of the proposed concept. Therefore, the proposed concept allows for information gathering from organisations to calculate these subjective criteria. The concept involves the organisation in pairwise comparing the cloud migration alternatives. In order to easily consider trade-offs and structured relationships between objective and subjective criteria, quantify them, and aggregate them; the cloud migration decision sub-module (part of the cloud migration alternatives module) uses AHP to facilitate the multi-criteria decision making.

In usual simple multi-criteria decision-making problems, all criteria are expressed in the same unit but this does not apply to this more complex process that uses the AHP to weigh criteria expressed in different criteria specific to the IaaS, PaaS, and SaaS service models. AHP helps with such challenges like, for example, weighing the cost of a cloud migration alternative —expressed in euros— against the respect to national regulations —expressed in Boolean rationales.

This module holds information on the capabilities and requirements of the cloud services in the usual three cloud service models. In addition, the cloud environments modeling module allows organizations' and small and medium-sized enterprises' software engineers to rate cloud offerings based on their experience while considering the services and restrictions of each cloud offering. The cloud environments configuration models repository includes the Unified Cloud API, which can be used to interact with different cloud offerings. Each software engineer can use the module to write comments about the cloud provider and its cloud environment configuration.

Moreover, users can rate services and express their satisfaction or dissatisfaction with the cloud environment in terms of its quality and reliability as well as its user-friendliness and usability. The cloud migration metrics sub-module, which is outside the scope of this dissertation, paves the way to allow for monitoring the selected cloud environment to help organizations manage the entire life cycle of the application systems deployed to it.

This section presents the architecture of the proposed concept that aims at facilitating organisations' decision process of cloud migrating their application systems. The architecture includes the cloud environment configurations models repository that can be organised, like in the first real-life evaluation scenario described by Chapter 7, in a selection of PaaS cloud offerings[502]. The following section provides an overview of the different dimensions driving the selection of Infrastructure-, Platform-, or Software-as-a-Service cloud offerings based on a number of cloud migration criteria. The proposed concept uses those dimensions to build the hierarchy of cloud migration criteria and tailor them to the treated cloud migration project and to the organisation that is assisted when weighing the cloud migration alternatives.

6.5 Workflow followed by the proposed cloud migration DS concept

The proposed concept operates according to the two simplified workflows below. These orchestrated and repeatable activities result after systematically organising resources into processes to provide services and process information. Figures 6.3 and 6.4 depict the sequence of operations and multiple simple and complex mechanisms that a person or group performs to cloud migrate the application systems of an organisation.

Aiming at simplicity, the two workflows guiding the cloud migration concept present a mid-level perspective that represents the different interactions of three stakeholders with ten operations grouped in four complex operations —that is, operations 1.1, 1.2, 2, and 3. The three stakeholders —on the left-hand side of Figures 6.3 and 6.4— include the organisation cloud migrating its application systems, the proposed concept and the cloud environments modeller. The cloud migration concept implements and semi-automates the described workflows. The four complex operations in these two figures include modelling operations —to let modellers specify

[502] Sections 6.11, 6.12, and 6.13 explain how to tailor the cloud migration concept to the evaluation scenario described in Section 7.2 of Chapter 7

the on-premises application system, the cloud migration criteria, and the cloud environment configurations—, re-engineering operations, and decision-making operations.

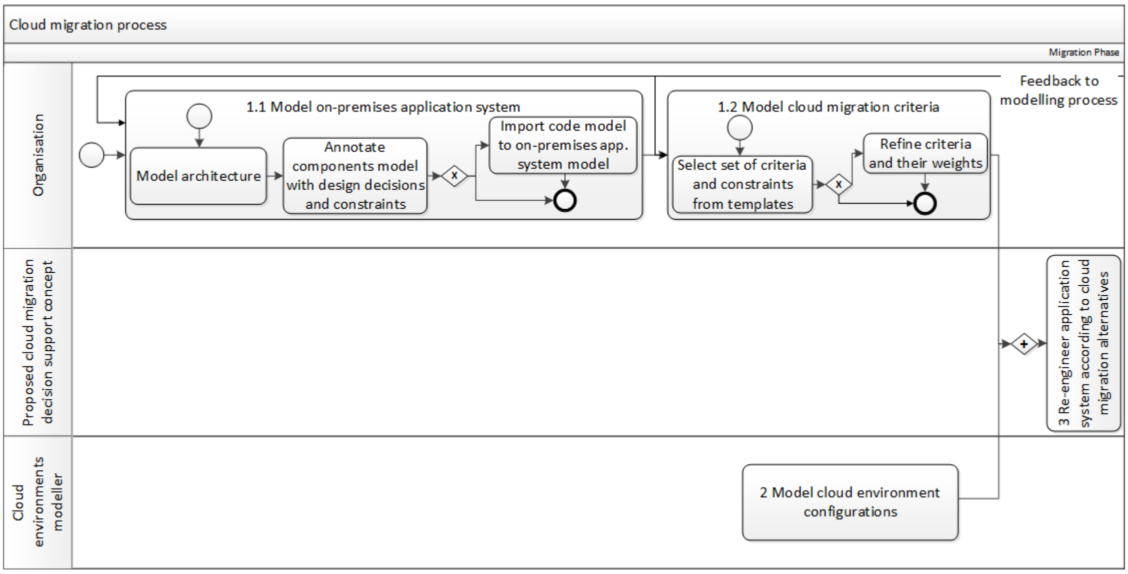

Figure 6.3: Workflow followed by the proposed cloud migration DS concept —part 1[503]

Figure 6.3 shows the first part of the workflow that the proposed concept follows, starting off with 1.1 Model on-premises application system. During this step, the organisation models the architecture of the target application system and annotates the software components model with design decisions and constraints. Next, an optional step lets the organisation import code model (if the actual source code is available) to add modelling details to the on-premises application system model that specifies the architecture and components design and constraints. This added-level or detail allows for performance or quantitative analysis of the on-premises application system and the cloud migrated application system. However, this is an optional step as the proposed works as supporting the decision-making of cloud migrating an application system and therefore works without this additional level of detail.

Once a member of the organisation has modelled the existing application system, 1.2 allows an organisation to describe the hierarchy of cloud migration criteria driving the cloud migration decision. Firstly, the cloud migration concept lets the organisation choose from a set of templates with cloud migration criteria to

[503]Own illustration

facilitate and speed up the task in a less error-prone manner. At the same time the organisation can fine-tune the cloud migration criteria and (if needed) specify metrics for them. Figure 6.3 represents this, at the top right-hand corner, with the step to Refine criteria and their weights to customise the cloud migration criteria to the decision-making process that the organisation follows.

The bottom swimlane represents the stakeholder cloud environments modeller and, in a very simplistic view, the step to 2 Model cloud environments with variability points in feature models. Figure 6.4 explains the last step in Figure 6.3 3 Re-engineer application system according to cloud migration alternatives.

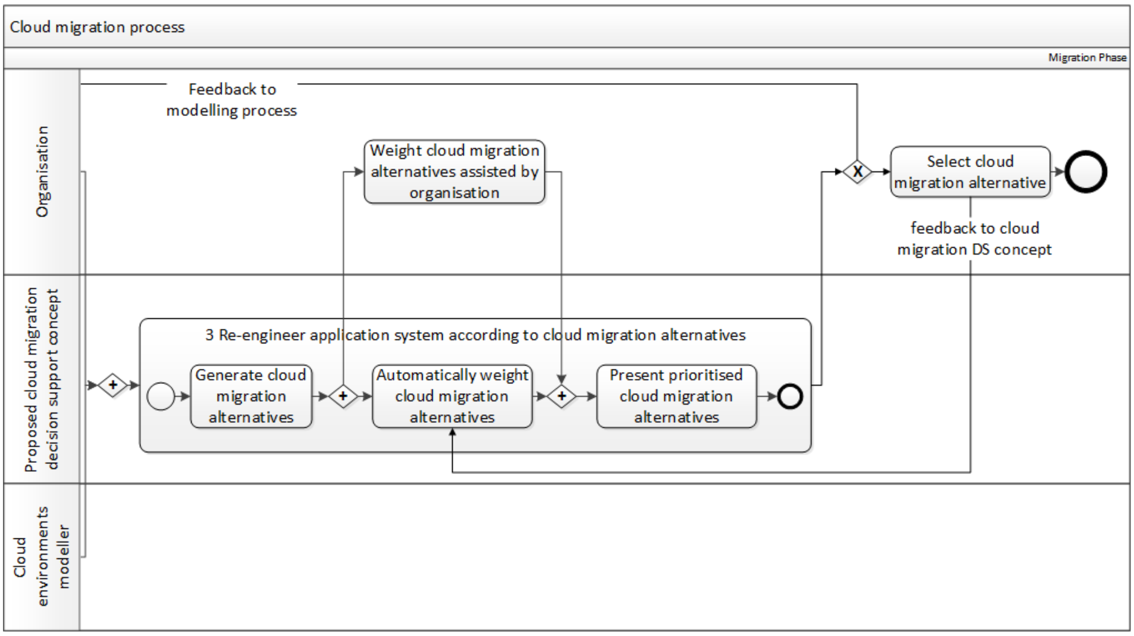

Figure 6.4: Workflow followed by the proposed cloud migration DS concept —part 2[504]

Figure 6.3 describes the last steps that the proposed concept follows until reaching the selection of the cloud migration alternative that the organisation follows to cloud-enable the target application system. The re-engineering of the target application system starts with generating the cloud migration alternatives with application system components running in both the local and the cloud premises. The cloud migration concept weighs each of these cloud migration alternatives according to the cloud migration criteria in a semi-automatic manner. Then, it automatically assesses the cloud migration criteria for which there is a particular metric and lets

[504] Own illustration

the organisation weigh in and assess the cloud migration alternatives for subjective cloud migration criteria or for objective criteria without an associated metric.

Finally, organisations can fine-tune the decision making by selecting the final cloud migration alternative of their preference. This decision goes back as a feedback loop to the automatic weighing mechanisms of the proposed cloud migration concept and to the modelling processes for: the on-premises application system and the cloud migration criteria.

6.6 IaaS-, PaaS-, and SaaS-specific cloud migration criteria

Cloud computing providers offer IaaS, PaaS, SaaS services. Depending on the level at which cloud providers offer their services, different service model-dependent cloud migration criteria gain importance and drive how organisations define the entire hierarchy of cloud migration criteria and relative importance; and in turn the cloud migration decision. The proposed cloud migration concept takes into account the cloud migration criteria at the level of the cloud stack at which the cloud migration and cloud environments selection happens to weigh each one of them for each cloud migration alternative.

Similar contents to those summarised in this section already appear in a paper published in collaboration with Bholanath Surajbali. However, the cloud migration criteria for the different deployment models are a contribution of this dissertation[505].

6.6.1 IaaS-specific cloud migration criteria

IaaS offerings reside at the bottom of the cloud stack with the rest of cloud service models building on top of it. At the IaaS level, cloud providers make their resources available, both physical and virtual. These resources include virtual machines and servers as well as storage, networking resources, and load balancers. Given the particularities of the Infrastructure-as-a-Service level, the literature study and requirements that; Figure 6.5 shows six cloud migration criteria affecting cloud migration at the IaaS level.

[505] cf. Surajbali and Juan-Verdejo (2014), pp. 275–282

IaaS service characteristics. The price, memory available to the user, number of virtual cores, and input-output bandwidth vary among each VM instance type. These criteria also include information about network access and configuration options for services. Consumers can also deploy their systems according to the different cloud deployment models (public, private, hybrid, community) or a partially migrated deployment.

IaaS portability standardisation. The cost can potentially be lowered when porting software from one IaaS environment to another one. The respect to standards and the adoption to portability methodologies aims at avoiding vendor lock-in problems and it can make porting services easier. At the IaaS level, consumers usually need to port software runtime environments configurations and APIs and they typically do it at the level of VMs. The Open Virtualisation Format[506] is a format for open packaging of virtual appliances on a VM and it describes how to manage, deploy, distribute, and run them.

Hardware infrastructure. At this level, cloud consumers can specify their processing needs as well as the memory they need to allocate to their application systems. IaaS environments offer different servers with varying processing power or CPUs as well as memory or capacity of instances.

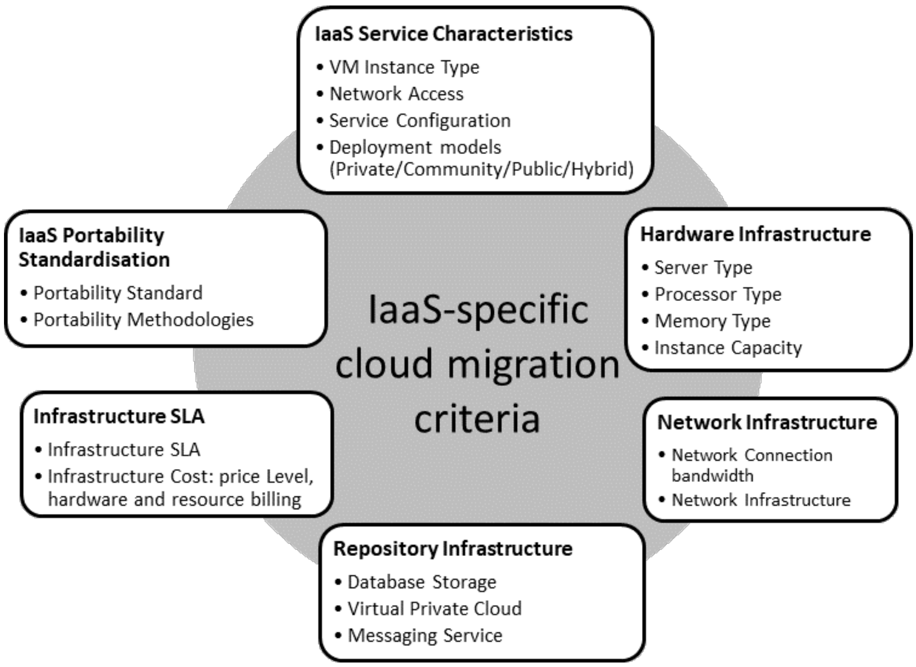

Figure 6.5: Infrastructure-as-a-Service offerings cloud migration criteria

[506]OVF is proposed by the Distributed Management Task Force (DMTF)

Infrastructure SLA. The Service-Level Agreement is a contract specifying performance metrics for the agreed services and the objectives on the level of service to be provided. It includes details on the agreed service availability and establishes the levels of reliability, elasticity capacities, and security provision. The SLA sets the minimal resource provisioning and the geographical location of the deployed services as well as the disaster management metrics and the service and auditing mechanisms. In addition, the agreement specifies the energy consumption and the cost, price level of service provision, and the metrics to bill service consumption.

Network infrastructure. IaaS cloud environments offer shared network resources and vary in terms of available bandwidth and networking infrastructure.

Repository infrastructure. Various messaging implementations, storage and database capabilities, and potentially virtual private cloud functionalities. Messaging services tend to get increasingly commoditised. For example, the Amazon Simple Queue Service (SQS) offers out-of-the-box messaging implementations including the Microsoft Message Queuing or the Java Message Service (JMS).

6.6.2 PaaS-specific cloud migration criteria

Six cloud migration criteria differentiate PaaS offerings from one another as Figure 6.6 depicts. The proposed concept takes these six high-level cloud migration criteria into account to help organisations in migrating their systems to cloud offerings at the Platform-as-a-Service level by weighing each cloud migration criterion and cloud migration alternative

Platform application. Essential characteristics related to the application system under study which consists of four sub-criteria. 1) Framework: the underlying framework used to build the application system, which can be a language framework such as Java, the .Net Framework, Spring; or a computing framework such that Java Enterprise Edition EE, JBoss. 2) Extensibility: built-in extensibility with mechanisms for application system monitoring, logging, and message queuing technologies. 3) Services: native or add-on services. A native service is typically part of the PaaS application system —such as the data storage service—; while

[506] Own illustration based on Surajbali and Juan-Verdejo (2014), p. 279
[507] Own illustration based on Surajbali and Juan-Verdejo (2014), p. 280

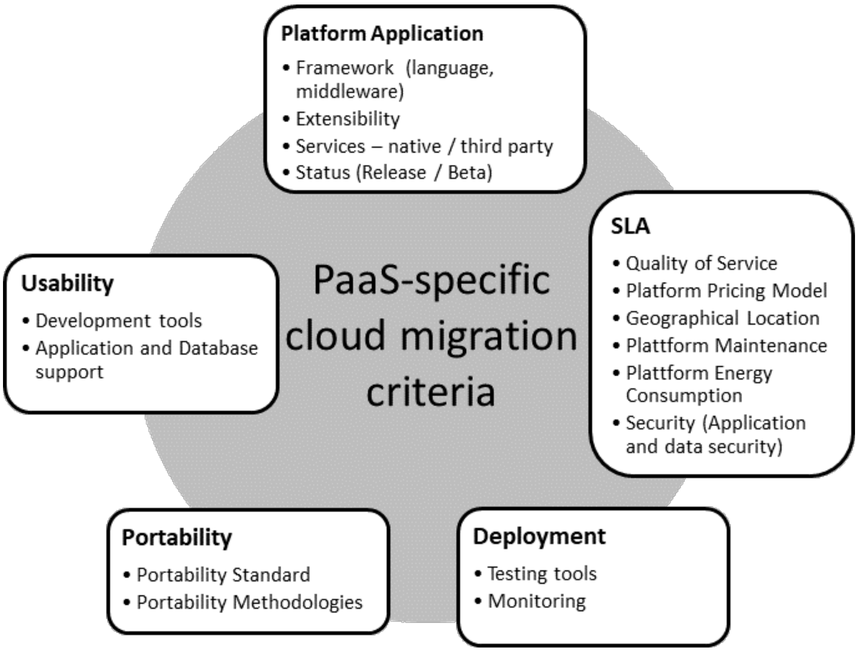

Figure 6.6: Platform-as-a-Service offerings cloud migration criteria[507]

an add-on service is provided by a third-party to be integrated with the application system —for example, a search engine. 4) Status: to describe the deployed application system in terms of beta or final release version.

Usability. The usability of a PaaS offering relates to the knowledge consumers need to maintain their application systems deployed to it and to extend its functionality. To that respect, providers usually offer development tools for rapid development and different support and training levels. In addition, their intrinsic usability might be different while offering differing quality of documentation.

SLA. SLAs specify the PaaS offering service contracts in terms of the Quality of Service delivered through the use of a particular PaaS offering. They have different properties regarding their pricing models, geographical coverage, and availability. Quality of Service: QoS specifies the response and resolution times the application system service requests, providing the minimal, acceptable and minimal levels of QoS supplied by the PaaS provider. Different pricing models have providers charge their customers depending on the use of resources. Billing can for example be either fixed or metered as well as follow a hybrid mechanism[508]. The geographical coverage can cover multi-geographical deployments offered to organisations. For an organisation subscribing to a PaaS provider, the geographical location presents

[508] See Chapter 2

legal concerns. For instance, companies operating in the EU single market shall abide by the EU GDPR regulation in how they treat personal data. Availability: PaaS providers commit to a certain level of availability for their PaaS offering.

Portability. Depending on the PaaS selection, cloud consumers can have access to infrastructures that comply to particular standards aiming at enhancing the portability of their application system components for both computing and data: OCCI, TOSCA, CIMI, and CAMP[509]. Which, in turn help avoiding vendor lock-in.

Deployment. Tools offered by PaaS providers help validate the deployment design of data and application software components as well as to test the deployment. These can be tools using model-driven approaches to check requirements[510], or software tools to help developers across the entire life-cycle of software development. Additionally, PaaS providers offer a number of tools to monitor application system and data components and its performance based on usage to report back to organisations to ensure proper QoS.

Security. The security level of a PaaS offering is crucial for organisations to select a PaaS environment configuration and to let them assess the overall application system security levels achieved. Organisations are interested in porting the data layer of many of their application systems —such as BI application systems— to PaaS. In those cases, the trustworthiness of providers is key as they store data on their behalf and could pose a security threat. The process to cloud migrate an application system must ensure data accessibility and integrity. Some PaaS users might choose to distribute data across different geographical locations. Likewise, they might consider different deployment models to partially migrate their application systems or use private PaaS. Private PaaS deployments can ensure very comprehensive data ownership to enforce security upon the data entrusted to them. Organisation sometimes trust public PaaS offerings or use partial or hybrid PaaS models. They have the option of adding computing resources to accommodate the demand of their customers and keep sensitive data in their premises in a partial migration or they migrate their data to a private cloud.

[509] See Section in Chapter 4
[510] cf. Juan-Verdejo et al. (2014b), p. 467; Fleurey et al. (2004), p. 29

6.6.3 SaaS-specific cloud migration criteria

As a result of the level of freedom provided by the most abstract cloud computing model, SaaS computing environment, the decision space at this level is a bit smaller. Data migration is higher in the priority list given that SaaS solutions work with those data via their offered services for e-mail, virtual desktop, Customer Relationship Management (CRM), enterprise resource planning (ERP), games, management information systems (MAN), Computer-aided design (CAD) software, or communication. Figure 6.7 shows three SaaS-specific cloud criteria.

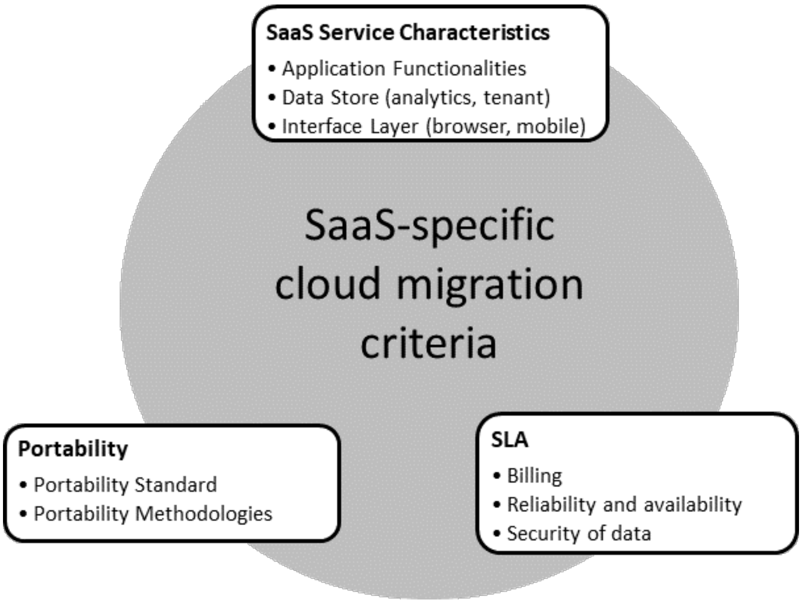

Figure 6.7: SaaS-specific cloud migration criteria[511]

SaaS service characteristics. These are characteristics related to functionalities of the application systems offered as a service, the data store for analytics and tenant-dependent, and the interface layer depending on the client device and browser.

Portability. SaaS portability and standardised methodologies can potentially alleviate costs when moving to a different SaaS provider while at the same time keeping business stability and avoiding SLA violations. With data portability, different application systems can share data components.

SLA. SaaS services specify a contractual agreement on the levels of reliability and availability offered in addition to determining how the consumer will be billed and how their data will be secured.

[511] Own illustration based on Surajbali and Juan-Verdejo (2014), p. 281

6.7 Model-driven PaaS migration DS concept

The migration of software systems to PaaS offerings represent a particularly interesting scenario given the fact that research gaps existed around the topics of interoperability, portability, and standards at the time of writing this dissertation[512]. This section explains the inner workings of how the proposed concept facilitates migrating application systems to PaaS offerings according to the six high-level PaaS-specific cloud migration criteria explained in Section 6.6. The model-driven PaaS migration DS concept is a part of the cloud migration concept and is evaluated by applying it to the PaaS offerings scenario[513] after tailoring the proposed concept to its application to this scenario —as Section 6.11 documents.

Organisations such as SMEs might achieve key competitive advantages by migrating their application systems to PaaS to increase their business agility and their velocity to market products. PaaS offerings hide some implementation details from the developer and make it easier to scale up and down the use of resources and to reduce costs. Underlying implementation aspects such as low-level details about virtual machines, servers, and load balancers as well as networking resources and data storage and access capabilities. This shortens the time to market PaaS solutions. PaaS environments offer additional simplicity because they run on top of IaaS offerings that take over some of the configuration, load balancing, and caching. Developers can use built-in or off-the-shelf services, tools, and functionalities to write code faster and with less errors by using software versioning and resource provision at the platform level. PaaS consumers get access to different libraries, APIs, and development tools to access storage and to deploy and execute their application systems.

Faster application system testing and deployment to PaaS environments entail that development teams easily try different configurations of their application systems and improve their quality as they test them, assess their performance, and identify potential compatibility issues. Organisations that deploy their application systems

[512] Part of the contribution of this section was the dissemination of the results by means of showing the poster at the ACM/IEEE 17th International Conference on Model-Driven Engineering Languages and Systems, MODELS 2014. This section is very related to the authored paper —*Model-Driven Engineering meets the Platform-as-a-Service Model*— and the poster to which the ACM SRC Poster Jury awarded the third prize of the competition.

[513] See Section 7.2 in Chapter 7 to read about the PaaS offerings evaluation scenario

to PaaS offerings do not have to configure web servers in multiple servers and handle the load balancing and replication between master-master and master-slave databases. Organisations have to manage redundancy, failovers, and heartbeats when running their application systems in IaaS offerings. PaaS providers overtake the management of the underlying operating system as well as the middleware and runtime layers. The migration of application systems to PaaS offerings might entail adapting them so that they allow for multi-instance application systems. Still, PaaS offerings let organisations simply add new services and use those services already developed for that specific platform. This relates to one fundamental challenge of working at the PaaS level, at the time of writing this dissertation: the vendor lock-in problem.

6.7.1 Bird's eye view of the model-driven PaaS migration DS concept

There are multiple PaaS providers to chose from as well as multiple configurations within them. The supplied execution environments, development tools, web servers, databases, or development tools vary across PaaS providers and the configurations they offer. As organisations usually have difficulties to select the PaaS offering that works best for them and accordingly often adapt their application systems to the selected PaaS configurations in a manual and hence error-prone manner. Model-driven approaches can of course facilitate this adaptation process.

The proposed cloud migration concept uses model-driven engineering to cope with the multiple architectural decisions to take and its repercussions across the entire software system. Meta models allow for describing the domain knowledge driving the cloud migration decision to PaaS offerings including the architecture of the target on-premises application system as well as the components properties and the description of the potential target PaaS offerings. In addition, the meta models describe the re-scattered components to the premises of the organisation and to the PaaS offerings, which result in alternatives to cloud-migrate application systems. These meta models are used in the proposed concept to support the design of the cloud migration deployment of the application system to PaaS. They

[514] Own illustration based on Juan-Verdejo et al. (2014b), p. 470

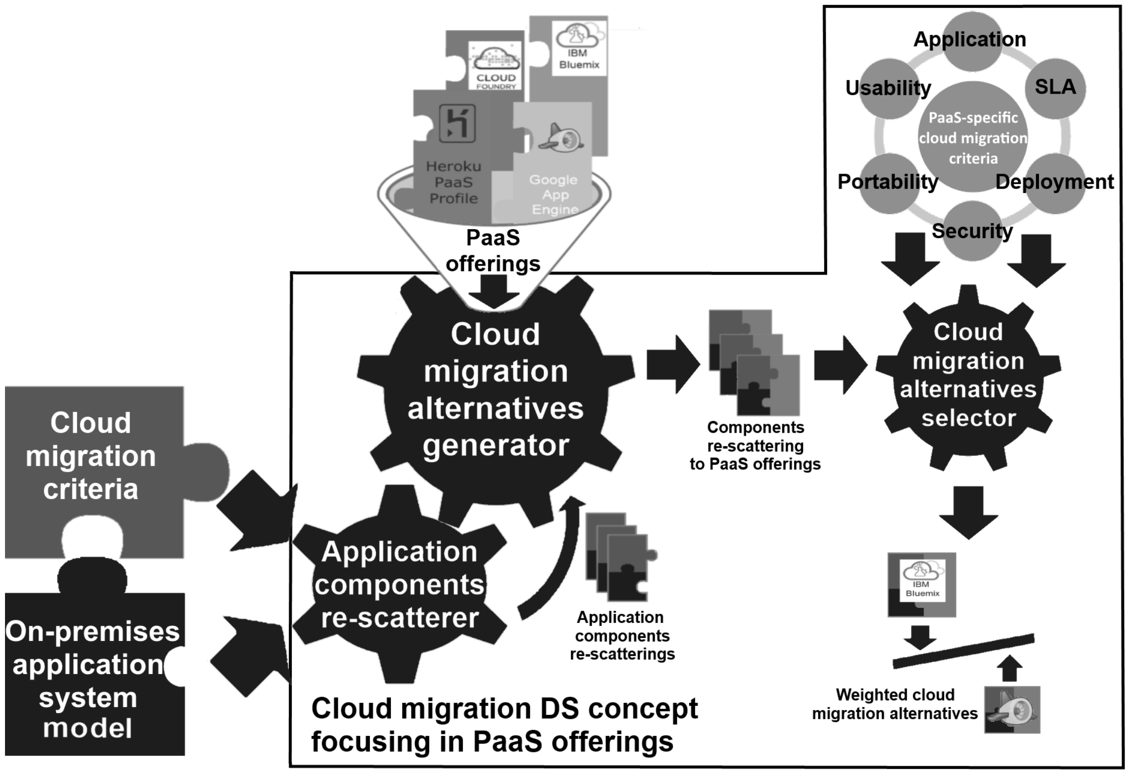

Figure 6.8: Bird's eye view of the model-driven PaaS cloud migration DS concept[514]

help select the alternative that best fulfils the organisation's cloud migration criteria but considers which components to keep locally to respect their constraints with respect to data privacy and security as well as governance, compatibility, and performance.

The migration to PaaS environments follows the high-level description of the cloud migration inside the black box of Figure 6.8 according to the three input models (the three puzzle pieces outside that box). On the left-hand side and top left-hand side the three input modes of the cloud migration concept include the model of the PaaS offerings —such as Heroku, IBM Bluemix, Google Cloud, or Cloud Foundry—, the cloud migration criteria, and the target on-premises application system model. The cloud migration concept assists the organisation migrating the application system to PaaS offerings guiding it to specify weights for each of the six PaaS-specific cloud migration criteria, at the top right-hand corner of Figure 6.8, according to their importance for the organisation. The meta model in Figure 6.11 allow for models describing the architecture of the application system in addition to its requirements and constraints. The third input model describe the PaaS environments configurations according to the meta model in Figure 6.10. Figure 6.8 describes how the cloud migration concept uses the transformation of the models to the application system components re-scattered to PaaS offerings and the local premises. Further, the proposed concept weighs —with the assistance of the organisation when needed— the different cloud migration alternatives according to AHP based on the six PaaS dimensions at the top right-hand corner as Section 6.6 explains.

Organisations prioritise different cloud migration criteria such as designing a PaaS-based application system which can be more easily extended or a system that is easier to be deployed. In addition, some other organisations might intend to improve the application system's security and portability. PaaS providers contribute to the variability of the cloud-based design with various tools to migrate or port application systems to PaaS as well as various cost models, followed standards, and security protocols. Likewise, the QoS and trust guarantees represent alternative criteria that organisations take into account. When considering PaaS migration options, organisations and SMEs may prioritize finding solutions that are reliable, performant, well-regarded, and cost-effective. They may also prioritise solutions that meet their technical requirements with minimal adaptation or development effort. However, it can be challenging to compare and evaluate different PaaS offerings based on these criteria, particularly in the long term, as there are few standards and metrics to guide the process. The model-driven PaaS migration DS concept assists organisations to consider all these different aspects and implications of the cloud migration that otherwise organisations find difficult to consider all together to make optimal decisions.

6.7.2 AHP for PaaS cloud migration DS concept

The cloud migration concept uses the six PaaS-specific cloud migration criteria to compare PaaS offerings, as shown at the top right-hand corner of Figure 6.8. Organisations may need assistance in incorporating all the relevant knowledge into the decision-making process when it comes to migrating application systems to PaaS. That is, they might need to consider conflicting dimensions or sub-dimensions as they encounter vendor lock-in issues when moving their application systems between PaaS providers. Some providers may lack adherence to standards, which in turn might hinder the portability of applications and data. Even when portability is technically feasible, the complexity of the process and the accompanying switching costs can deter organizations from attempting the migration project[515]. Organisations aim at switching between PaaS providers with ease and transparency for any reasons they consider such as changes in service level agreements or rising costs. But they also want to minimise risks like data loss or an inability when deciding to run their application systems on a different platform.

The box in Figure 6.8 presents three of the mechanisms used in the proposed concept that focus on migration to PaaS offerings. These mechanisms include the application system components re-scatterer and the cloud migration alternatives generator. Both of which take into account the model of the on-premises application system as well as the cloud migration criteria to design how to re-scatter the application system and its components to the premises of the organisation and to PaaS offerings. The cloud migration concept might keep components locally due to privacy and data protection requirements concerning sensitive data whose data owner or controller does not permit to leave the premises local the organisation. Finally, the cloud migration alternatives selector assess each cloud migration alternative by weighing them in relation to one another with regards to the design of how to migrate the application system. That is, components re-scattering to PaaS offerings in Figure 6.8. Subsequently, the cloud migration concept selects the best cloud PaaS migration alternative: weighed cloud migration alternatives. The proposed concept is a decision support concept that uses the Analytic Hierarchy Process (AHP) as Figure 6.9 exemplifies with six PaaS-specific cloud migration criteria. Here, the proposed concept weighs each cloud migration criterion against each other to compare the cloud migration alternatives —hence, the cloud migration alternatives at the bottom of Figure 6.8— that entail migrating to either the Google Cloud, Heroku, or Cloud Foundry following the local and global rankings done with AHP. Section 4.2 and Figure 4.4 explains this in Chapter 4.

[515]cf. Gonidis et al. (2013), p. 275

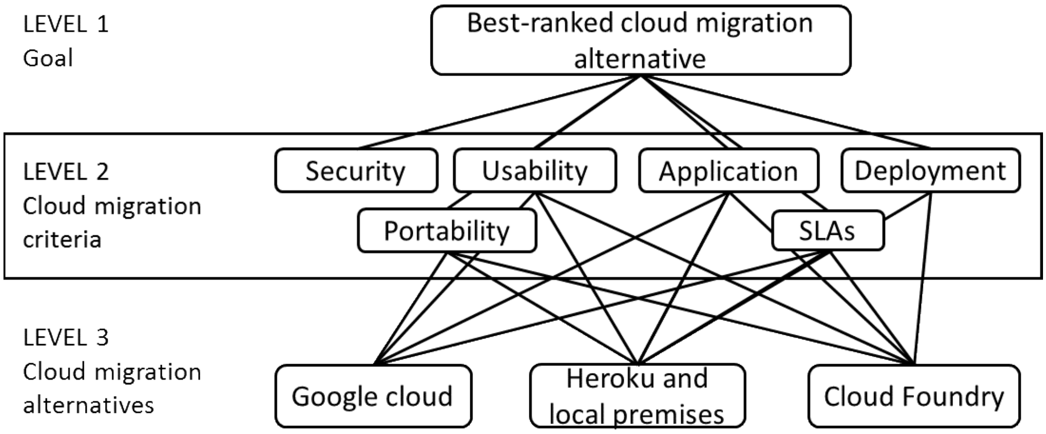

Figure 6.9: Cloud migration criteria and alternatives represented in the AHP as used in the model-driven PaaS migration DS concept[516]

6.7.3 Meta models for PaaS migration

Figures 6.10, 6.12, and 6.11 show the meta models that the proposed concept uses to assist organisations in the cloud migration decision-making process. These meta models incorporate the six PaaS-specific cloud migration criteria explained in Section 6.6.

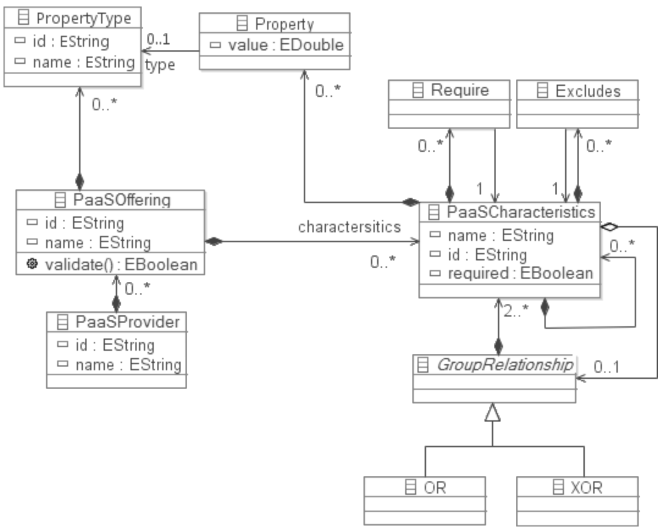

Figure 6.10: Meta model to which the PaaS offerings conform[517]

In particular, the cloud migration concept allows for profiling the on-premises application system via meta models while the PaaS criteria description and PaaS offerings configuration follow other two meta models. Previously to the migration,

[516]Own illustration following the Analytic Hierarchy Process (AHP) that Figure 4.4 highlights in Section 4.2 of Chapter 4
[517]Own illustration based on Juan-Verdejo (2014), p. 5

the proposed concept supports the selection of PaaS environments according to the six migration criteria described above. These criteria include particularities of the application system to be migrated, the targeted deployment configuration, the aimed service-level agreement (SLA), and some dimensions related to the usability of the environment and its facilities to increase security and portability. The cloud migration concept targets higher levels of portability to offer better levels of service and to keep vendor lock-in to a minimum. It facilitates selecting a PaaS environment and uses the three above meta models to help decide how to modify the application system to make the best out of the PaaS configuration selection. That is, to increase the scores per PaaS migration criteria.

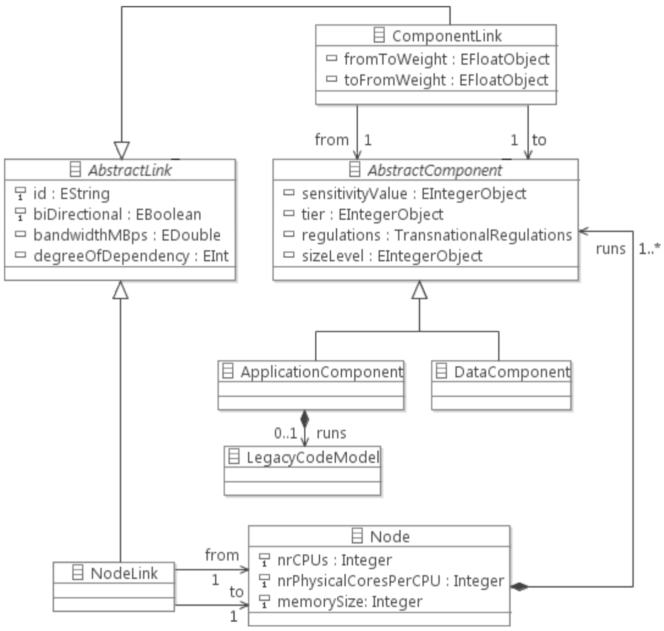

Figure 6.11: Meta model describing on-premises application system models[518]

Figure 6.10 show the meta model that describes a PaaS offering and that can be used to validate these offerings. As it describes, a PaaSProvider can bring about several PaaSOfferings with specific PaaSCharacteristics. PaaSCharacteristics can group other PaaSCharacteristics in addition to including or excluding other characteristics. The grouping works with a GroupRelationship as an OR relationship when several PaaSCharacteristics can be present at the same time or XOR relationships if only one of them can exist. The PaaSCharacteristics of a PaaSOffering offer a particular value of each Property of a PropertyType. In addition, the meta model shown in Figure 6.11 allow organisations to define the

[518] Own illustration based on Juan-Verdejo (2014), p. 5

architecture of their application system as well as its requirements and constraints. Organisations can model their application system's data and components, specify in which nodes these components run, and how these components are connected.

If organisations use the two input models based on both meta models in Figure 6.10 and Figure 6.11, the proposed concept can generate the cloud migration alternatives to move the target application system to PaaS offerings. These alternatives consist of a model that complies to the meta model in Figure 6.12. The meta model describes how to re-distribute the software components composing the to-migrate application system to PaaS offerings and the local premises if needed. In addition, the model describes an architecture with some components (*AbstractComponents*) migrated to a PaaSProvider and some others kept locally at the LocalPremises. This meta model also enable the measurement of PaaSOffering scores per DimensionType. In this case, a DimensionType is any of the six mentioned criteria influencing the PaaS migration.

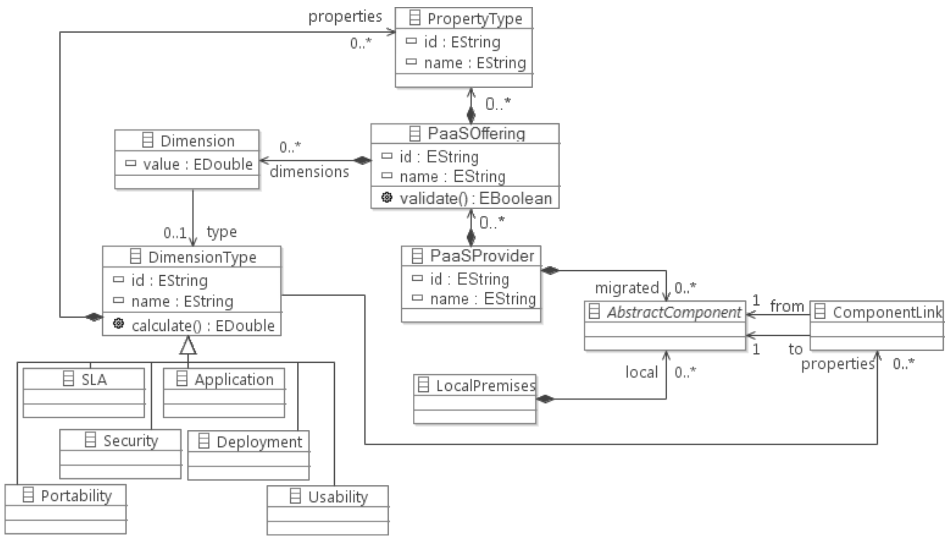

Figure 6.12: Meta model of the application system re-scattered to PaaS offerings[519]

6.7.4 Discussion on model-driven PaaS migration

The uniqueness of the proposed concept with a clear focus on the migration to PaaS offerings strives for re-scattering the application system components to the local premises and the different PaaS offerings to improve the assessment

[519]Own illustration based on Juan-Verdejo (2014), p. 5

of the organisation's PaaS-specific cloud migration criteria. All the proposed PaaS cloud migration alternatives respect the constraints of the target application system including those related to the allowed delays in component-to-component communication or some other specifications that depend on the selected platform. An example of these specifications might entail supplying a specific database management system (for example MySQL or MongoDB) or a particular application system server, such as Tomcat. Finally, the alternatives to PaaS migration need to respect the policies of the organisation with regards to data protection according to the sensitivity of their data and might need to keep some software components storing sensitive data at the local premises.

6.8 Cloud migration DS concept for soft reservations

A soft reservation is a mechanism for cloud environment providers to offer flexible reservation of their resources that benefits both cloud consumers and providers. The proposed concept is used to describe soft reservations by using the cloud environments modelling module[520]. This is an example of an advanced mechanism that the cloud migration concept could weigh to improve the decision making with regard to the effects of technologies that cloud environments might support in the future. Likewise, Section 6.9 extends the concept to consider cloudlets. The specification of soft reservations in cloud environments configurations with the use of the cloud environments modelling system affects the cloud migration decision. The cloud migration concept assesses each cloud migration alternative according to the cloud migration criteria and cloud environments configurations, including those that depend on the specification of soft reservations.

Soft reservations usually affect the score of a cloud migration alternative in terms of cost as they increase the accuracy of the predictions of cloud providers about resources' utilisation and the provisioning demand. That way, cloud providers can timely react to the changes in demand, which benefit cloud consumers as it offers cloud environment providers the possibility of reducing the price of their offerings to compete with other cloud providers[521].

[520] See Figure 6.2 in Section 6.4

[521] The research to offer the cloud migration concept for soft reservations was documented in two research papers to which this section very closely relates: the paper the author submitted with

6.8.1 The concepts of hard and soft reservations

Cloud providers that offer soft reservations as a resource allocation mechanism can more accurately predict their workload based on service monitoring data and cloud consumers' usage patterns. This lets them use hard and soft reservations to optimise resource allocation with the extra knowledge on how cloud consumers use resources and how they provision them.

Cloud consumers communicate workload forecasts in the long term through soft reservations for providers to compute the best strategy to allocate their resources. Once they are more certain of their resource requirements, cloud consumers actually claim (with a hard reservation) the resources booked through soft reservations. Soft reservations work, therefore, as an insurance policy that guarantees resource allocation at a cheaper rate to give cloud consumer incentives to put some effort into forecasting their upcoming resource consumption. The actual billing of the allocated resources only becomes effective when the cloud consumer issues the corresponding hard reservation. The same logic applies if cloud providers were to share their hardware utilisation data —with, of course, due respect to the privacy of all cloud consumers— to the cloud consumer. Although it is more difficult to economically motivate them to do this, it could increase the trust that cloud consumers put in providers.

Cloud consumers can use these two mechanisms to reserve and pay for the resources they allocate and even speed up the resources allocation. Given the additional information shared between cloud consumers and providers about their estimations through soft and hard reservations, cloud providers accurately estimate the resources to provision at any point in time and improve how they manage their resources. Cloud providers might, as an example, put physical machines to sleep to minimise resource consumption. In addition to the development of algorithms for cloud providers to react to consumer reservations, cloud providers can identify upcoming capacity changes to improve the placement of VMs at run time by taking into account the costs of the different reconfiguration options. These reconfiguration options are cloud migration alternatives within the cloud migration

Vahid Mohammadi, Adrian Juan, and other authors cf. Mohammadi et al. (2013) pp. 223–229 and to a previous paper Mohammadi et al. (2012), pp. 1–2 concept that semi-automatically, with the help of cloud consumers, weighs them according to their cloud migration criteria.

6.8.2 Cloud environments models with hard and soft reservations

Figure 6.13 shows a snapshot of the hard and soft reservations that multiple cloud consumers issued to a particular cloud provider. It lets cloud providers identify the time in advance with which cloud consumers planned their resources utilisation on the X-axis.

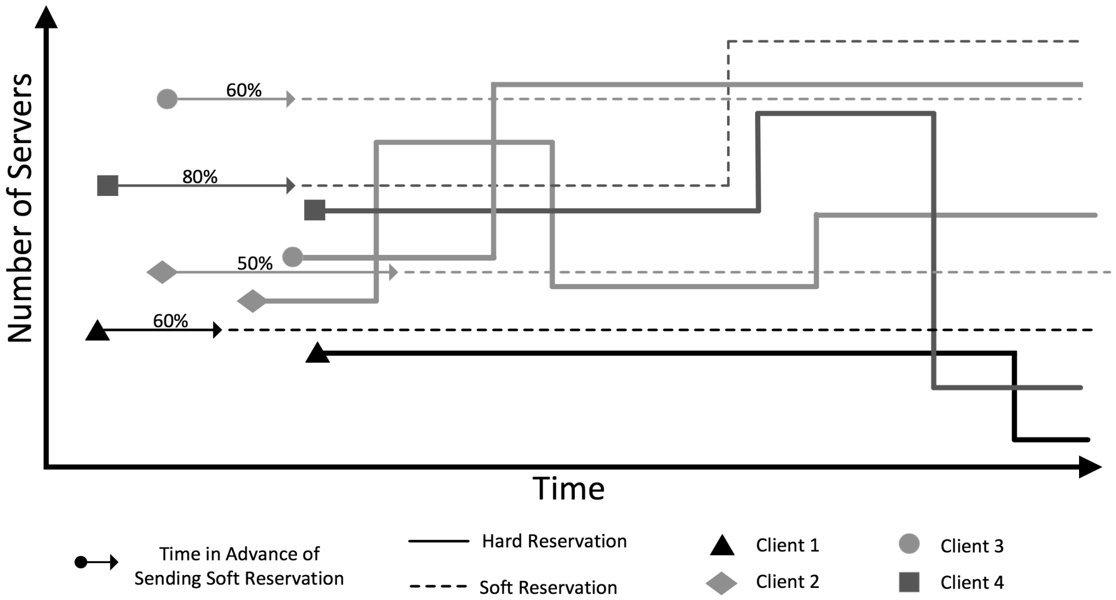

Figure 6.13: Snapshot of hard and soft reservations in the selected cloud environment[522]

Cloud consumers either issue hard reservations on the fly or make them correspond to previous soft reservations. With soft reservations, cloud consumers book resources in advance for a certain time range —being it minutes, hours, or days— to compensate the perceived risk of suffering from potential workload surges, while benefiting from obtaining resources at lower cost. Cloud consumers save costs if they correctly estimate their future resource demands and timely communicate them to cloud providers.

[522] Illustration by Vahid Mohammadi in collaboration with the author of this dissertation and others Mohammadi et al. (2013), p. 225

Still, cloud consumers can issue hard reservations, at higher cost, without previous soft reservations and obtain unanticipated capacity so that they accommodate the workload of their systems at that time.

6.8.3 Hard and soft reservations and how they affect cloud migration criteria

Soft reservations intend to benefit from the trade-offs they offer to cloud consumers and providers alike in conjunction with the higher cost of hard reservations. Soft reservations are much cheaper than hard reservations for cloud consumers as they just offer the right to gain access to a set of resources for a specific amount of time if they actually need them. Cloud consumers will pay some extra cost, if they unnecessarily allocate resources, but this expenditure will still be lower than the paying a hard reservation. This suggests researching into the pricing models to enact policies that strike a balance that avoids motivating cloud consumers to oversubscribe soft reservations that remain unclaimed. Non-allocated soft reservations would only work as an insurance policy for the consumer against high resource provisioning costs. In a similar situation to the one that Figure 6.13 depicts, cloud providers use issued soft reservations to plan their capacity on-line and to react to hard reservations by dynamically allocating new capacity. They might use standard mechanisms and provision servers that were on stand-by to use heuristics to plan according to hard and soft reservations over time. These mechanisms give providers incentives to lower the total cost of ownership (TCO) of the cloud environment infrastructure by planning their capacity.

6.8.4 Modelling the softness level of soft reservations

Four dimensions tune the soft reservations in a sort of softness level. The provisioning interval is the amount of time during which the soft reservation guarantees that the cloud environment will provision the requested resources after a corresponding hard reservation. A smaller provisioning interval guarantees a faster resource allocation upon hard reservation. The validity period is the time frame for which a soft reservation is planned.

A reservation for an extended time period, such as a month, might be more important than a soft reservation for a shorter period, such as a single day. Different validity periods might be subject to different considerations when cloud providers plan the capacity. The time in advance with which a cloud consumer issued a soft reservation before the validity period may also have varying priority. A soft reservation for next week could have higher capacity planning priority than another one for next month to timely accommodate the load. Alternatively, the oldest soft reservation might be assigned a higher priority due to its importance that triggered careful planning in the first place. Finally, the level of uncertainty defines the probability that the cloud consumer does not issue a hard reservation after a soft one or its likelihood of not needing the softly reserved resources.

6.8.5 Illustrative example of the proposed cloud migration DS concept for soft reservations

Figure 6.14 exemplifies two cloud environments with, on the right-hand side of the figure, soft and hard reservations and without them in the simple model that the left-hand side of Figure 6.14 shows.

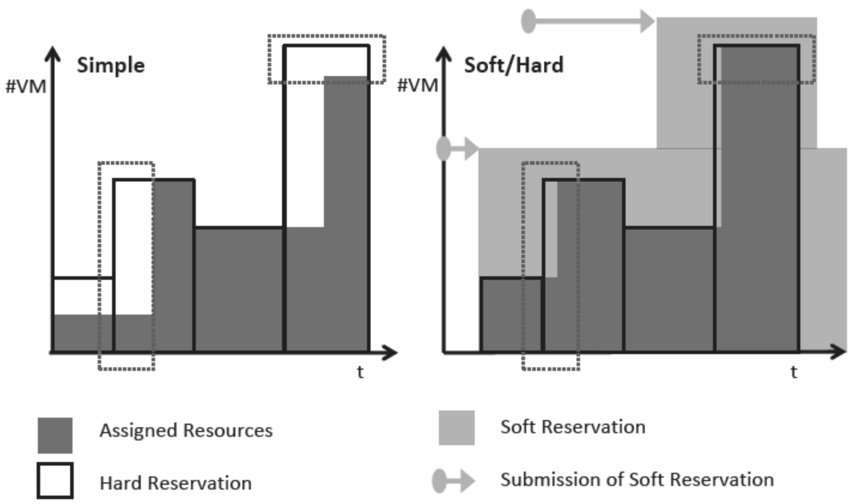

Figure 6.14: Example of computation utilisation in two selected cloud environment with and without hard and soft reservations[523]

Both parts of the Figure consider the allocation of resources across the Y axes (#VM) and time across the X axes (t). The example considers as fixed both the

[523] Illustration by Vahid Mohammadi in collaboration with the author of this dissertation and others Mohammadi et al. (2013), p. 226

provisioning interval and the level of uncertainty. The distance between the grey dot and arrow shows the time in advance with which the cloud consumer softly reserves the resources. The left-hand side of Figure 6.14 shows two potential problems with traditional model whereby there is some delay to allocate the resources the cloud consumer requested via a hard reservation. The vertical box in the plot on left-hand side of Figure 6.14 exemplifies this delay that happens because the cloud consumer directly issues a hard reservation and therefore the cloud provider cannot plan for this event and reacts on the fly. The horizontal box in the plot on left-hand side of Figure 6.14 exemplifies shows this delay combined with a situation whereby the cloud environment under-provisions the cloud consumer. The right-hand side however minimises these problems. The cloud consumer issues a soft reservation beforehand and lets the cloud environment plan to timely allocate the requested resources with less delay. The soft reservation lets the cloud provider find the resources by postponing a non-urgent task such as, for example, a routine calculation based on the data in the system. The success of the soft reservation lies within the ability of cloud consumers to plan their capacity according to the future demands that the cloud consumer estimates.

The proposed concept includes reservations in the cloud environments model to assess the cloud migration alternatives in accordance with the consumer cloud migration criteria. Soft and hard reservations affect the assessment —in terms of costs, performance, availability, and scalability— and have, therefore, to be taken into account to support the cloud migration decision.

6.9 Cloud migration DS concept for cloudlets

This section summarises the adaptations to the proposed concept to address network problems with the use of geographically near cloudlet-based micro data centres with the capabilities to do the necessary processing. Like Section 6.8 with soft reservations, this section explains the customisation and mechanisms needed to enable the cloud migration concept to assess the value added by cloud migration alternatives that include cloudlets as potential computational nodes. A number of standard base VMs and the pre-configured VMs run in the cloudlet. The proposed cloudlet-based migration to cloud environment uses resource-rich components that the cloud migration concept describe in the architecture that deploys some software components to virtual machines on the cloud and cloudlet premises. As a result, these resource-rich components running in near nodes timely respond to resource-poor nodes or mobile devices[524].

6.9.1 Motivation to consider cloudlets in the proposed cloud migration DS concept

Mobile broadband communications allow people who live in rural areas to communicate with the rest of the world. In regions with lacking communication resources via cable-based infrastructures, increasingly developed wireless networks can unlock new business ventures. That combined with the use of cloud computing and the market penetration of smart phones market enable more users to use computing via their mobile devices. Even if mobile networks are quite pervasive today and allow people to use cloud-based services, a limited set of cloud providers do exist in countries with a majority of low and medium-low income citizens. As a result, mobile cloud users access infrastructure at far distances and incur in wide-area communications. That introduces at times latency problems and longer response times that create unacceptable mobile user experiences.

6.9.2 Illustrative example of cloudlets

As an scenario motivating the research, a region with sub-par communication networks could be inhabited by a group of coffee farmers who own large coffee plantations. They might want to use mobile devices to record data about their coffee plants on a daily basis so as to get data-based suggestions to undertake or calculations to better production and help where it is best to offer their product. As an example, some regions might experience severe drought some months and should increase the frequency of irrigation. If farmers were to cloudify their systems to benefit from using additional collaborative storage and elastic computation while being able to use cloudlets as a middleware holding pre-computed or cached results for mobile devices in the geographical area of the cloudlet to increase

[524] Tesgera and the author of this dissertation worked on this particular application of the proposed cloud migration concept to scenarios implementing cloudlet-based computing. This section relates very closely to the published research paper: cf. Tesgera et al. (2014), pp. 253–253

performance. The cloud migration concept assist organisations in designing mobile cloud-based application systems so that they use cloudlets to help, in this case, coffee plantation owners access geographically near services to accommodate the challenges to access the back-bone cloud environment in regions with networks with low bandwidth and high latency.

6.10 Cloud migration concept tailored to the evaluation

The evaluation of the proposed concept, as Chapter 7 explains, applies the cloud migration concept to three realistic cloud migration scenarios. They offer the opportunity to take the proposed concept out of the lab and apply it to the cloud migration of real-life application systems. However, these three scenarios, being different from each other and quite specific, require adapting the proposed concept, whose general architecture Section 6.4 describes, to the particular scenarios of migrating application systems to PaaS offerings, the case of cloud migrating BI application systems, and the question of how to best use cloud environments for collaborative networks and their systems.

The following sub-sections explain the motivation to use these three scenarios in the evaluation of the proposed concept and the necessary concepts (specific for each scenario) to be understood to adapt the architecture of the proposed concept to them. In addition to offering an initial hindsight into how to adapt its architecture to fulfil the requirements of each of these three scenarios, the following sub-sections abstract from these customisations of the proposed concept to provide some details into how to re-use the concept and its sub-components in other fields of application and scenarios out of the scope of this dissertation. Hence, how to apply the proposed concept to future work in cloud migration —see Section 8.4, Chapter 8, for more details on research beyond the scope of this dissertation.

6.11 Scenario 1: cloud migration concept for PaaS offerings

The architecture of the proposed concept is tailored to the evaluation scenario to assess the effects of applying the concept to the cloud migration to PaaS offerings. Its prototypical implementation realises then the functionalities described in the architecture explained below. Section 7.2 shows in Chapter 7 the actual assessment[525]. The architecture of the proposed concept[526] is tailored to the particularities of PaaS by using model-driven approaches, as Section 6.7 explains, in order to evaluate interoperability, standards, and portability at the PaaS level.

6.11.1 Cloud migration criteria of PaaS consumers

PaaS providers supply computing platforms at a higher level of abstraction than IaaS, whereby the cloud-enabled target application system uses the PaaS operating system, programming language, execution environment, database, and web server. All this opens up the door to integrating customised provider-dependent tools as a part of the proposed concept for cloud migration decision-making support.

PaaS has established itself as the lynchpin for organisations to take advantage of the decreased IT costs and the increased development speed because organisations use standardised software components provided by platforms. For organisations, the decision of which PaaS cloud service model to choose for their application systems is becoming increasingly complex. Likewise, the decision on how to change from one cloud service model depend on an ever-increasing number of cloud migration criteria. This is due to the PaaS cloud offerings heterogeneity that support different runtime platforms or services. Some of these criteria are particularly relevant for PaaS migration that makes organisations prioritise acquiring their technological agility while minimising the vendor lock-in effect they suffer. The proposed concept assists organisations in building an architecture that

[525] The architectural adaptation explained in this section closely relates to a paper published by the author of this dissertation, cf. Surajbali and Juan-Verdejo (2014), pp. 275–282

[526] Figure 6.2 describes the architecture of the proposed concept before customising it to the needs of any evaluation scenario

extends and migrates their application systems according to their future needs stated in the cloud migration criteria while focusing on tackling PaaS vendor lock-in issues.

Given the increasingly complex ecosystem of heterogeneous PaaS offerings and the PaaS-specific cloud migration criteria; it becomes increasingly difficult to take into account the holistic impact of the cloud migration when selecting a PaaS offering. Some criteria that are specific to the case of migration to PaaS environments relate to the efficient deployment, the security level that the PaaS provider offers according to the secure protocols they use, the QoS guarantees or cost-sharing models they offer, and trust issues that organisations might have with them. Especially at the PaaS level, organisations face vendor lock-in issues due to the lack of PaaS providers that comply to existing standards. This makes porting data, application systems, and infrastructures difficult. But even if the portability cloud migration criteria is supported in theory, the high complexity and the switching costs still might discourage organisations from porting their application systems to a different PaaS environment[527].

The proposed concept intends to overcome this hurdle by providing the cloud portability tier to let PaaS consumers use its API via the broker. The use of software components and APIs modularises concerns with components that only do one thing. This, in turn, helps improving the portability and might arguably facilitate integrating the concept in marketplaces of PaaS offerings to achieve higher levels of exploitation.

This potential integration might be a factor to speed up the adoption of the proposed concept as PaaS consumers and providers might decide to take on the concept for their cloud migration projects after they could test it within a marketplace free of charge.

6.11.2 Concept architecture of tailored to PaaS migration

Figure 6.15 adds the cloud portability tier and the cloud migration broker to the basic architecture in Figure 6.2 in Section 6.4. The portability tier supports the

[527]cf. Juan-Verdejo and Surajbali (2016), pp. 11–23; Molano et al. (2018), pp. 712–719

migration from one PaaS offering to a different one. The broker allows end users to track the offerings at the Infrastructure-, Platform-, and Software-as-a-Service levels as well as the services they provide and the interdependencies between different cloud migration criteria. However, its main goal is to make it easier for cloud adopters to integrate and extend cloud services so that they are more productive and can offer better service provision and maintenance.

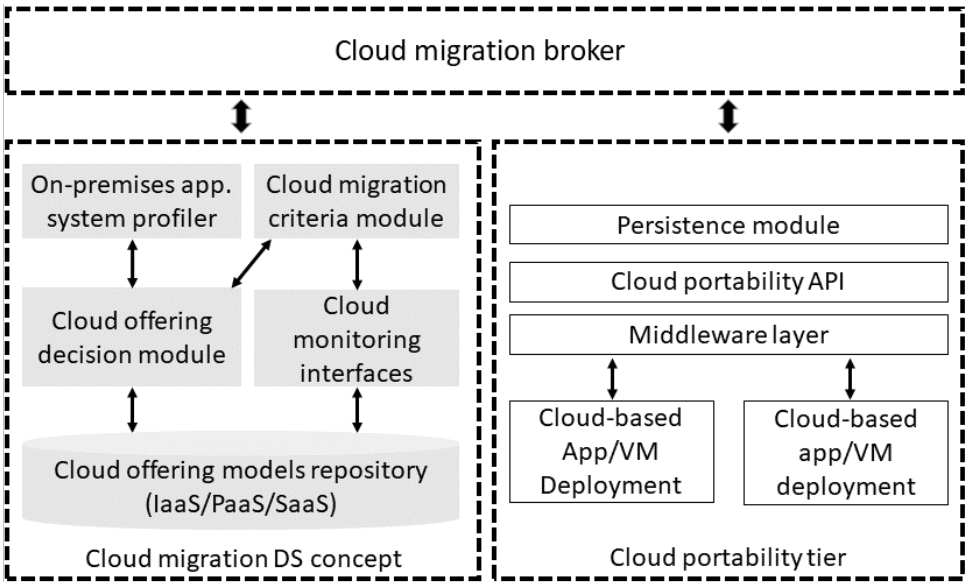

Figure 6.15: Initial customisation of the architecture of the proposed cloud migration decision support concept[528]

The broker mediates between PaaS consumers and the cloud migration decision support and cloud portability tiers[529]. Accordingly, it offers services that provide value to cloud consumers and to providers as well. The broker in that manner intends to help organisations find appropriate cloud offerings according cloud migration criteria. In addition, the proposed concept assists them in the decision-making processes of how to cloud migrate their application systems. The proposed concept uses the broker to design how to, for example, migrate the business logic to one cloud offering and the data layer to another one. The broker enables that without neither modifying the settings of the target application system nor using additional software while avoiding to put the application system and the data it guards at risk. The cloud migration concept lowers the decision effort for organisations to pick a different PaaS offering to which to port the application system.

[528] Own illustration partially based on the PaaSport Deliverable D1.2 PaaSport Consortium (2014b), p. 11

[529] Section 7.2 provides more details about the cloud migration broker in Chapter 7

The concept saves development time and enables the transparent exchange of application system and data components with different cloud providers.

Cloud migration broker

The proposed architecture envisions a cloud migration broker that mediates the migration of organisations' software solutions to virtualised cloud environments. It abstracts from the details of Figure 6.2 within Section 6.4to offer a re-usable component that addresses the challenges of the migration to PaaS offerings and, potentially, for the migration to other levels of the cloud stack. This would be a contribution to other decision-making processes to migrate systems to cloud environments.

Section 6.6 works as input to the broker with the explanation of how the selected cloud service model affects the decision making for cloud migration and how to adapt the target application system so that organisations can migrate it. As cloud service models build on top of each other, when migrating to PaaS offerings, the underlying infrastructure is the IaaS level at which organisations deal with virtual machines, servers, network, or storage. PaaS cloud services offer more flexibility than SaaS offerings that make organisations agree to the legal constraints, the SLA, and the pay model prior to moving the data layer to their cloud premises.

Modelling cloud migration projects to PaaS with the proposed concept

The cloud migration broker builds on top of the proposed concept as well as the cloud portability tier in order to facilitate using the proposed concept for the migration to PaaS environments. This way, PaaS cloud consumers can use the cloud migration and portability functionalities that use the Analytic Hierarchy Process to formalise part of the the cloud migration multi-dimensional problem. That is, organisations can define their cloud migration criteria, the cloud environment configurations, and the architecture and properties of the application system as input models to the cloud migration concept.

The proposed concept weighs viable cloud migration alternatives through the broker which assesses the value of metrics of the cloud migration criteria according to the importance of each criterion with respect to other criteria. In those cases for which the concept does not provide any qualitative metric, the organisation may manually assess the weight of each alternative for each PaaScloud migration criterion. The concept supports the migration to PaaS according to the many interdependent organisations' cloud migration criteria by facilitating the complex cloud migration decision process. It weighs cloud migration alternatives using different PaaS offerings or that plan to move between PaaS environments in order to help organisations select the best cloud migration alternative for their cloud migration project. This evaluation scenario spotlights cloud migration criteria such as application system portability and the prevention of vendor lock-in issues.

Cloud portability tier

Following the recommendation by the cloud offering cloud decision support tier, the cloud portability tier assists in porting the application system to a different PaaS environment and in planning the deployment of the application system. The persistence module and cloud portability API take care of the processes to port and deploy application systems. The persistence module manages the dependencies of application systems in the deployment, un-deployment, start, stop, and migration to cloud environments of the target application system. These dependencies relate to the executable formats, compilers, libraries, and operating systems. The persistence module considers these dependencies and allows for the addition of more by developers. The cloud portability API allows managing the design of the deployment of application systems, independent of the specific API of the cloud environment. Figure 6.15 shows the middleware layer as an adapter bridging the gap between the native API offered by the cloud service and the cloud portability API module. The middleware bidirectionally converts the functions between the cloud portability API and the native API. Thereby, it enables the seamless communication between cloud services and address vendor lock-in concerns.

Similarly to the proposed concept, multiple standardisation bodies work on unifying incompatible cloud APIs conceived by cloud providers[530]. Portability standards are a concern when migrating application systems from one cloud provider making

[530] Cloud API standardisation efforts include multiple standards cf.CAMP OASIS (2014), pp. 29–39, URL see Bibliography), Open Cloud Computing Interface or OCCI (Metsch et al. (2010), p. 3), and Cloud Infrastructure Management Interface or CIMI (Cloud Management Initiative (2021), p. n/a)

use of one standard to another cloud provider using a different one. Furthermore, the proposed middleware layer supports cross-standard implementations so that standards interoperate like proposed in some research papers by Bromberg[531]. That's an approach to solve standard incompatibility in portability across cloud models. That is, when a cloud environment adheres to a cloud standard, such as, for instance, OCCI IaaS, and another one follows a PaaS standard such as CAMP.

6.12 Scenario 2: cloud migration concept for business intelligence

Organisations choose their cloud service model depending on the cloud environment capabilities and requirements. However, the task of choosing their provider can be difficult as organisations must also think of potential compatibility issues with their BI software components, the services offered by the cloud environment, and potential interoperability and portability issues. To illustrate the complexity around selecting a cloud offering and adapting their application systems to them, the cloud migration DS concept considers three particular instances of cloud migration of business intelligence (BI) systems[532] as an explanatory example of the migration of a standard shown in Figure 6.16.

6.12.1 Architecture of BI application systems

The adaptation to this scenario includes offering cloud migration templates for the cloud migration of BI systems and customising the cloud migration criteria. The cloud migration concept includes metrics to automatically weigh BI cloud migration criteria for each cloud migration alternative. Usual BI cloud migration criteria include agility and time to value as the development cycle speeds up when using cloud infrastructures and cloud solutions are ready to scale according to the particular temporary needs of the system and the market. The proposed cloud migration concept weighs the different cloud migration alternatives according to

[531] Bromberg et al. proposed cross-standard implementations cf. Szvetits and Zdun (2016), pp. 55–58; Bromberg et al. (2011), p. 446

[532] cf. Juan-Verdejo et al. (2014a), p. 43

the total cost of ownership and assist organisations by letting them compare their cloud migration alternatives within that dimension.

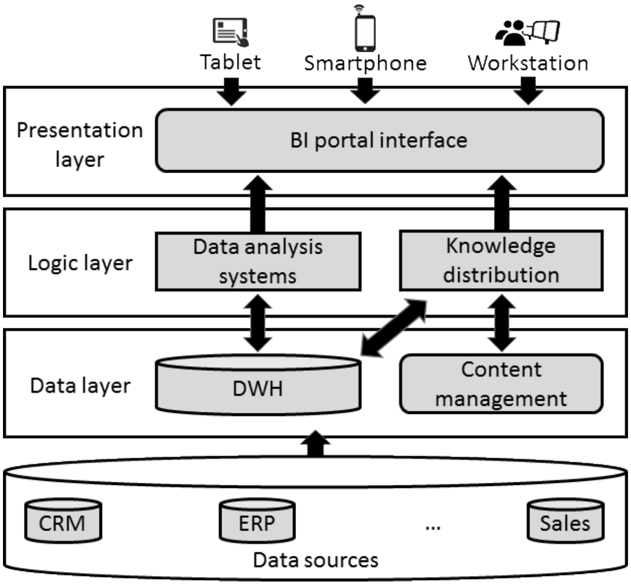

Figure 6.16: Business intelligence system running on the organisation premises[533]

In addition to the agility, cloud environments provide services that cloud-enabled BI systems can use to be more flexible and accelerate the development life cycle as organisations do not have to deploy those services in their local premises but use services from real-time streaming solutions to services for Dare Warehousing and for Big Data processing. Additionally, cloud infrastructures potentially offer higher levels of security than the local premises of organisations running BI systems as cloud environments typically offer authorisation mechanisms, as well as data-level security with encryption of data in motion and at rest. Cloud offerings can focus on their core business that includes security aspects that motivate them to differentiate their products offering services to enable IP restrictions as well as auditing and threat detection services.

The proposed concept, as the left-hand side of Figure 6.2 in Section 6.4 shows, lets BI software engineers rate the cloud offerings in the cloud offerings repository rate based on their experience. They can leave a comment and rate a particular cloud provider and offering, expressing their satisfaction or dissatisfaction with regards to quality, usability, reliability and user-friendliness of the cloud offering. This is of value for future software engineers who can leverage each others' experiences in

[533]Own illustration based on Juan-Verdejo et al. (2014a), p. 44

cloud migrating or in porting an application system like the one shown in Figure 6.16 to a particular cloud offering to improve their cloud migration projects. Section 7.3 within Chapter 7 delves into the topic of cloud migration of business intelligence systems and evaluates the fitness for this purpose of the proposed concept.

6.12.2 Cloud migration alternatives for BI applications systems

Many BI models follow a three-tier architecture consisting of a presentation layer, a logic layer, and a data layer. Structured and unstructured data comes from operational systems for the data layer to store it. The logic layer uses different implementations of methods to analyse those data. Finally, the access layer enables BI users to obtain the high-level data they need. The adaptation of the proposed concept to the particular evaluation scenario that Section 6.12 explains in Chapter 6 entails adapting the cloud portability tier on the right-hand side of Figure 6.2 (see Section 6.4) with the considerations below about the different implications of porting BI systems to cloud environments.

Cloud migrating legacy BI application systems

Organisations have different interdependent constraints and cloud migration criteria to port their BI application systems from one cloud environment to another one; and want to achieve specific objectives in different dimensions. Nevertheless, when they improve the system in a dimension, as increased data storage, they could negatively affect the application system in another one like for instance cost reduction. As these dimensions are allegedly not of equal importance to organisations and they are usually ready to trade one for another. An organisation cannot overcome some of the constraints. As an example, one organisation might not be allowed to move its BI data to a public cloud environment. The nature of the application system components to be ported, the application system's users, and the offering gained by porting the BI application system to cloud environments and how to choose the best offerings are crucial decisions for organisations.

Partial cloud migration of BI business logic or data layers

An organisation can choose to move only part of the BI layers to cloud environments. For instance, for security reasons, an organisation may choose to keep the data layer on its premises while using BI tools in cloud environments to extract information from those data. At the same time, the organisation needs to make sure the system's performance is not affected by wide area communications introduced by partially migrating the BI business logic or data layers.

Portability of a cloud-based BI application system from one cloud offering to another

The BI software engineer must evaluate each cloud offering to reduce the effort to adapt its BI application system to the new cloud environment. The selection of the cloud provider and cloud offering that requires less adaptation of the ported application system reduces costs. The adaptation effort depends on the levels of transparency achieved when porting an application system. That is, the portability achieved without using any additional software, changing the application system settings, or putting the application system and data at risk.

6.13 Scenario 3: cloud migration concept for collaborative networks

Collaborative networks systems benefit from using cloud environments as organisations can virtually connect and make alliances to co-create and design products and services together. Collaboration systems once moved to cloud environments provide platforms for dynamic continuous collaboration on a global scale to co-design and operate massively customised service-enhanced products. This section summarises how the architecture of the cloud migration concept —the architecture in Figure 6.2 within Section 6.4— was adapted to address the needs to cloud-enable collaborative systems. The architectural adaptations and changes to the existing models that this section describes prepares the grounds to implement the prototype of the cloud migration concept[534] to evaluate the cloud-based collaborative networks scenario as Section 7.4 describes in Chapter 7.

[534] These changes and architecture adaptation is summarised in a previous paper, cf. Juan-Verdejo and Surajbali (2016), pp. 11–23

6.13.1 Architecture of collaborative networks systems

Advances seamlessly connecting the digital and real world to enhance collaboration come now in different flavours and names. From the Internet of Things (IoT) to Industry 4.0 and smart production, they entail collaboration over networked and intelligent Cyber-Physical Systems. Cloud computing combined with advancements in Cyber-Physical Systems, presents new opportunities in manufacturing to produce high value-added products that can quickly reach the market. Both the Internet of Things (IoT) and Internet of Services bring about the fourth industrial revolution. Digitalising logistics and techniques around manufacturing can drive the improve factories with machine-to-machine communication used to tailor highly customized products to individual customer needs. Cyber-Physical Systems are connected and use cross-organisationalservices in addition to internal ones to collaborate in real-time. That way services contribute to increase the business agility of manufacturing companies and offer them additional flexibility. These architectures leverage cloud-based services from various cloud providers as well as data stored in them through different interfaces to share them. With these architectures still organisations usually need to keep on preventing the formation of data silos which could result in vendor lock-in issues due to differences in service interfaces among providers. The cloud migration concept uses cloud environments at different abstraction levels including the Infrastructure-, Platform-, and Software-as-a-Service cloud models to profit from the cloud offering that better match their needs over time. Cloud-based collaborative systems can this way use multiple cloud environments that allow stakeholders to share resources with each other over the Internet in a controlled manner.

The evaluation of the cloud migration concept that Section 7.4 explains in Chapter 7 combines collaborative systems with the advancements in embedded intelligence that mobile devices and Cyber-Physical systems help deliver a variety of application systems serving the environmental, civilian, military, and government sectors in addition to industry and aerospace. Despite the widespread use of these interconnected collaborative systems, their limited computing power and battery life limit their ability to run computation-intensive tasks. To address this issue, Cyber-Physical Systems use cloud environments to outsource part of the computation, which is a crucial step towards the implementation of Industry 4.0. The concept of Industry 4.0 also relates to what is known as the Smart Factory, which is characterised by Cyber-Physical Systems that monitor physical processes and create a virtual representation of the physical world to undertake decentralised decisions in an effective and accurate manner[535]. Cyber-Physical Systems are interconnected to collaborate and cooperate through internal and cross-organizational services to provide value-added complex products and services. That is usually named, the Internet of Things (IoT), which is connected with the industrial automation that increases the business agility and flexibility of organisations adopting these collaborative systems as well as letting them achieve mass customisation at lower costs.

6.13.2 Cloud migration criteria

Cloud computing acts as the stepping stone to build Industry 4.0 and IoT systems that use Cyber-Physical Systems and virtually infinite cloud resources to process and store analysed information so as to share them in real time with any device anywhere.

The adoption of cloud infrastructures presents, in addition to great potential to let upcoming technologies interplay, some challenges related to how to select a cloud environment to avoid data silos and lock-in issues. Organisations need to decide which cloud environments to use, but choosing a cloud service model for their applications can be difficult and adapt the architecture and workflows as well as their business processes can become costly. Heterogeneity in cloud offerings adds complexity to the decision-making process and desigh to cloud migrate an application system. In Industry 4.0 that use collaborative systems, rapid and accurate decision-making can lead to increased productivity and flexibility while keeping high quality.

Huge loads of data from internal and external sources are used by companies and organisations in various formats. Collaborative systems usually analyse those data in real-time to help solving problems in an efficient manner while improving

[535] cf. Hermann et al. (2016), pp. 3934–3936, 3928; Kagermann et al. (2013), pp. 18–26; Brettel et al. (2014), pp. 37–38

existing processes. Given the large volume of data and tight time constraints, these systems analyse the data in an automated manner. Models of existing knowledge domains, combined with data analysis processes, automate the decision-making processes required to achieve effective outcomes. Such as decisions. The integration of Industry 4.0 and Cyber-Physical Systems within a cloud-enabled environment provides additional intelligence and knowledge captured through their sensors and networks. That opens the path to using on-demand computation and memory to analyse data with advanced modelling and context analysis or correlation that takes into account the shared data in a collaboration and customised manner. Modern cloud computing architectures also usually support product development using microservices that can be tailored by consumers to enable them to configure products and services while reconfiguring manufacturing systems. Industry 4.0, cloud-based design and manufacturing, and cloud manufacturing apply the cloud computing paradigm to computer-aided product development and IoT systems.

The challenge of migrating collaborative and Industry 4.0 systems to cloud environments stems, in part, from the difficulty of selecting a particular cloud offering to adapt the target application system to best profit from using those cloud-enabled collaborative systems. Organisations migrating collaborative systems consider interdependent constraints and cloud migration criteria specific to migrating their target system.

They typically trade one cloud migration criterion off with others across the entire cloud stack as Section 6.6 describes. This is in fact one of the adaptations applied to the proposed concept so that it is used for the particular scenario of cloud migrating collaborative systems. These adaptations include designing specific metrics to assess the cloud migration criteria for the different cloud migration alternatives for collaborative networks systems. Cloud migration criteria that range from the usual motivation to using cloud infrastructure, reducing costs, to providing an ubiquitous and simple access to the collaborative systems over the internet with well-known protocols.

Organisations present cloud migration criteria for centralised platforms for collaborative environments. They define the need for flexible cost models, modularity, and extensibility. Collaborative systems generally benefit from scalability in multi-tenant environments but their cloud migration criteria vary for different organisations. In collaborative systems, security and trust lie at the foundation of what make these systems work over the Internet and therefore it is one of the fundamental aspects to implement these systems.

6.13.3 Everything as a Service for collaborative networks

Collaborative systems in Industry 4.0 settings enables the seamless integration of changes in this network. That allows manufacturers to anticipate disruptions in the supply chain by simulating and evaluating the best supply alternative to then choose it. Organisations embracing the principles of Industry 4.0 have unique requirements for their complex systems and can benefit from the selection of specific cloud offerings at various levels of the cloud deployment stack. Utilising multi-cloud environments can also minimize the risk of data silos posing portability issues. The architecture outlined in Figure 6.2 within Section 6.4 uses, in the third scenario, a cloud migration broker to schedule and deploy components to the appropriate XaaS (Everything as a Service) level cloud environment, as well as to access services and microservices across the entire cloud service stack.

Collaborative systems have clear needs for functionalities that support organisations in setting collaborative spaces with different roles and permissions that govern the access to the data they share. The provision of these services become a part of the decision-making process of cloud migrating collaborative systems as well as the different mechanisms that cloud providers provide in their cloud offerings to enable organisations to share knowledge and data across virtual organisations. Collaborative systems need interoperability as they include different systems that should communicate and maintain the knowledge of the virtual organisation across its entire life cycle. Virtual organisations that should be very easy for organisations to access. Finally, the cloud environments model will be adapted to include whether the cloud offering provide groupware-related services to let organisations collaborate as it is an important cloud migration criteria for many of them. These functionalities come together with the requirement to provide functionalities to create and manage collaborative spaces as well as to go beyond interoperability and offer seamless service integration.

This section provides the first adaptations to the proposed concept needed to let the prototype assist organisations in designing cloud-based architectures for collaborative systems across multiple layers of the cloud stack. Cloud-based architectures to avoid portability and vendor lock-in issues while unravelling the whole potential behind modern computing architectures for IoT deployments for Industry 4.0 settings.

6.14 Discussion on the proposed concept

The outcome of pursuing the third goal of this dissertation —developing the cloud migration concept— is documented in this chapter (see Figure 6.17). In addition to Chapter 6, Chapter 7 also comes back to this result as the development of the proposed cloud migration concept happens in an iterative and incremental manner.

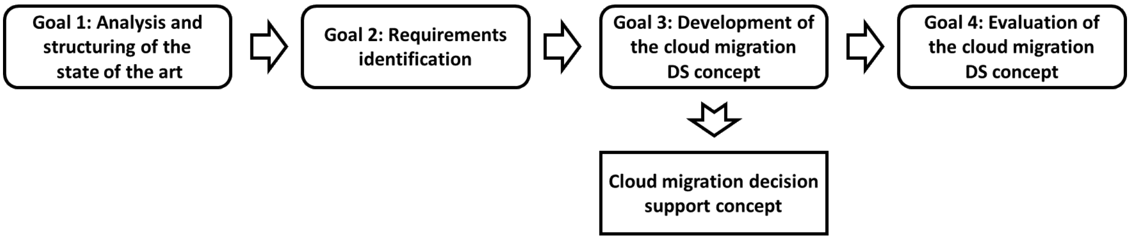

Figure 6.17: Result of the third goal[536]

The development of the proposed concept involved coming back and forth from the concept and design to the prototype implemented for its evaluation. The presented concept and formalisations for multi-criteria application system cloud migration was developed for this iterative and incremental evolution facilitated by the architecture of the proposed concept designed for flexibility and rapid evolution to enhance the presented metrics and include new ones. The result of the third goal is the proposed concept so that it works as the input of the evaluation of the formalisation and the proposed metrics. Achieving the fourth goal requires the prototypical implementation of the concept that Chapter 6 describes in order to evaluate the proposed cloud migration concept in three evaluation scenarios.

The multi-dimensional decision-making process that the proposed concept carries out to migrate application systems to cloud environments is explained in this chapter. Additionally, this chapter formalises the cloud migration in the proposed

[536]Own illustration

concept by using the cloud migration criteria, the target application system, and the model of the cloud environment configurations. With the aim of helping organisations cope with the multi-sided effects of the cloud migration, the proposed cloud migration concept builds on top of the AHP to assist organisations to cloud migrate their application systems and assist organisations in adapting their application systems to cloud environments according to their many interdependent cloud migration criteria.

Potential metrics for objective and subjective criteria for cloud migration are presented to assess accountability, agility, assurance, cost, performance, security, privacy, and usability. For the sake of clarity, several examples of these cloud migration criteria are presented to explain how the proposed concept trades a criterion off with others and how it only generates alternatives that comply to the application system and organisational constraints.

To cope with organisational constraints related to privacy, some cloud migration alternatives re-scatter application systems' components to both cloud environments and the local premises to alleviate sensitivity issues. The proposed concept weighs these alternatives by taking into account the proposed cloud migration criteria metrics and their importance relative to other criteria. When the proposed concept does not provide any metric for a particular cloud migration criteria or sub-criteria the organisation manually assists in weighing.

7. Scenario-based evaluation

The scenario-based evaluation pursues gaining insight about the proposed concept and how its capabilities perform in an operational context; namely, the provision of cloud migration support to organisations involved in three specific scenarios. The analysis of what works and does not work, when applying the prototype to the praxis of cloud migration projects, helps detecting chances to iteratively and incrementally improve the concept design. In addition, this chapter assesses whether the proposed concept meets its objectives and answers the four research questions laid out by this dissertation[537].

The evaluation strategy follows the two complementary phases shown in Figure 7.1. These phases reinforce and affect one another by learning from their intermediate findings and, more specifically, from the analysis of the results of iteratively and incrementally executing them. In those iterations, firstly, the prototypical implementation and testing of the proposed concept takes place, which results in releasing the first version of the prototypical implementation of the proposed cloud migration DS concept; also called InCLOUDer[538]. This first evaluation phase includes the three tasks on the left-hand side of Figure 7.1 followed by the phase designed to apply the prototype to the three structured scenarios used to evaluate the proposed concept in realistic cloud migration projects (see the right-hand side of the figure).

[537] Both the objectives of this dissertation and the research questions can be reviewed Section 1.2 which explains the aim of this dissertation in Chapter 1

[538] The author of this dissertation also named the prototypical implementation of the proposed cloud migration DS concept as InCLOUDer in a previous paper: cf. Juan-Verdejo et al. (2014b), pp. 467–474. More details on InCLOUDer are provided in Section 7.1

The prototypical implementation phase is re-executed to improve the prototype according to the lessons learnt as the evaluation scenarios phase is carried out and as organisations' needs become more apparent. The three real-world scenarios, to which the prototype is applied, stem from industrial and research collaborations to migrate: application systems to PaaS and business intelligence application systems as well as collaborative networks to cloud offerings.

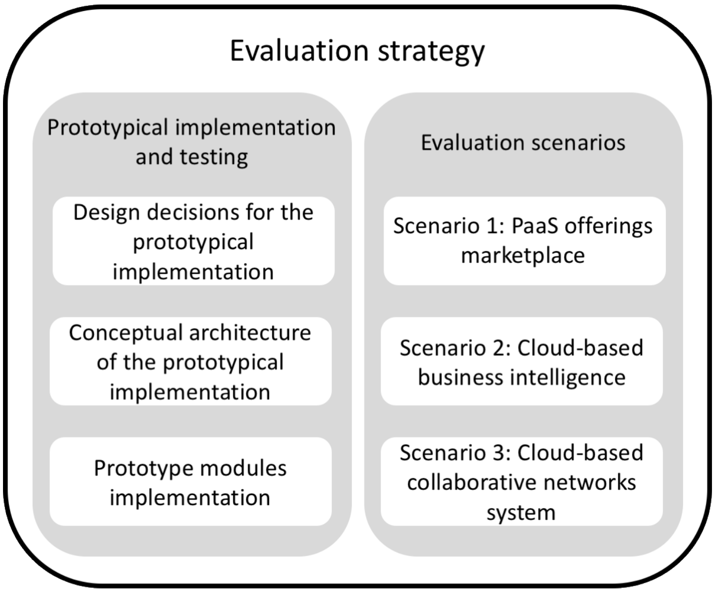

Figure 7.1: Evaluation strategy[539]

In addition to fitting into the overall evaluation strategy shown above, each evaluation scenario builds systematically on the five components of the research frame of reference[540]. In addition to being used for designing and building the cloud migration concept, the research frame of reference is also used to guide the evaluation of each evaluation scenario according to the principles of design-oriented research[541].

The evaluation scenarios start with the holistic cloud migration support —that is, the first component of the research frame of reference— to then delve into the details of on-premises application system modelling, cloud environments modelling, cloud and inter-cloud architectures, and cloud migration criteria —the other four components shown in Figure 4.13. Based on the research frame of reference, the research artifacts are proposed in the integrated concept design for each of

[539]Own illustration
[540](The research frame of reference is shown in Figure 4.13 in Section 4.7 of Chapter 4)
[541]Section 1.5 lays, in Chapter 1, the principles of design-oriented research

the three evaluation scenarios by specifying their functionalities and role towards achieving their intended goals. The context of this chapter is framed in Figure 7.2 around Goal 4 —that is, the evaluation of the cloud migration DS concept— and the last artifact of this dissertation: three scenarios successfully evaluated by via a prototype.

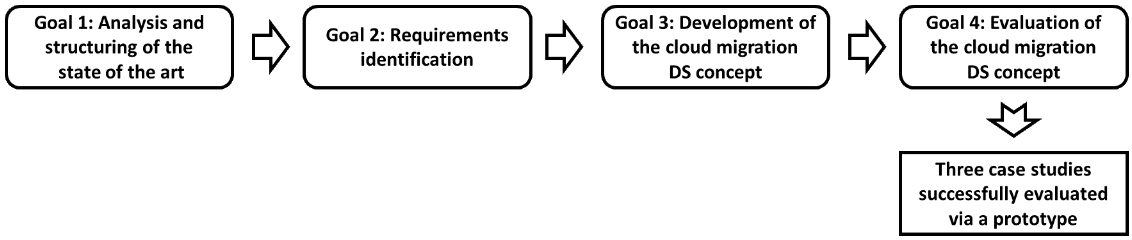

Figure 7.2: Result of the fourth goal[542]

The evaluation leverages multiple research methods. Prototyping and scenarios to evaluate the proposed concept. Interviews, questionnaires, and workshops to gather feedback assessing the proposed concept. Brainstorming and workshops are also applied to get intermediate feedback and, accordingly, steer the development of the prototype and the further specification of the scenarios[543]. The development of the InCLOUDer prototype is based on the proposed concept (prototyping research method) and it is applied to three scenarios (scenarios research method)[544]. The implementation of the conceptual design[545] as a prototype is the mechanism chosen to test and evaluate its applicability to real-life cloud migration projects. As the prototype is a software tool with which the research community can reproduce the evaluation scenarios, they are thoroughly described in Sections 7.2 to 7.4[546]. In addition, the software architecture of the prototype aims at making it an extensible work that the research community can use in domains different from the areas of application of the three evaluation scenarios; hence, out of the scope of this dissertation.

[542] Own illustration

[543] See Section 4.7 in Chapter 4 for more details on the used research methods

[544] This chapter, combined with Chapter 6, fulfils the final phase of the research design shown in Figure 6.1, Chapter 6, the iterative development and evaluation, to deliver the fourth goal of this dissertation, the evaluation of the cloud migration concept

[545] Chapter 6 explains the conceptual design of the cloud migration DS

[546] Reproducibility ensures consistent results when using the code of the prototype on the same data. Replicability entails obtaining consistent results with a different concept, data, or method used to answer the same scientific research questions, which are formulated in Chapter 1

7.1 Prototypical implementation and testing

The prototypical implementation and testing phase[547] involves domain experts slightly aware of this research and researchers from other disciplines. Their tasks include testing the prototype in order to report its usability and fidelity to the concept so that their feedback is used to alter the proposed concept[548]. This phase is iteratively and incrementally executed, starting by taking design decisions for the prototype, which evolve from the requirements of the proposed concept (see Chapter 5) to propose technological solutions that fulfil them. Next, the architecture of the prototype is designed, according to those design decisions, so as to facilitate integrating the prototype modules that implement the functionalities of the proposed concept.

The implementation of the prototype used in the three evaluation scenarios, is based on the Eclipse Rich Client Platform (RCP)[549]. The prototype allows organisations to model the on-premises target system, their cloud migration criteria, and the potential target cloud environment configurations. The InCLOUDer prototype generates a set of cloud migration alternatives that it recommends, in order, according to the ranking created using the Analytic Hierarchy Process (AHP)[550]. The implementation of the prototype ensued after the iterative and incremental design, implementation, and refinement of the five software modules shown in the architecture that Figure 7.4 depicts. That is, the modules to model the on-premises application system, the cloud migration criteria modelling, and the cloud environments in addition to the generator of the cloud migration alternatives and the module to semi-automatically weigh them. These software modules are combined into the prototype in order to deliver a faithful realisation of the proposed concept that can be applied to the three evaluation scenarios explained in Sections 7.2, 7.3, and 7.4.

[547] See the left-hand side of Figure 7.1

[548] The involved co-workers and researchers are aware of the concepts and ideas proposed in Chapter 6

[549] The Eclipse Rich Client Platform (RCP) is a software package allowing developers to create Eclipse plug-ins or a Rich Client Application: cf. Billings et al. (2018), pp. 234–238; McAffer et al. (2010), pp. 15–27

[550] Figure 4.4 explains AHP in Section 4.2, Chapter 4

Although this dissertation only involves one person in the software development, the implementation followed the principles of agile development. More specifically, agile principles related to Scrum methodologies rather than to test-driven development, Kanban, or behaviour-driven design[551]. With Scrum, the large implementation projects is divided into easily manageable sprints, facilitates communication with other researchers, and the short sprints facilitate agilely gather feedback and change course. Rigorous and robust, test-driven or behaviour-driven development are too time-consuming for a single researcher developing a software prototype. Kanban works better for less stable development process where new change requests constantly come from stakeholders[552] by focusing on managing knowledge work balancing JIT (Just in Time) and work overload[553].

7.1.1 Design decisions for the prototypical implementation

The following design decisions are taken as a part of the decision-making process to define the software architecture of the InCLOUDer prototype. They stem from the evolution of the conceptual design of the cloud migration DS concept, explained in Chapter 6, so as to adopt technological solutions that deliver the requirements listed in Chapter 5. Figure 7.3 maps the requirements of the proposed concept to the design decisions taken to implement them with the InCLOUDer prototype.

In the realm of application systems engineering, the requirements analysis phase encompasses tasks to discover the needs and conditions that the prototype has to meet to fulfil the needs of the relevant stakeholders[555]. That is, organisations that could use the proposed concept to cloud migrate their application systems.

[551] Test-driven development (TDD), Kanban, and behaviour-driven design (BDD) are paradigms studied for their potential application to the implementation of the InCLOUDer prototype. Refactoring: cf. Haendler and Neumann (2019), p. 307, 310–312; Eberlein and Prado Leite (2002), p. 4; TDD: Karac and Turhan (2018), pp. 82–84; Beck (2003), pp. 13–15; Kanban: Anderson and Carmichael (2016), pp. 9–12; Kniberg (2011), pp. 17–19; BDD: Lübke and Lessen (2016), pp. 17–20; Smart (2014), pp. 12–27

[552] cf. Lei et al. (2017), pp. 62–65; Anderson (2010), pp. 11–16

[553] Kanban comes from the Just-in-Time or JIT delivery methods in lean manufacturing

[554] Own illustration mapping the requirements of the cloud migration DS concept, which Chapter 5 describes, mapped to the functional and non-functional design decisions taken for its prototypical implementation that Figure 7.4 sketches above

[555] The requirements analysis phase usually includes analysing, documenting, validating, and managing the software or application system requirements. The documentation of a set of actionable, measurable, and traceable requirements let analysts test and compare them to the identified business needs and opportunities: cf. Kuhrmann et al. (2017), pp. 30–39; Bourque et al. (1999), pp. 35–44; Sommerville and Kotonya (1998), pp. 12–22

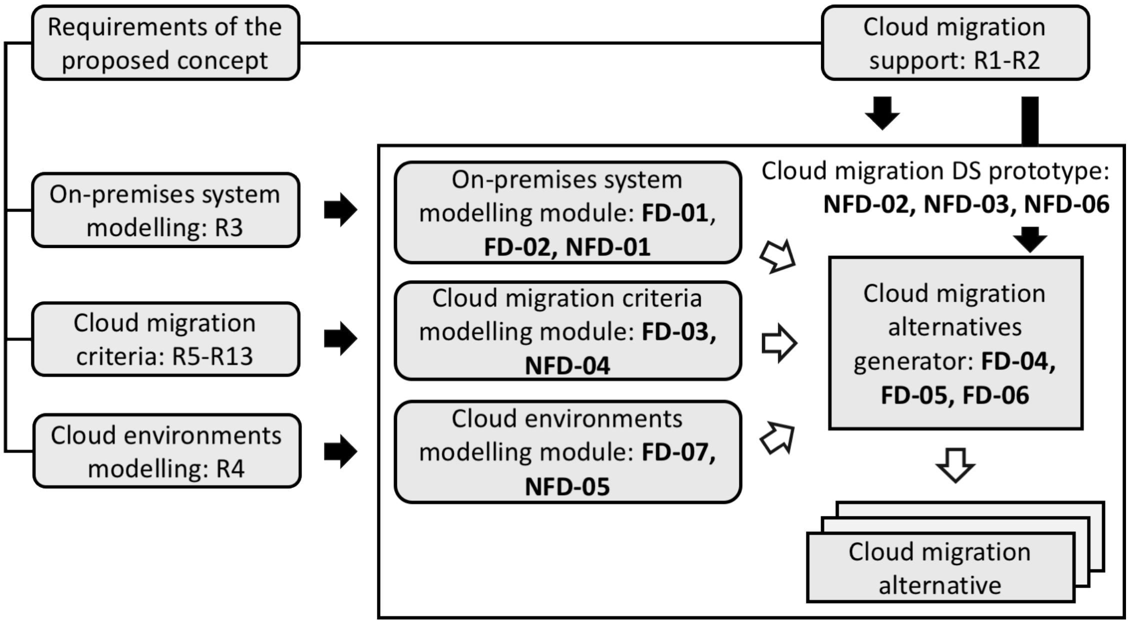

Figure 7.3: Requirements of the cloud migration DS concept mapped to the functional and non-functional design decisions taken for its prototypical implementation[554]

The elicitation, analysis, and recording of the design decisions result after agilely developing the proposed concept in line with cloud migration research[556]. As suggested by agile methodologies, change is fundamental to the research process of this dissertation. Accordingly, it is crucial to design and implement the prototype so that it quickly and easily accommodates changes due to novelties in the state of the art of cloud migration or findings by this evaluation. The agile implementation of the prototype delivers frequent, small, and re-factored releases to keep software quality high. Additionally, short development cycles facilitate pivoting to the right objectives if the implementation diverted from the research frame of reference of this dissertation or the requirements of the proposed concept in addition to facilitating altering the research course to bridge newly identified research gaps.

The design decisions are described without taking into account in which moment of the development process they arose. Explicitly defined design decisions document the decision-making process that gives birth to the software architecture of the prototype. They narrow down the requirements of the proposed concept so that the prototype becomes easier to understand, explain, maintain, and evolve. The

[556] The adaptation of requirements engineering to incorporate agile practices in software development is based on multiple empirical studies and the study of how successful software development teams work: cf. Lucassen et al. (2016), pp. 388–393; Cao and Ramesh (2008), p. 60; Cohn (2004), pp. 17–31

rest of this sub-section lists the design decisions taken for the implementation of the prototype; both from a functional (what the prototype has to do and why) and non-functional (how the prototype has to be) perspective.

Design decisions with functional implications

These design decisions restrict the freedom when implementing the functionalities of the prototype by clearly defining what it has to do and how. They describe the functionalities of the prototype based on its input, behaviour, and output. Design decisions *FD-01* and *FD-02* specify how to implement the *R3* requirement of the proposed concept, which stated that organisations should be able to model their application systems targeted for cloud migration[557].

FD-01: Modelling the target application system
Organisations can arguably benefit from using a sub-system to model their component-based architectures and the interfaces through which they interact. This is an input model used to generate potential target deployments to cloud environments and the local premises subject to the cloud migration criteria. The target application system models can also integrate source code models into their appropriated components if needed (see *FD-02*). In turn, this allows for creating metrics to use the source code details and dependencies to assess the achievement of the cloud migration criteria of an organisation based on them.
Related concept design requirement: *R3*

With relation to *FD-02*, organisations can use this automatic modelling based on the source code to facilitate their task but not at the expense of accuracy and realistic modelling. Therefore, organisations still require the approach and mechanisms to fine-tune these models according to their knowledge. The need for this sort of modelling stems from the need to use actual code to improve the quantitative evaluation of the metrics to assess the extent to which the cloud migrated solution delivers the desired cloud migration criteria.

[557] Section 5.1 explains the literature analysis related to on-premises application system modelling in Chapter 5

FD-02: Modelling the target application system using source code
Organisations sometimes need to model complex target application systems they intend to migrate to cloud environments to consider all their particularities. That is why, they need a way to alleviate the burden of modelling their application systems with the use of tools that head start the modelling of the component-based architecture according to the source code they have already deployed to their premises[558]. Nevertheless, modelling and its fine-tuning is still possible without source code in a manual manner.
Related concept design requirement: *R3*

Design decision *FD-03* details *R5* and *R6* to offer a method to let organisations define their priorities when cloud migrating their application systems[559]. In addition, *FD-04*, *FD-05*, and *FD-06* also relate to *R5* and *R6*. These design decisions need to abide by the concept requirements for business-related, technical, provider-related, and security and privacy cloud migration criteria; that is requirements *R7* to *R13*. They lead to functional solutions that can generate cloud migration alternatives, allow organisations to model their cloud migration criteria and how they depend on one another, and perform the automatic or user-assisted weighing of cloud migration alternatives. Design decision *FD-03*, *FD-04*, and *FD-05* also relate to the requirements for cloud migration support in cloud and inter-cloud architectures: *R1* and *R2*. The prototype has to consider cloud and inter-cloud architectures when modelling the cloud migration criteria as well as when generating the cloud migration alternatives and automatically measuring them for the different criteria.

[558] Some mechanisms to extract models follow the initiative of the Architecture-Driven Modernisation (ADM) within the Object Management Group (OMG) to employ meta-models. As an example, reverse-engineered code models corresponding to Knowledge Discovery Meta-Model (KDM) of the Architecture-Driven Modernisation have got the potential of delivering this particular design decision

[559] Requirements *R5* and *R6* are gathered via the literature analysis of the cloud migration criteria as Section 5.1 explains in Chapter 5

FD-03: Modelling cloud migration criteria and their interdependencies
Cloud migration criteria vary from one organisation wanting to migrate its application system to cloud environments to the next one. Even among the same organisations different views might exist. Therefore, organisations require a flexible manner to easily model their criteria and priorities when moving to cloud infrastructures. Organisations do not want to learn a new methodology but to effortlessly digitalise their thoughts so that the prototype can interpret them and help them make decisions. The cloud migration criteria are complex and contain multiple sub-criteria that can comprise additional sub-criteria as well. They conform a taxonomy with criteria and sub-criteria that relate and depend on each other. The developed sub-system to model the cloud migration criteria needs to address these complexities and particularities. As the definition of cloud migration criteria is quite cumbersome, this design decision arises to foster the re-usability of the criteria across organisation. Either as unique elements or as templates that organisations share or apply throughout their entire organisation.
Related concept design requirements: *R1*, *R2*, and *R5* — *R13*

FD-04: Generating cloud migration alternatives
The prototype arguably needs to generate cloud migration alternatives as the output model based on the input models. The alternatives, which include the partial scenario whereby the premises local to the organisation are also used, represent the solution space that the prototype weighs —with or without human intervention— in order to find out the most suitable ones and recommend those to the organisation. The prototype has to arguably use these alternatives to improve the cloud environment configuration selection and the architectural deployment.
Related concept design requirements: *R1*, *R2*, and *R5* — *R13*

FD-05: Automatic weighing cloud migration alternatives
Organisations could benefit from reusing cloud migration criteria and sub-criteria without manual effort if the particular criterion metrics and criteria template previously existed. Likewise, they can profit from the capabilities of the prototype to, to the largest extent possible, automatically weigh each cloud migration alternative. Alternatives are weighed with respect to one another and according to the degree to which they help organisations achieve their cloud migration roadmap.
Related concept design requirements: R1, *R2*, and *R5 — R13*

FD-06: User-assisted weighing cloud migration alternatives
In those cases whereby the prototype cannot automatically weigh the cloud migration alternatives, organisations could have the option to intervene and weigh the alternatives for the particular cloud migration criterion or sub-criterion that the prototype cannot automatically weigh. This might usually be because of its subjective nature, which would make the prototype rely on assisting organisations in deciding whether the cloud migration alternative fulfils a particular criterion. The prototype could facilitate this assessment process and seamlessly integrate it with the ideal automatic weighing so that the user does not suffer from a steep learning curve.
Related concept design requirements: *R5 — R13*

The *FD-07* design decision provides additional details related to the needs for modelling cloud environments as stated by *R4*. Additionally, *FD-07* goes hand in hand with requirements *R1* and *R2* that require modelling cloud environments according to the particular cloud and inter-cloud architectures. Some prototype design decision (such as *FD-07*) emerge from multiple requirements as they relate to each other and only make sense in conjunction.

FD-07: Modelling cloud environment configurations
Experts with enough knowledge of a cloud environment could model the cloud environment configurations it offers. In order to generate cloud migration alternatives, the proposed concept combines the cloud environment configuration models with the models of the target application system and the cloud migration criteria of organisations. Several concepts and tools exist in the state of the art to model cloud environments and their variability, such as feature models. Existing tools —which stem from contributions by related research and industrial projects as open source solutions, services, black-boxed libraries, or other artifacts— will change to be either updated or deprecated. In such a volatile environment, the architecture prototype can facilitate interchanging these cloud environments modelling concepts and tools by having an encapsulated and modularised architecture that accommodates the evolution of the prototype. This dissertation integrated an existing tool to model cloud environments to proof the viability of the proposed concept.
Related concept design requirements: *R4*

Design decisions with non-functional implications

Quality design decisions impose constraints on the design and implementation. They include performance, trust, reliability, or security design decisions. Non-functional design decisions define how the prototype has to be. Design decision NFD-01 —extendible target application system modelling via source code— allows for modelling the target application system prior to migration based on its source code, which closely emerges from the *R3* requirement of the proposed concept.

NFD-01: Extendible target application system modelling via source code
The target application system modelling architecture built in an extendible manner enables adding new Knowledge Discovery Meta models (KDMs) as they appear. That way, the prototype will be able to easily include new object-oriented languages not modelled by the time of publishing this dissertation.
Related concept design requirements: *R3*

NFD-02: Data gathering to automate the cloud migration alternatives weighing
The prototype can rely either on the models and metrics defining how to quantify the cloud migration alternatives according to each cloud migration criterion. The prototype can automatically quantify cloud migration criteria if metrics exist to quantify them using source code models extracted from the code base, the target application system, its components' properties, and the cloud environment configurations.
Related concept design requirements: *R1-R2*

Design decisions NFD-02, NFD-03, and NFD-04 let the prototype work with the cloud migration criteria affecting for the generated cloud migration alternatives. They stem from the *R5* and *R6* concept requirements. NFD-02 allows for efficient data gathering to automate the cloud migration alternatives weighing. NFD-03 and NFD-04 intend to facilitate extending automatic weighing mechanisms and cloud migration criteria models, respectively. In addition, NFD-02, NFD-03, and NFD-04 abstract from the business-related cloud migration criteria related to *R7*, the technical cloud migration criteria (see *R8* & *R9*), the provider-related cloud migration criteria (see *R10* & *R11*), and the security and privacy cloud migration criteria, as described by *R12* and *R13*.

NFD-03: Extendible automatic weighing mechanisms
The prototype use weighing mechanisms based on metrics to automate the process of assessing the value of cloud migration criteria. The prototype should let organisations easily extend these mechanisms at the same pace with which they define new criteria and sub-criteria. The weighing mechanisms assess the value of the criteria for the cloud migration alternatives, according to the three input models, on a quantitative manner similar to mathematical formulas or simulation-based metrics.
Related concept design requirements: *R1-R2*

NFD-04: Extendible cloud migration criteria models

The re-usable cloud migration criteria models have to accommodate the needs of organisations to extend those criteria taxonomies and to customise them to their particular domain and application system. Organisations require intuitive mechanisms to extend these cloud migration criteria so that they are willing to adopt the criteria modelling approach due to its flat learning curve and good UI design.

Related concept design requirements: *R5-R13*

NFD-05: Exchangeable cloud environment configurations modelling modules

Cloud environment configuration modelling sub-systems describe exemplary cloud environment configurations but not all. Multiple approaches exist to model cloud environments and usually researchers describe cloud offerings with their prototypical implementations (rather than cloud providers themselves). In the future, a new approach might appear and even become so standard that even cloud providers would be enticed to use it to describe their own offerings. Therefore, the prototype incorporates encapsulation mechanisms and might even adopt component- or framework-based middlewares to allow for easily exchanging one cloud environment configurations modelling module with another one.

Related concept design requirement: *R4*

Design decisions NFD-05 and NFD-06 are usual software engineering practices applied to the proposed concept. These are design decisions taken to provide modules to model extendible cloud migration criteria and cloud environment configurations. At the same time these decisions intend to lower the coupling between the software components forming the prototype. NFD-05 stems from the *R4* requirement for cloud environments modelling while NFD-06 emerges from the requirements for cloud migration support systems.

NFD-06: Low component-to-component coupling
The prototype integrates loosely-coupled components that interact through their well-defined interfaces following the information hiding principle. The prototype should deliver designs with lower degrees of interdependence between software modules so that the components do no present strong relationships between them. Lower levels of cohesion could facilitate exchanging the components delivering each of the functional requirements of this dissertation with new and better components. Additionally, the prototype should, in this manner, offer the possibility to integrate new components for new functional requirements as well.
Related concept design requirement: *R1-R2*

7.1.2 Conceptual architecture of the prototypical implementation

The InCLOUDer prototype stems from applying this dissertation's theoretical study to real scenarios according to the conceptual architecture shown in Figure 7.4. It shows where the knowledge and logic of the cloud migration concept resides, in addition to what the input and output information, models, and knowledge are to be expected.

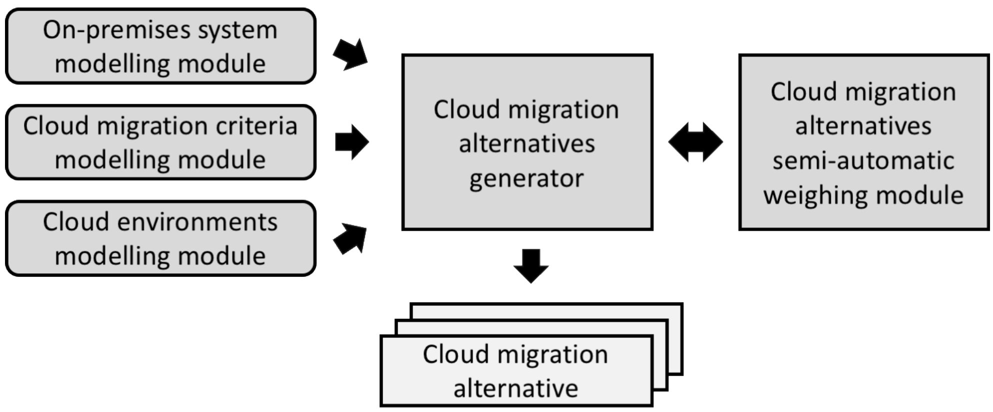

Figure 7.4: Architecture of the InCLOUDer prototype[560]

The output models of the three modelling modules embedded in the prototype, on the left-hand side Figure 7.4, are the input for the the cloud migration alternatives generator in the middle. The generator creates the different options to

[560]Own illustration

cloud migrate the target system with the aim of suggesting the highest-ranked cloud migration alternative according to the organisation's cloud migration criteria computed by the cloud migration alternatives semi-automatic weighing module. This module assesses the value of a cloud migration criterion for the generated cloud migration alternatives according to pre-configured metrics. In addition, it allows organisations to manually fine-tune the automatically ranked cloud migration alternatives according to their view on particular cloud migration criteria or to manually assess the alternatives for a criterion in absence of associated metrics. The cloud migration alternatives generator supports additional functionalities to, for example, rule out those alternatives not conforming with the application system model constraints. This way, the ranked cloud migration alternatives contain valid migration alternatives to deploy the application system to cloud environments and the local premises that represent.

The InCLOUDer prototype, in its 12.1.0 version, is distributed under Apache License 2.0 as Figure 7.7 shows as part of the *About information* containing contact information to the author. Its start page shows, in Figure 7.5, the implementation of the workflow explained in Chapter 6[561]. The screen shot shows the RCP-based prototype[562] together with the approaches to model the input of the proposed concept as well as the cloud-migrated solution. Figure 7.5 depicts how the organisation can access all options in the BI Cloud Migration Migration Project because the prototype checked all the pre-conditions needed to access each functionalityin compliance to the workflows designed for the proposed concept.

Based on Eclipse RCP, the prototype runs as a stand-alone application system with multiple software components and OSGi bundles. EMF is used to implement the modelling modules that support the cloud migration decision, with Ecore as the core meta-model used to model the input and output of the prototype[564].

[561] Figures 6.3 and 6.4

[562] cf. Billings et al. (2018), pp. 234–238; McAffer et al. (2010), pp. 15–27; as Section 4.3 and Figure 7.9 show

[563] Own illustration. Screen shot of the prototypical implementation of the proposed concept.

[564] EMF is a modelling framework that uses a structured data model to generate code for building tools and other application systems cf. Mengerink et al. (2016), pp. 2–4; Steinberg et al. (2008), pp. 11–32. The models created comply with the Ecore meta-model while using front-end tools for Graphical Modelling Framework (GMF). EMF offers tools and runtime support to produce Java classes to view, edit, and run models when the developer specifies a model in the XML Metadata Interchange (XMI). That is, an Object Management Group (OMG) standard working

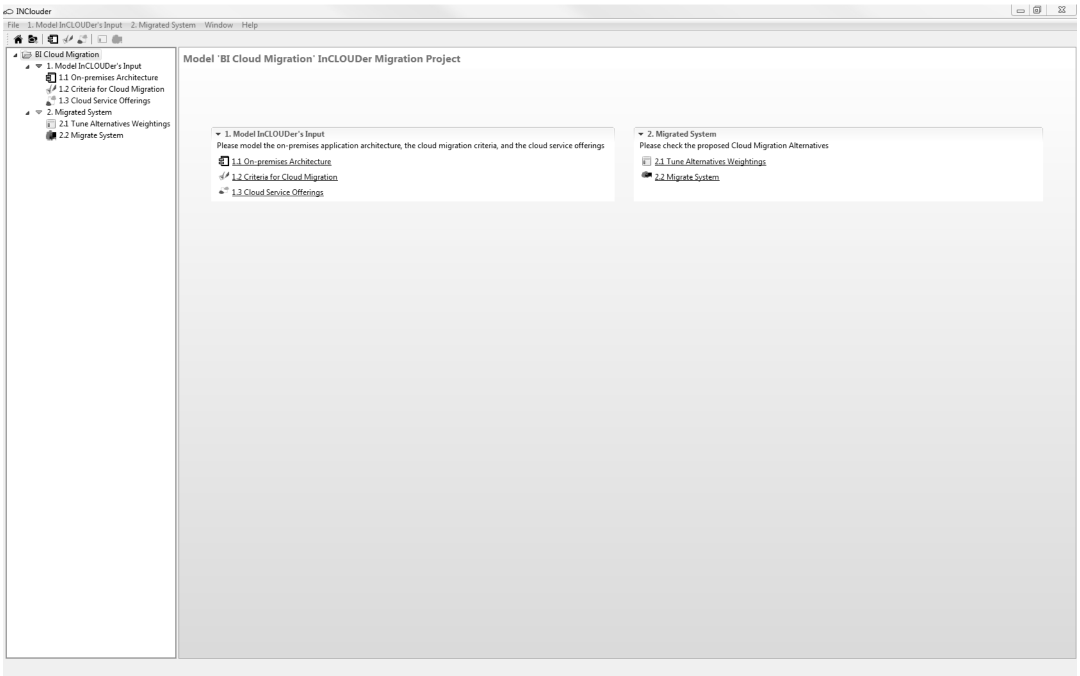

Figure 7.5: Start page of the prototypical implementation of the proposed concept[563]

In addition, the prototype uses the EMF Platform UI to easily integrate the out-of-the box runtime environment, user interface (UI), help components, and other basic building blocks. The cloud environments modelling module, which is explained in detail in the next sub-section, allows for describing cloud environment configurations as the example in Figure 7.6 shows. In particular, the figure shows the configuration options offered by the Rackspace cloud environment. That is, the different VMs Rackspace offers with varying processing power, memory capacity, disk space, operating system, and off-the-shelf software applications.

The InCLOUDer prototype incorporates the proof-of-concept and proof-of-value implementations of the proposed concept to test the idea presented in this dissertation[566]. The evaluation takes a design-oriented stance to iteratively and incrementally implement the prototype according to the scientific study of the state of the art of cloud migration and the experience gathered in cloud migration projects[567].

with Extensible Markup Language (XML) to exchange metadata information

565 Own illustration

566 This dissertation holds on to the practice-oriented nature of the design-oriented information systems research as Section 4.7 in Chapter 4 explains: cf. Ciriello et al. (2018), pp. 566–568; Dolata and Schwabe (2016), p. 67

567 Nunamaker and Briggs noted that projects within the realm of design-oriented information systems

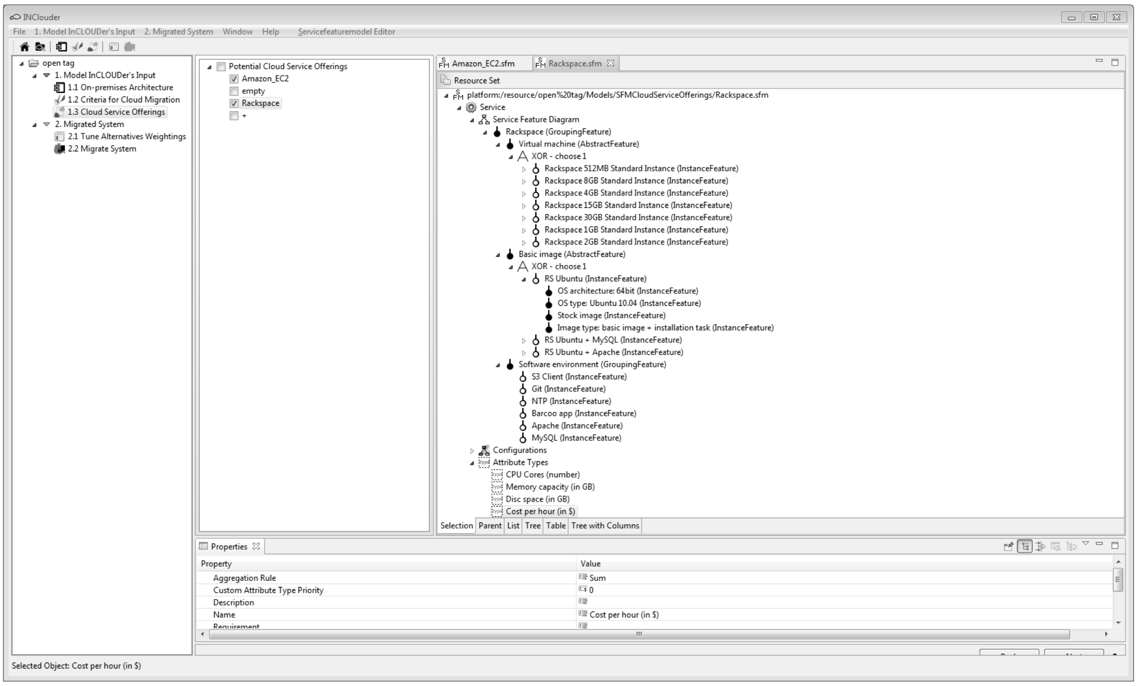

Figure 7.6: Cloud environments modelling module prototypical implementation to model the cloud environment configurations[565]

The first software implementation iterations delivered the proof-of-concept prototype, which paves the way towards executing the proof-of-value phase. This phase envisions having potential users of the concept test the prototype. In the case of this dissertation, potential users include domain experts of organisations who want to migrate their application systems to cloud-based architectures. The proof of concept aims at testing the idea proposed with the cloud migration DS concept with the use of the implemented prototype. The proof-of-value stage answers the question of how the new solution creates value and how understanding the value creation process contributes to the existing theory. The next step would involve proof-of-use studies to assess whether the proposed concept and prototype could attract growing self-sustaining communities to practice with them, which goes beyond the scope of this dissertation and, hence, belongs to the future work of this dissertation[568].

can belong to three stages: the proof-of-concept, proof-of-value, and proof-of-use stages: cf. Giboney et al. (2019), pp. 11–13; Nunamaker and Briggs (2011), pp. 12–14. The authors for moving research beyond proof-of-concept implementations to reach the proof-of-use state in order to trigger richer scientific contributions. Proof-of-concept prototypes are used to proof technical feasibility. Phenomena and outcomes are studied with the proof-of-value prototype. Finally, proof-of-use prototypes help achieve holistic understanding of their environment and context of use at the cognitive, emotional, social, physical, economic, and political levels

568 cf. Giboney et al. (2019), pp. 11–13; Nunamaker and Briggs (2011), pp. 15–16

569 Own illustration. Screen shot of the prototypical implementation of the proposed cloud migration

Figure 7.7: About description of the prototypical implementation of the proposed cloud migration DS concept: InCLOUDer 12.1.0[569]

As the proof-of-use phase remains for the future work, the design of the prototype and the scenario-based evaluation aims at enabling future releases of the updated prototype based on the outcomes of the proof-of-concept and proof-of-value research phases that could help in the proof of use. The proof-of-concept phase assesses the technical feasibility of the prototype while its efficacy is tested during the proof-of-value phase. The holistic understanding of the context related to the migration of application systems to cloud environments could ensue in the development of an entrepreneurial venture in the highly volatile business environment of cloud migration. Such a technology-oriented business undertaking has got a high growth potential as cloud adoption grows and the research idea turns into a business idea developed in accordance with that increasing adoption. A potential proof-of-use phase would have the potential to take into consideration the social, political, economic dimensions of the context of cloud migration together with the fundamental cognitive, emotional, and physical context surrounding the operation of the upcoming cloud migration systems.

Based on this architecture, the next section goes into the details of the modelling modules that the architecture integrates. These modules include the cloud migration alternatives generator, the modelling modules providing the generator with input models describing the reality of the treated cloud migration projects, and the semi-automatic weighing module.

DS concept

7.1.3 Prototype modules implementation

The following provides more details on the software modules conforming the architecture of the InCLOUDer prototype shown in Figure 7.4.

On-premises application system modelling module

Modisco was used for this module in conjunction with its multiple tools for modernising application systems, see Figure 7.8[570] shows. The framework follows model-driven approaches to support mature and flexible software evolution activities.

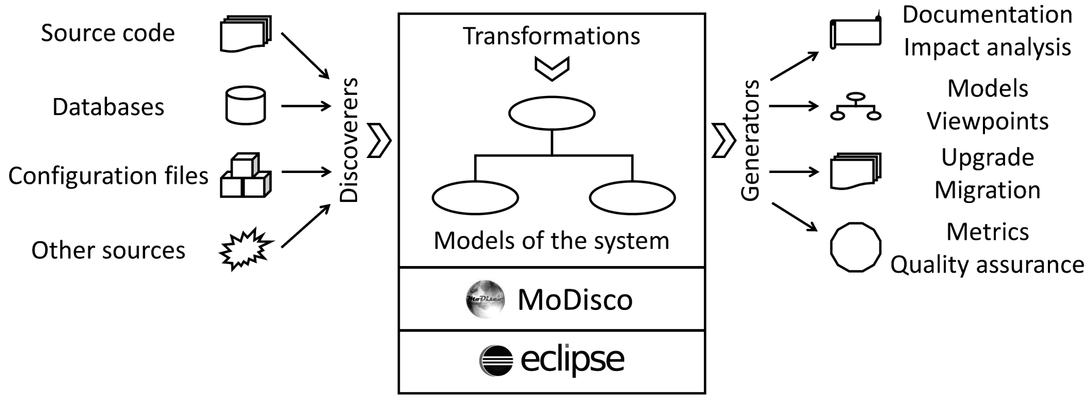

Figure 7.8: MoDisco model-driven extensible framework candidate for the on-premises application system modelling[571]

The study of the MoDisco framework and its implementation of the KDM and SMM model specifications[572] motivated adopting it to implement the on-premises application system modelling module of InCLOUDer. The prototype uses KDM to extract the structure of existing software —if the code is accessible— and SMM to specify metrics to quantify the scores of a cloud migration alternative for a particular cloud migration criterion. The KDM and SMM meta models are optional and the prototype can operate without them if the organisation in charge of the migration could not work with the actual legacy code. This happens at the expense of the level of accuracy and automation for criteria metrics based on these models. In addition to providing KDM and SMM out of the box, MoDisco is an Eclipse RCP package relatively easy to learn and to use that uses the same set of tools that

[570] cf. Bruneliere (2018), pp. 42–55; Bruneliere et al. (2014), p. 1012; Madiot (2010), pp. 12–18
[571] Bruneliere et al. (2014), p. 1025
[572] Section 4.3 in Chapter 4 introduced the MoDisco KDM (Knowledge Discovery Meta model) and SMM (Structured Metrics Metamodel) models

developers use to create Eclipse plug-ins, Rich Client Application, or Rich Ajax Applications. RCP stand-alone application systems, like the InCLOUDer prototype, use the Eclipse platform technologies and the software components shown in Figure 7.9.

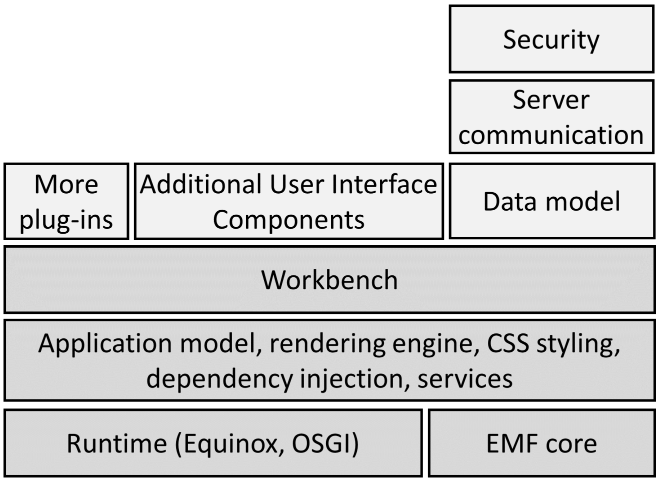

Figure 7.9: Eclipse RCP Application components by Vogel in 2016[573]

The prototype uses the Eclipse runtime software components (Equinox, OSGi), the EMF core, and the UI Core (JFace and the Standard Widget Toolkit or SWT)[574]. OSGi specifies the modular approach to component-based development of Java application systems[575] so that the runtime components provide the APIs and framework to run modular Eclipse-based application systems. Eclipse uses the SWT UI component library as well as JFace, which offers additional APIs such as viewer classes to implement the model-view-controller pattern in addition to file buffers and text editors. As for the workbench, it displays all other UI components, including views, editors, perspectives, and wizards. Finally, EMF allows for modelling a data model and for using it at runtime.

The on-premises application system modelling module lets organisations model the target application system for cloud migration and the software components that compose it. Such a deployment description of the application system is one input model that the prototype uses to analyse how suitable it is to cloud migrate it. Organisations can model their deployments as a composition of different

[573] Vogel (2015), pp. 411–436
[574] cf. Vogel (2015), pp. 411–436
[575] OSGi services are dynamic software components defined with the OSGi programming model and Equinox implements its specification and works as the platform's runtime
[576] Juan-Verdejo (2014), p. 5

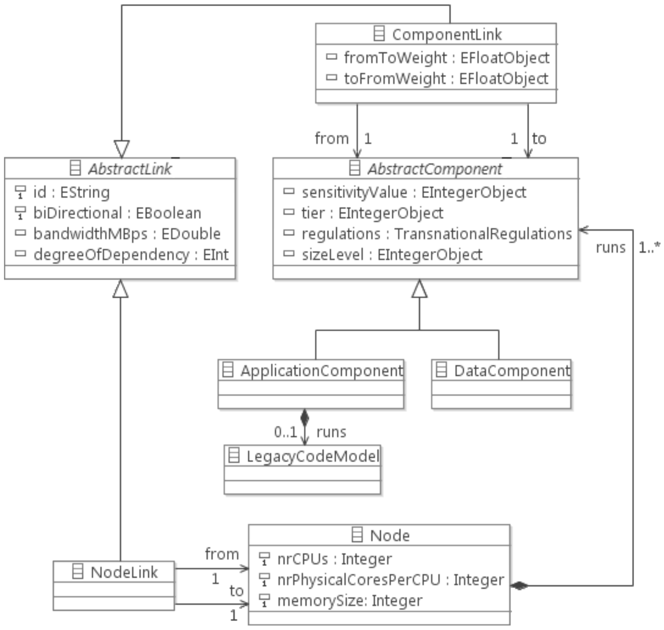

Figure 7.10: On-premises application system modelling module meta model[576]

data and processing software components running in different computing nodes. The application system modelling module relates to the deployment diagrams in the Unified Modelling Language (UML) that model the physical deployment of artifacts on nodes[577]. Figure 7.10 shows the meta model used to let organisations specify their application system's requirements, constraints, and architecture. Organisations can thereby define their application system's data and computing components, specify in which nodes they run, and how they connect to one another.

Figure 7.11 shows the editor to model the on-premises architecture of the application system to describe the processing and data software components, how they are connected, and the sensitivity of the data that they process or store. These details are specified with the components' properties view and, in the case of the data sensitivity, they alter each component in Figure 7.11 with a the key at their bottom right corner indicating either the low, medium, high, and secret sensitivity level of those data-related operations. This information is used to restrict the solution space of potential cloud migration alternatives and to accordingly penalise alternatives that cloud migrate sensitive data without the necessary precautions. This ranking is done by taking into account the sensitivity levels in combination

[577] cf. Platt and Thompson (2019), pp. 1452–1456; OMG (2015), pp. 21–35, URL see Bibliography

[578] Own illustration. Screen shot of the prototypical implementation of the proposed cloud migration DS concept

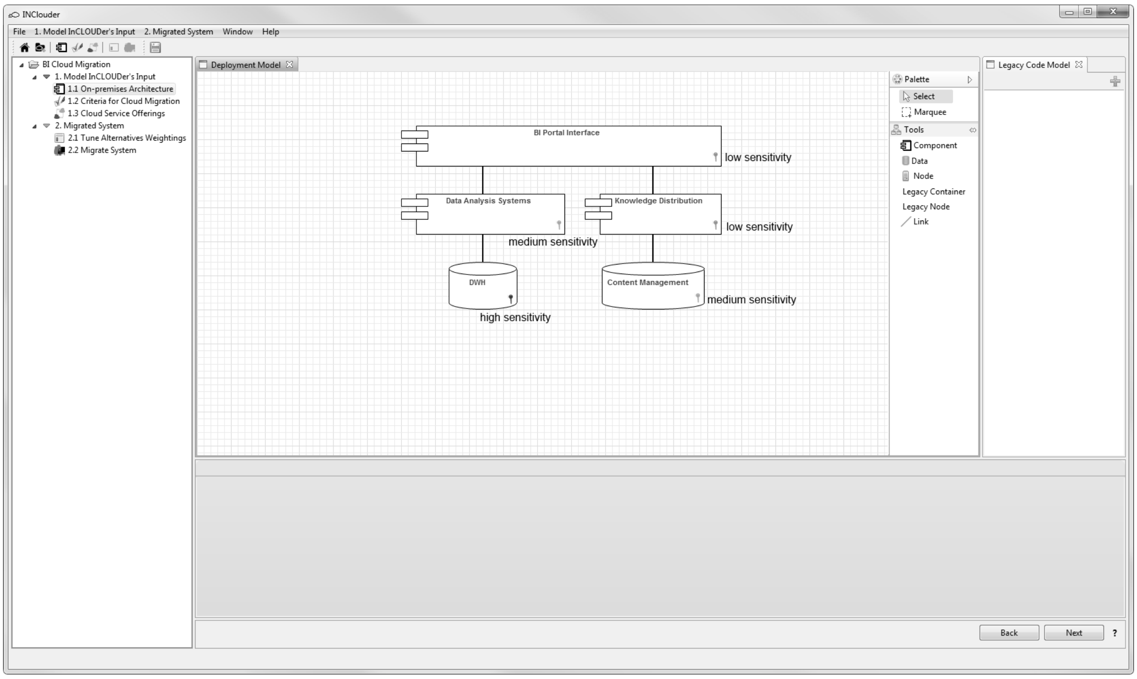

Figure 7.11: On-premises application system architecture modelling[578]

with the deployment design, the requirements of the application system, and its constraints. The goal is to respect the constraints and increase the fulfilment of the requirements of the target application system and of the organisation migrating it.

Cloud migration criteria modelling module

The prototype offers an initial set of cloud migration criteria that organisations may tailor to their priorities for the cloud migration project they undertake. Organisations can also adapt the hierarchy of criteria and how to weigh each criterion against the rest of them, via pairwise comparisons that depend on the cloud migration problem at hand. A criterion can broken down into multiple sub-criteria that, in combination according to pairwise-compared sub-criteria, quantify it. The prototype help rank each cloud migration alternative for each specific quantified criterion according to the tailored and extensible cloud migration criteria taxonomy defined by the organisation[579]. Criteria are quantified either via criterion metrics proposed by the prototype, metrics specific to the organisation, or human intervention[580]. The design of the cloud migration criteria modelling module is informed by previous

[579] The use of the cloud migration criteria taxonomy was inspired by AHP and is informed by the Service Measurement Index framework cf. Ortiz et al. (2018), pp. 555–557; Siegel and Perdue (2012), pp. 411–415; and SMICloud Garg et al. (2011), pp. 212–214

[580] Section 6.3 explains, based on a published paper, a particular example of a cloud migration criteria model based on the explained extensible criteria taxonomy: cf. Juan-Verdejo et al. (2014b), pp. 468–473

research works that use the Service Measurement Index framework, SMICloud, and AHP[581]. The prototype uses the Analytical Hierarchy Process to address the multi-criteria decision-making in order to make the comparison of multiple criteria measured in different units feasible. Section 6.2 provides, in Chapter 6, more details that motivate and explain the use of AHP. The cloud migration criteria modelling module is part of the mathematical decision-making technique integrated in the prototype in order to rank cloud-migrated versions of the target application system. Organisations are supported in finding the relative importance of each criteria in comparison with one another and in isolation. In the end, as Figure 7.14 shows, the prototype uses these weighed criteria (and sub-criteria) to semi-automatically weigh the alternatives in order to find the highest-ranked cloud migration alternative.

Cloud environments modelling module

Cloud environments can be modelled, as a part of the input needed to carry out the workflows in Chapter 6, with the module shown in Figure 7.12.

The figure shows a list of the potential cloud service offerings, Amazon and Rackspace, to which the organisation can migrate its application system. On the right hand-side of the figure, the editor[582] uses feature models to allow for modelling the selected Amazon EC2 cloud offering and its possible configurations. This modelling module offers several views (selection, parent, list, tree, table, and tree with columns) to model different cloud environment configurations in combination with the view at the bottom. The latter allows for specifying different properties of each feature and attribute of the cloud environment feature model.

[581] Section 4.2 and Figure 4.4 explain in Chapter 4 the related research works and Chapter 6 explains AHP in the context of the proposed cloud migration DS concept including the cloud service measurement index cf. Ortiz et al. (2018), pp. 555–557; Siegel and Perdue (2012), pp. 411–415; the research work by CLOUDS Laboratory of the University of Melbourne, SMICloud Garg et al. (2011), pp. 212–214; and AHP Saaty (1977), pp. 245–252

[582] The integrated cloud environments modelling module stems from previous research works cf. Wittern et al. (2012), pp. 138–140; Solomon (2017), pp. 35–45

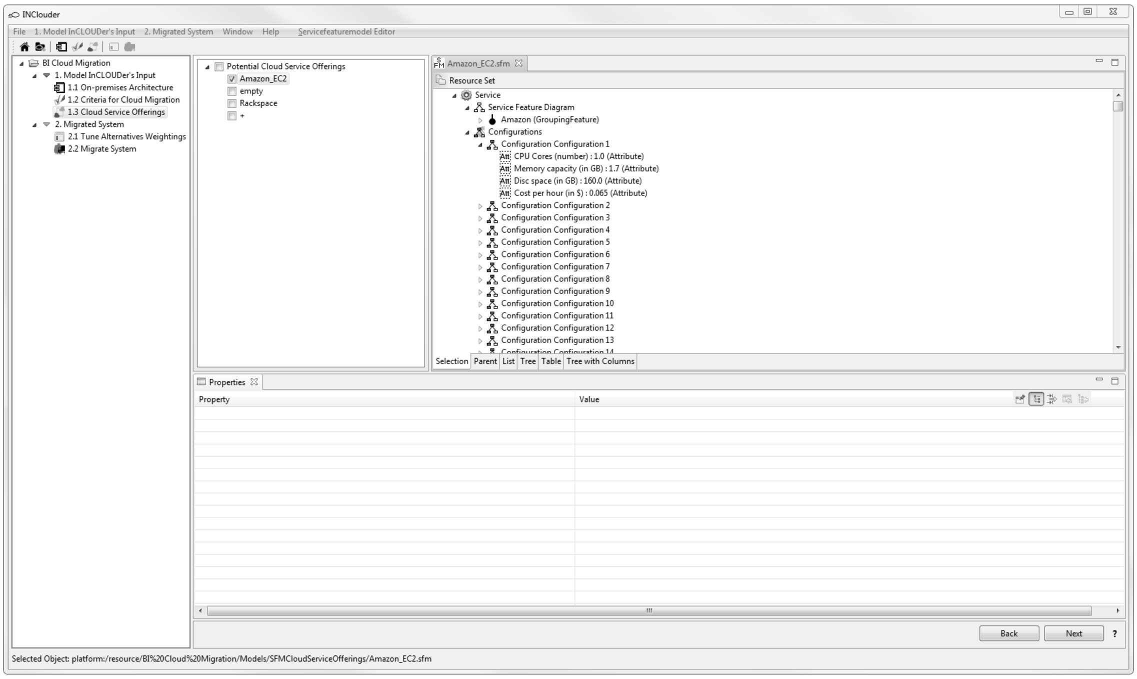

Figure 7.12: Cloud environments modelling module integrated in the prototype[583]

Cloud migration alternatives generator

The migration alternatives generator takes into account the three models described above and computes valid cloud migration alternatives that follow the meta model specified in Figure 7.13. Firstly, the prototype rules out the cloud migration alternatives not conforming to the constraints described in the application system model. The alternatives generator takes into account the component-based architecture of the application system and its requirements to find out the valid deployment alternatives that re-scatter software and data components to the selected cloud environments and the local premises to cope with privacy-related requirements. The prototype can then quantify the remaining cloud migration alternatives for each cloud migration criteria either according to criterion metrics or with user intervention.

On its left-hand side, Figure 7.13 shows the cloud migration criteria with Dimension and its different DimensionTypes. On the right-hand side, the cloud offering class (CloudOffering) describes the different cloud environment configurations across the entire cloud stack (see Section 6.6, Chapter 6, for more details). PropertyType is

[583]Own illustration. Screen shot of the prototypical implementation of the proposed cloud migration DS concept
[584]Juan-Verdejo (2014), p. 5

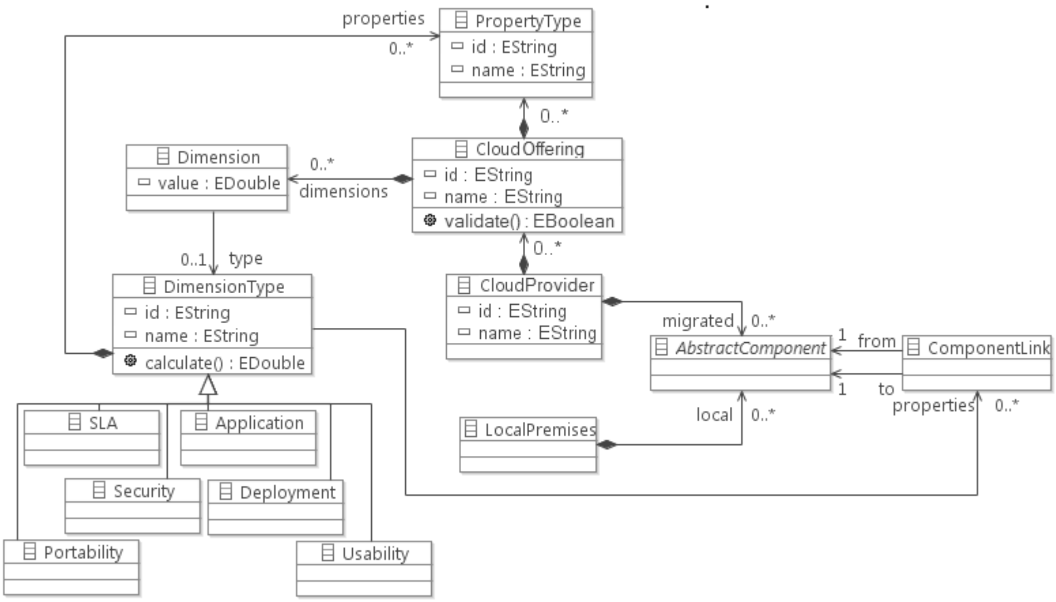

Figure 7.13: Cloud-enabled application system meta model[584]

linked to DimensionType by an association to let the prototype use information about the different cloud migration criteria in conjunction with the different properties of each cloud environment configuration to rank the cloud migration alternatives. Subsequently, the prototype quantifies, with or without human intervention, the set of cloud migration alternatives according to either the cloud migration criteria specified by organisations or one of the out-of-the-box criteria sets offered by the prototype.

Cloud migration alternatives semi-automatic weighing module

Provided that the organisation had already specified a particular criterion with a corresponding metric, this module can automatically calculate the score of each cloud migration alternative with respect to that criterion[585]. Once the hierarchy of cloud migration criteria has been built, based on the sub-criteria and criteria shown on the bottom and centre of Figure 7.14, the prototype prioritises them. Next, the weighing module calculates the eigen vector so as to easily work with all criteria weights at the same time[586]. Further, the prototype quantifies each criteria for all cloud migration alternatives and uses this calculation to compute the

[585] Section 6.3, Chapter 6, explains the semi-automatic alternatives weighing via an example in the context of the third step followed by the proposed concept as Section 6.2 describes

[586] Section 6.2 provides more details related to the steps the prototype follows for the semi-automatic weighing of the cloud migration alternatives

global ranking of all alternatives. For criteria for which metrics have been defined, the automatic weighing step quantifies the generated migration alternatives[587]. Were those metrics to be absent, the decision maker should manually determine, with the help of the prototype, how good the alternatives perform in relation to a particular cloud migration criterion. Additionally, human intervention can still fine-tune the prioritisation of the cloud migration criteria or adjust (or change) the automatic weighing given by the prototype to a particular alternative for a criterion or sub-criterion.

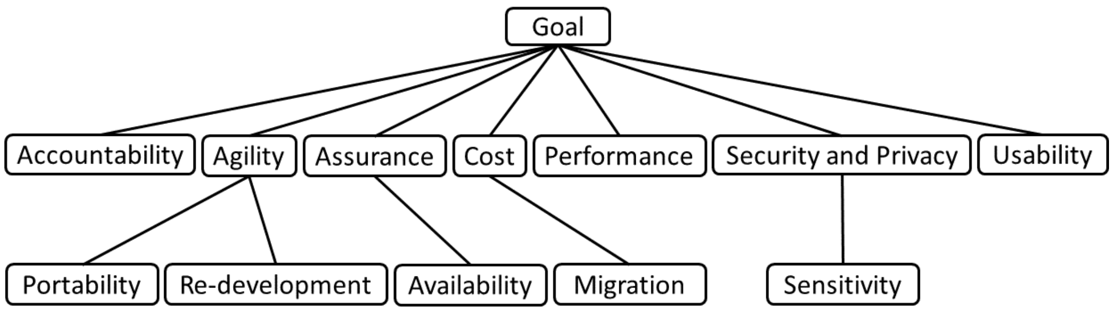

Figure 7.14: Criteria and sub-criteria semi-automatically weighed[588]

The prototype uses AHP to implement the multi-criteria decision-making process to pick the best-fitting cloud migration alternative according to the cloud migration problem at hand. The prototype relies on the information mentioned above to build up the cloud migration alternatives and criteria, compute the local and relative weights per criterion to be able to provide a ranking of cloud migration alternatives in descending order. The prototype uses values from one to nine (as AHP does) to assign relative importance to a cloud migration criterion over the others[589]. The prototype uses the module described here to compare the cloud migration criteria and normalise them as they use differing dimensional units: boolean values, numeric, sets of values, or ranges of values.

Finally, the prototype aggregates all these relative values to conform the overall weight of a cloud migration alternative in comparison with the others so as to find a global ranking. Relative ranking values of the different criteria are computed according to the Key Performance Indicators that organisations specify as a

[587]See Section 7.1, Chapter 6, for more details on the automatic weighing of cloud migration criteria and alternatives

[588]Own illustration

[589]cf. Ortiz et al. (2018), pp. 555–557; Siegel and Perdue (2012), pp. 411–415; Garg et al. (2011), pp. 214–216; Saaty (2005), pp. 14–32; Zeleny and Cochrane (1973), pp. 12–23

taxonomy of attributes and sub-attributes. The ranking is a complex task that depends on many factors as it is a decision-making problem with multiple criteria. Consequently, traditional weighed sum-based methods do not suit the complex structure of attributes conforming a hierarchy, which motivates the adoption of AHP. AHP helps when having such a complex structure of cloud migration criteria. Especially, because these aspects and factors are both objective and subjective and at the same time quantitative and qualitative.

7.2 Scenario 1: cloud migration to PaaS offerings

PaaS offerings offer higher levels of abstraction and therefore more advanced functionalities on top of IaaS offerings. This usually speeds up the development of application systems that compose the modules and functionalities provided by the PaaS provider. This evaluation scenario is selected due to the research to be done at this level of abstraction in terms of standards, portability, and interoperability. PaaS presents some particularities that have to be taken into account in the cloud migration to this deployment model. Additionally, most of the cloud migration research focuses on migration to IaaS working at the level of VMs, servers, storage, load balancers, and network. PaaS is a more abstract computational model that has its consumers work with execution runtime environments, databases, web servers, and different development tools. It allows for less freedom in the implementation with more development speed and tools for testing, deployment, and management in exchange[590].

7.2.1 Evaluation scenario setup

The first evaluation scenario (see the research frame of reference[591]) focuses on evaluating the cloud migration concept and how it assists the cloud migration of different application systems to PaaS offerings. This evaluation scenario involves organisations, and SMEs in particular, that want to cloud migrate their application

[590] For more details of the particularities of the migration to PaaS offerings see Sections 6.6 and 6.7

[591] The cloud migration to PaaS offerings, as Figure 7.1 shows on its right-hand side, is the first element of the research frame of reference. It focuses on cloud-enabling different application systems provided by SMEs to the selected PaaS offering and how the proposed concept assists organisations in this cloud migration

systems to a PaaS offering. The PaaS offerings are provided by different cloud offerings —Flexiant, Amazon web services, CloudBees, or Google cloud. This evaluation scenario leverages a research project for the convenience it offers to get appropriate respondents[592].

Motivation

The research shown in Chapter 6 has analysed the challenges and opportunities of cloud migrating application systems to the one of the three main cloud service models. In addition, Sections 6.6 and 6.7 explain the research performed at the PaaS level due to the less maturity of the research at this level; especially on topics such as interoperability, portability, and standards. Motivated by the research on cloud migration to the PaaS service model, the first evaluation scenario is selected to assess the support provided by the concept to the particular case of cloud migration to PaaS environments. The particular cloud migration criteria listed in Chapter 6 are specified according to the cloud migration projects in order to assess the effectiveness of the concept for PaaS migration. The focus of this evaluation scenario lies on the criteria of particular relevance for PaaS and especially for those related to countering vendor lock-in with the selection of cloud migration alternatives that support interoperability, portability, and PaaS standards. In addition, this evaluation scenario is included in the dissertation because the potential exploitation of the concept can be assessed as well.

The conclusions of this evaluation scenario will be drawn with regards to the PaaS service model and could therefore be reused to cater to the needs of supporting the migration to new platforms. The analysis of the potential for exploiting the concept after the conclusion of this dissertation intends to facilitate the adoption of the concept by PaaS adopters that need support to cloud migrate to PaaS.

[592] The evaluation scenario of cloud migration to PaaS offerings used the availability of the PaaSport EU research project (cf. PaaSport Consortium (2016a), n/a, URL see Bibliography) for the first evaluation scenario of this dissertation. For reference, see multiple publicly available deliverables: PaaSport Consortium (2014a), pp. 1–149; PaaSport Consortium (2015), pp. 1–59; PaaSport Consortium (2014b), pp. 1–131; PaaSport Consortium (2016b), pp. 1–29

Evaluation method

Four workshops are designed to gather the feedback of twenty-nine respondents from different organisations. The respondents are representatives of PaaS consumers which belong to one of four associations of IT-related SMEs[593]. They are decision-makers with strong software engineering background in charge of cloud migrating their application systems to PaaS offerings. They are usually either project managers or managers. Their feedback reflects the analysis of the cloud migration alternatives suggested by the proposed concept to migrate their application systems to the selected PaaS. The proposed concept suggests to organisations cloud migration alternatives according to the description of their organisations' cloud migration criteria and of their on-premises application system. They assess the impact of the cloud migration and the portability functionalities, in terms of the effectiveness reached when migrating their application systems to cloud environments due to the use of the cloud migration decision support concept.

The respondents assess the design to migrate their application systems to cloud PaaS offerings proposed by the cloud migration concept. The proposed concept offers the on-premises modelling module to let organisations describe their target application system to migrate their business application systems to Platform-as-a-Service offerings or between PaaS offerings. The cloud environments modelling module is used to model the PaaS offerings the four PaaS environments considered: Flexiant, Amazon web services, CloudBees, and Google cloud. The evaluation focuses on the assessment on the cloud migration functionalities and the cloud migration of application systems to PaaS and between PaaS offerings rather than on the integrated cloud environments modelling. The aim of the evaluation with this scenario is to gather qualitative feedback of whether the functionalities of the proposed cloud migration DS concept helps participants cloud migrate their application systems. The decision-making processes to cloud migrate their actual application systems to PaaS offerings is informed by the priorities and business

[593] Four organisations are involved in the evaluation scenario via one of the four associations of small and medium enterprises of IT professionals from Germany, Latvia, Sweden, and Turkey that belong to industries of ICT, ICT and education, and gaming: a German association of IT SMEs (BITMi), a Latvian professional association of ICT providers and ICT educational institutions (LIKTA), a Swedish games industry professional association (GCM), and a Turkish ICT organisation (TBV)

environments of the organisations, both of which the participants model with the assistance of the proposed concept.

The questionnaires and open discussions prompt the PaaS consumers to assess the efforts they underwent to design a cloud migration alternative in comparison to a scenario whereby they do not use the proposed concept. The questionnaires use a 5-point Likert ordinal scale[594]. In addition, the evaluation focuses on the functionalities that the proposed concept offers for the particular needs of PaaS consumers. The goal is to weigh whether PaaS consumers need these functionalities and whether the proposed concept fulfils their needs with them. Finally, part of the discussion analyses the potential for exploitation of the cloud migration concept if integrated into a PaaS marketplace.

Core evaluation aspects

The evaluation of the proposed concept via this scenario focuses on three aspects. One aspect entails assessing how the proposed concept benefits organisations (in this case PaaS consumers) by assisting them in cloud migrating their application systems. And in particular, how the functionalities of the proposed cloud migration concept support the cloud migration to PaaS offerings. The second aspect involves the analysis of whether the cloud migration concept fulfils the cloud migration criteria of the involved organisations which are PaaS consumers. Finally, the third aspect focuses on how to increase the potential for exploitation of the proposed concept by analysing the impact of integrating it in a PaaS offerings marketplace to facilitate the migration between cloud offerings PaaS configurations and the adoption of the concept.

7.2.2 Cloud migration project modelling

Multiple cloud migration criteria drive the decision of how to cloud migrate an application system to PaaS offerings. Figure 7.15 shows the AHP model of this decision to migrate to the CAS Open cloud environment. Although five kinds of cloud migration criteria are included in the hierarchy, this section focuses on the portability because of its importance for the migration of computation and data.

[594] The 5-point Likert ordinal scale that the evaluation scenario uses: 1) strongly disagree, 2) disagree, 3) neither agree nor disagree, 4) agree, and 5) strongly agree

The portability of data and application system components is cumbersome given the many decision-making processes to undertake. Enterprises and organisations of small and medium sizes usually consider migrating to PaaS technological solutions to save costs and increase the speed with which they market their software solutions. However, when software solutions are tailored to a particular PaaS cloud provider, organisations might be negatively affected by vendor lock-in issues.

The proposed concept aims to tackle these portability issues according to the with the high-level architecture illustrated in Figure 7.16. It incorporates the broker to streamline the migration of software products to different PaaS providers considering all steps in the life cycle of the target software solutions. Stages that entail identifying PaaS offerings in the market, analysing them with respect to the organisation's cloud migration criteria to decide which PaaS offering fits in the framework of the requirements of organisations while making it easier to deploy cloud-enabled application systems while facilitating the management and monitoring IT-enabled processes. PaaS offerings offer a higher-level development platform which hides low-level details from the developer, such as the operating system, the load balancing, and the data storage and access; this, in turn, increases the application system development performance and reduces the capital expenditure. PaaS developing platforms enable programmers to write code and create application systems without worrying about software versioning or limited infrastructure resources. A PaaS provider makes a runtime framework available so that developers can more easily program and execute the application system's code. For example, there are libraries and APIs to provide easy access to the computational and storage resources. In particular, organisations can take advantage of productivity gains that allow them to develop and deploy their application systems. By leveraging PaaS, development teams could be more likely to dare to experiment with various configurations, evaluate the performance of their application systems, and identify compatibility issues during the testing and deployment phases of the applications development. The use of the Platform-as-a-Service model in the business world enables companies to concentrate on their main business goals.

[595] Own illustration following the Analytic Hierarchy Process (AHP) that Figure 4.4 highlights in Section 4.2 of Chapter 4

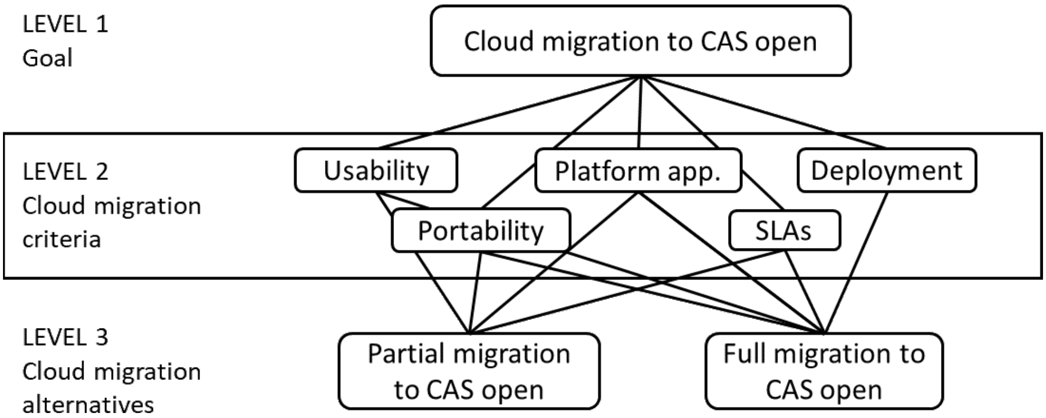

Figure 7.15: Cloud migration criteria and alternatives in the cloud migration to PaaS offerings evaluation scenario[595]

New PaaS providers face tough competition. Giant cloud providers have the financial backing to invest in marketing and publicise their offerings even at a loss. The proposed concept tailored to the migration to PaaS leverages the cloud migration broker that might boost the ability of organisations and enterprises to cope with vendor lock-in issues. They can choose and move between different cloud PaaS offerings as they select the most reliable, well-reputed, or cost-efficient PaaS offering; or simply the one that meets their technical requirements. The cloud migration concept used for this evaluation scenario[596] intends to let organisations more effortlessly and openly change their PaaS providers as required. For instance, when a SLA is violated or when costs become excessive. Transferring an organisation's data and applications systems should not be at the expense of losing any data or being unable to run the application system on a different platform. The proposed concept uses the cloud environments modelling module and its output model to capture the particularities of the selected PaaS offering and the adaptation of the target application system. These models help organisations by comparing PaaS offerings taking into account various factors such as resource allocation, pricing, and quality of service QoS models. This assists organisations in selecting from different offerings with varying characteristics the one that best matches the computing needs of the services and application systems that organisations intend to migrate to PaaS.

[596] Figure 7.16 shows the architecture of the cloud migration concept used adapted to this evaluation scenario

7.2.3 Evaluation results and interpretation

The evaluation is based on the responses by the twenty-nine respondents related to PaaS adoption and more detailed feedback collected via discussions with two project managers of one of the participating SMEs after showing them the cloud migration concept to subsequently address the topic of cloud migration to PaaS offerings.

Cloud migration support for PaaS migration

The assessment first considers the cloud migration functionalities of the proposed concept tailored to the specific case of migrating to Platform-as-a-Service offerings instead of to Infrastructure- or Software-as-a-Service deployments. Accordingly, some of the questions and discussion revolve around the topic of PaaS portability and interoperability, vendor lock-in issues, and PaaS-specific migration challenges. The focus of this evaluation scenario lies in the migration of the application systems of the respondents to PaaS offerings and between PaaS offerings. In addition, the success of the cloud migration to PaaS offerings is evaluated with respect to the cloud migration criteria in conjunction with the assessment of the potential for exploitation of the cloud migration concept in upcoming PaaS marketplaces.

This is according to the feedback of the two project managers assessing the PaaS-based cloud migration supported by the proposed concept and their experience cloud infrastructures and PaaS in particular. PaaS consumers understand that moving their application systems and data to a particular PaaS offering can trap them due to vendor lock-in issues. That happens regardless of the (uneven) support that platform providers offer for portability and compliance to standards. These two respondents acknowledge that some PaaS offerings pose greater risks than others, which is why a decision support concept to assist in the cloud migration comes in handy.

Cloud migration criteria fulfilment

The proposed concept might, according to these two primary respondents, relieve PaaS consumers from part of the burden of failing to choose appropriate offerings according to their cloud migration criteria as several of the offered functionalities make the application system increasingly portability-friendly. The responses of the twenty-nine respondents, PaaS consumers, rate most questions to assess the design of the PaaS-enabled cloud migration alternatives suggested by the proposed concept quite favourably —most of the rated questions show median values around four (in a 5-point Likert scale). This included rating whether the proposed concept improves the migration of their legacy systems to PaaS by suggesting cloud migration alternative designs and it also includes rating the cloud migration design between PaaS offerings (ratings average 4.03 in a 5-point Likert scale).

There is, according to the ratings averaging 3.28, potential for improving the on-line monitoring of the cloud migration criteria and, accordingly, of SLA contracts; which makes sense as the proposed concept offers cloud migration alternatives to fulfil the cloud migration criteria at design time rather than at run time. In addition, the two primary respondents acknowledge the value added provided by the proposed concept as they consider a useful innovation the possibility of considering the architecture of the on-premises application system to recommend the migration to a particular PaaS offering according to the suggested cloud migration alternatives.

Overall the feedback related to the cloud migration functionalities for a PaaS cloud migration is positive as the ratings denote an improvement for four questions directly related to the functionalities and the subsequent discussions, which relate to the core contribution of the proposed cloud migration DS concept.

Exploitation potential of the proposed concept

The third aspect of the evaluation assesses the potential for exploitation of the proposed concept. This aspect depends to a large extent on the already mentioned functionalities to support the design of PaaS cloud migration alternatives that fulfil the cloud migration criteria of organisations. In addition to the added value of the concept, other factors like how easy it is to integrate, use, or understand the proposed concept might incentivise or hinder its adoption for PaaS migration support and, potentially, for PaaS marketplaces in particular. This assessment relies on the feedback by the two primary respondents of the evaluation. The integration of the cloud migration concept in a marketplace would depend on the modelling of the offered PaaS offerings so that these are taken into account to generate valid cloud migration alternatives.

The understandability is a metric that includes attributes describing how easy it is for PaaS consumers to understand the proposed concept and its applicability to real-life problems. It factors in how easy it will be to integrate the evaluated concept in other application systems and to potentially add additional software components.

The two primary respondents assess the understandability quite positively. They give a median value of four to the dimensions related to understandability and easiness of use with fewer ratings lower than the median value. The levels of understandability of the the cloud migration concept show that it lets the two respondents accurately and timely finish a cloud migration design task instead of reducing their productivity by forcing PaaS consumers to devote more time and effort to perform the task than actually needed. If this were the case, PaaS consumers could be motivated to manually undertake the cloud migration or to potentially use an alternative marketplace.

7.2.4 Discussion and evaluation lessons learnt

This discussion is based on both the results of the questionnaires filled in by the twenty-nine respondents and the informal discussions with them; in addition to the more in-depth discussion with two stakeholders. The PaaS consumers discuss the cloud migration alternatives according to their experience working with their target application systems and their cloud migration criteria. The informal discussion is guided by a set of pre-prepared questions about the cloud migration concept and the cloud migration alternatives it suggests[597]. This set of questions is designed to it easy for PaaS consumers to explain the basis behind their responses to the questionnaires in order to unveil their underlying views and motivations as well as behavioural triggers and barriers. Their responses shed light on whether they need the functionalities to assist them when cloud migrating their systems and

[597] Multiple questions compose the set of pre-prepared questions about the cloud migration concept and its suggested cloud migration alternatives in order to guide the discussion: How would you rate overall the usefulness of the proposed concept and its integration in the PaaS marketplace? What would you consider the most important highlights and functionalities of the cloud migration concept? How would you assess the potential time-to-market for a fully developed cloud migration concept service? What is the major drawback of the proposed concept and its integration in the PaaS marketplace? Which functionalities would you add to the proposed concept and its integration in the PaaS marketplace? *Any comments on the technological issues that the proposed concept*?

to which extent the proposed concept fulfils their needs to be supported and the achievement of their cloud migration criteria. The respondents explain how would they use the proposed concept, why, and how easy it is to understand the concept.

All in all, the discussion helps analysing the three core evaluation aspects[598] to derive eight evaluation lessons learnt that contribute to highlight potential improvements to the cloud migration concept. These lessons learnt result after analysing whether the cloud migration concept fulfils the requirements explained in Chapter 5 according to the respondents. This evaluation scenario is selected given the potential to contribute to a research field relatively immature at the time of writing this dissertation and the potential to apply the lessons learnt to new mechanisms and processes that PaaS providers implement to provide cloud services. PaaS developers use the different tools and libraries provided by providers in addition to particular programming and services. The findings explained in this evaluation scenario could be reused in similar setups. Additionally, the evaluation scenario provided a test bed to benchmark the exploitation potential of the proposed concept. The two primary respondents are also relevant to analyse their perspective as PaaS providers of the app-based CAS Open cloud environment (in addition to other ten stakeholders, who attend the workshop and who are also PaaS providers that participate in the discussion) providers also take part in one workshop, they do not fill in the questionnaires mentioned above but participate in the discussions. They add their perspective as PaaS providers and thereby offer a broader vision into studying how to increase the potential of exploitation of the proposed concept for entire ecosystems of marketplaces of PaaS offerings.

Cloud migration semi-automatic mechanisms

The respondents participate in the workshops because they are interested in migrating their application systems to PaaS cloud environments or between PaaS cloud environments. Twelve of the respondents were start-ups and SMEs already running their products in cloud offerings and favoured profiting from using the proposed concept to migrate from one cloud provider to another one according to their cloud migration criteria. Their main cloud migration criteria for migrating to a

[598] The evaluation scenario setup explains these three core evaluation aspects: how the proposed concept helps organisations in the cloud migration to PaaS, whether the concept supports them more easily realise their cloud migration criteria, and how easily the concept could be exploited

different PaaS offering had to do with financial and performance-related aspects. In addition, most respondents consider that some of the most important relevant cloud migration criteria for them to adopt PaaS offerings relate to increasing their business agility and reducing their costs.

On the whole, the respondents, who are project managers in charge of the cloud migration, express during the discussions that they find it fundamental to use cloud migration support in order to be less dependent on the selected Platform- or Infrastructure-as-a-Service offering. Three of the project managers share in an informal discussion that the proposed concept could benefit from the further development of cloud migration mechanisms with the proposed concept increasingly abiding to upcoming and existing standards. According to their opinion, these actions have the potential to solve vendor lock-in problems and to reduce uncertainty when selecting PaaS offerings. Similarly to the decision-making process involved in cloud migrating an application system, the automation of the cloud migration process depends on the type of target application system, which directly affects the necessary pre-deployment steps and the needed update management mechanisms.

Lesson learnt 1: to continue the development of semi-automatic cloud migration mechanisms for the cloud migration concept to fully integrate PaaS standards (and upcoming standards), pre-deployment and deployment steps, and updates administration, and cloud management. Supported requirements: *R3*, *R5–R6* (*R7–R13*).

Cloud-based data portability

Most of the twenty-nine interviewed PaaS consumers reported that their application systems access user and customer data in databases; typically, relational databases. This is a common obstacle for PaaS consumers when migrating their application systems to or between PaaS offerings[599]. To that extent, PaaS consumers might benefit from using complex functionalities that extend the data portability mechanisms of the cloud migration concept in order to make it easier for them to move data between database versions and PaaS providers and their

[599] For more details on the challenges to migrate user and customer data to PaaS offerings, see a previously co-authored research paper: cf. Surajbali and Juan-Verdejo (2014), pp. 278–280

offerings. A project manager and manager in an SME consider that although the cloud migration concept effectively assists the data migration, it should consider using advanced portability libraries. This would mean adding extra steps to the cloud migration concept to automate the migration of user and customer data to the selected PaaS offering. The selection of a the particular PaaS offering partially depends on how the target application system will use the PaaS offerings and the business models they implement.

Lesson learnt 2: to add finer-grained PaaS cloud migration mechanisms to increase the automation of the migration of user and customer data from (relational) databases to PaaS by adopting advanced data portability libraries. Supported requirements: *R1–R2*, *R5–R7*.

SLA- and monitoring-based cloud portability

With regards to going towards automating portability, the discussion with the participating cloud migration stakeholders that have already adopted PaaS computing models bring about that the provided models and APIs could unify the interaction and deployment to PaaS offerings. Especially, when providing connectors to well-established and industry-leading PaaS providers. They express the importance of improving, in the long term, the portability libraries and of further automating the proposed concept to make the process more effortless. Both improvements on the proposed concept could lead to increasingly automating the cloud migration, which could, according to the two primary respondents, expose the full potential of the proposed concept applied to marketplaces. They share that the concept of the monitoring approach for service level alerts is applicable to real-world settings whereby consumers of PaaS offerings (or offerings at any level of the cloud stack going beyond this evaluation scenario) migrate their application systems to different offerings or cloud providers upon SLA violations.

Lesson learnt 3: to advance the implementation of the cloud portability libraries and automate the cloud migration criteria monitoring as SLA monitoring that automatically triggers cloud migration actions. Supported requirements: *R1–R2*, *R5–R6*.

Lower-level cloud portability and interoperability mechanisms

Cloud portability and interoperability at the Infrastructure-as-a-Service level is quite mature due to the reach of its standardisation. At the time of this evaluation, only a few standards directly target the PaaS level, which makes the analysis of portability and interoperability at the PaaS level even more interesting. PaaS deployments show many particularities —with respect to the IaaS on which PaaS builds upon— that result in different PaaS capabilities and functionalities that need specific standards for them. The PaaS service model might benefits less from standardisation than its IaaS counterpart because of how greatly platform components, capabilities, and functionalities vary between PaaS providers and offerings. These characteristics complicate standardising at the PaaS level and accordingly, the discussion with the majority of PaaS consumers bring forth that they perceive as high added value that the proposed concept assists the design of migration between PaaS offerings —that is, PaaS portability— by distinguishing between different levels of granularity such as (micro)services and software components in addition to data and network resources. These respondents consider that it is helpful to get suggestions by the proposed concept at this high level as it considers platform- and cloud-specific requirements and restrictions. Cloud-related requirements and restrictions that are modelled with the cloud environments modelling module of the cloud migration concept.

The discussion with a project manager and a business manager of an SME suggest that the future work of this dissertation (that is, out of the scope of this dissertation) should address portability and interoperability issues at a finer-grained level. This includes adding more implementation details to the cloud migration alternatives that are related to the selected PaaS offering so as to accommodate the differences in management APIs. At the same time, they consider the unified API integrated in the cloud migration concept at the PaaS level as a well-thought key asset. These two respondents mention that the use of portability libraries and the monitoring of the cloud migration criteria offer the potential to help them protect their investments. However, the development of the concept should, in their opinion, deepen in the automatic deployment and portability —of both data and computing— in addition to the flexibility offered by the suggested cloud migration alternatives.

Lesson learnt 4: to address portability and interoperability at a lower level of abstraction including additional PaaS management APIs and implementation mechanisms for deployment and portability execution (in addition to planning). Supported requirements: *R1–R2*, *R4*.

Integration of the proposed concept in PaaS marketplaces

A small group of project managers of different PaaS consumers organisations are motivated to consider the effects of using the proposed cloud migration concept in conjunction with the PaaS marketplace they want to form with other organisations that act as PaaS providers. The planning of the deployment to the PaaS offerings available in the marketplace could arguably become easier when using the proposed concept. The concept includes the pre-deployment configuration and components re-scattering. In addition, it was comforting to the discussion group that the concept allows them to easily revisit the PaaS selection phase with a lower risk of damaging the performance of their application systems.

The aforementioned small-scale discussion highlights the potential for the proposed concept to reduce the time-to-market of their PaaS products in a cloud marketplace. To achieve that, firstly the potential for exploitation could be, in their opinion, increased —an option out of the scope of this dissertation could be to offer additional degrees of automation for the deployment and portability of the cloud migration alternatives— to then advance in the integration of the cloud migration concept in a marketplace to facilitate the PaaS migration and portability. In terms of exploitation, they share that they would consider a motivating aspect to join a marketplace using such a cloud migration concept given that a marketplace reaches a critical mass of PaaS providers competing in price and value. Value competition leads PaaS to differentiate their cloud offerings by providing different capabilities, services, and configurations subject to cloud provider-specific restrictions. The key here is striking the right balance between differentiation, to increase their market share and retain it under market pressure, and creating vendor lock-in issues. The discussion shed light on how these trade-offs —between price and value or between favouring PaaS providers or consumers— show the potential of combining a marketplace with the proposed concept so that PaaS consumers can adopt an PaaS offering now and migrate to a different one later on. It is more interesting to them to join a vibrant marketplace, with lots of PaaS providers, that gives them multiple options to reversibly select the most beneficial PaaS offering at the time. This could attract PaaS consumers to a marketplace that offers them the cloud migration concept to avoid vendor lock-in within a virtuous circle whereby more PaaS consumers attract more PaaS providers and vice-versa.

Lesson learnt 5: to integrate the proposed concept with marketplaces of PaaS offerings in order to motivate PaaS providers to join it, given that the concept makes offerings searchable and paves the way to access a potentially large pool of PaaS consumers in a marketplace.
Supported requirements: *R2*, *R7*.

Incentivising the adoption of the proposed concept for exploitation

The above lesson learnt draws attention to how the proposed concept can provide incentives to PaaS providers and consumers to join a marketplace. But things can as well go the other way, a PaaS consumer fears that their ratings could endanger the business of PaaS providers and deter them from joining a marketplace that integrates the proposed concept. Especially, in cases whereby they would not profit from serving numerous PaaS consumers. This situation would turn into a negative vicious circle encouraging PaaS providers to leave a marketplace. The proposed concept should not become a double-edged sword that damages cloud providers while helping cloud consumers overcome their challenges related to cloud migration. A large pool of PaaS consumers using a marketplace could help PaaS providers overcome these hindrances but it is difficult to achieve in the initial stages of a marketplace. The proposed concept should, according to the PaaS consumer, avoid migrating application systems to a different PaaS offering as a result of temporary QoS degradation. It should only happen in cases of long-term degradation or otherwise it would make it difficult for PaaS providers to create long-lasting provider-consumer sensible relationships. A temporary degradation of some of the cloud migration criteria should not necessarily trigger a full-blown migration to a different PaaS offering unless the deployment violates the design constraints of the organisation.

Lesson learnt 6: to consider the holistic set of cloud migration criteria in the long term and not just a temporary problematic set of criteria, especially, when achieving higher degrees of automation in the on-line continuous cloud migration support.
Supported requirements: *R5–R6*.

The effects of cloud portability on small PaaS providers

The two primary respondents refer to the potential advantage that the proposed concept makes available for small organisations that supply PaaS offerings as its portability functionalities makes it less risky for PaaS consumers to select a smaller or less known provider. PaaS consumers can always easily move to another provider were their previously selected provider to go out of business or fall short of the cloud migration criteria. This way, small competing players could attract investments to them and foster the adoption of their PaaS offerings. Additionally, the use of the cloud migration functionalities could bring about risks of losing their client base to PaaS providers.

Lesson learnt 7: to also address the needs of PaaS providers because finding a balance between their requirements and those of PaaS consumers could have better chances to bring a successful cloud migration concept to the market, especially, if integrated into a marketplace.
Supported requirements: *R1–R2*.

Economic effects of the proposed concept in the business of PaaS providers

One of the PaaS providers points out that recommending or rating higher different PaaS offerings depending on their technical capabilities could offer some added value as this could reveal new business opportunities. Accurate suggestions by the proposed concept to use a specific PaaS offering to run particular application systems according to a set of cloud migration criteria could help PaaS providers target the long tail of the market. That is, with smaller Platform- and Software-as-a-Service offerings specialising in some kind of user needs. These providers might move their products to blue oceans and would, according to their feedback, benefit from being able to offer their consumers additional support to pre-deploy simple application systems and war files. This would come in handy in addition to the mechanisms for defining SLAs matching the cloud migration criteria that the two primary evaluators rate as intuitive. In order to increase the exploitation potential of the cloud migration concept, these same mechanisms could arguably be used to implement monitoring services in agents that could trigger cloud migration design efforts. However, the two primary respondents are sceptical of how complex monitoring tasks would work in real settings as well as their effects on the migration to different cloud offerings.

It is fundamental for the cloud migration concept to have access to details about the PaaS offerings in order to assist cloud consumers in selecting the cloud offering appropriate to them. Therefore, creating an actual product based on the proposed concept requires a business-oriented strategy to stimulate PaaS providers into describing their offerings and into delivering appropriate interfaces to make Quality-of-Service measurements available. This might be easier given that it might be beneficial both for providers, to provide a better service, and for their potential customers, which might require this information.

Lesson learnt 8: to combine business incentives with technical aspects and interfaces to help in monitoring SLAs and, accordingly, cloud migration criteria so that they offer their PaaS offerings in a marketplace and bring it to a success. Supported requirements: *R2*, *R5–R7*.

7.2.5 Evaluation-based cloud migration concept refinement

The iterative refinement of the cloud migration concept entailed smoothly and dynamically moving back and forth from the requirements for cloud migration to the design and implementation of the proposed concept according to the eight lessons learnt during the evaluation. Given the interest of the PaaS consumers in participating in a marketplace of PaaS offerings and the opportunities this presents for the exploitation of the proposed concept, Figure 7.16 shows a software architecture designed to integrate the concept in a marketplace of PaaS offerings[600]. The exploitation opportunities stem from the fact that integrating the cloud migration concept in a vibrant ecosystem of PaaS consumers and providers benefiting from

[600] Figure 6.15 shows in Chapter 6 the software components composing the initial architecture of a marketplace of PaaS offerings integrating the proposed concept

participating in marketplace that offers them win-win relationships might open the way for the participants in a potential marketplace to adopt the proposed concept for their cloud migration projects after testing it within such a marketplace.

The refined architecture intends to cater to the needs of marketplaces by brokering cloud offerings, in general, and PaaS offerings, in particular. The elements of Figure 7.16 in grey colour are fundamental components of the cloud migration concept that are fully designed as part of the concept and its prototypical implementation. The white components mark the additional software components that implement functionalities, according to the eight lessons learnt with this evaluation scenario, to overcome the stated challenges and take advantage ot the opportunities. These software components are necessary to integrate the cloud migration concept in a PaaS offerings marketplace that are not fully implemented. These include the client and broker components at the top as well as the cloud portability tier on the bottom right-hand side of Figure 7.16.

The architecture in Figure 6.15 in Chapter 6 shows an initial customisation of the proposed concept that uses a cloud migration broker. That architecture does not focus on PaaS offerings in particular neither is it designed according to the needs of SMEs offering and consuming cloud services. The assessment of this scenario motivates defining the cloud portability tier on the right-hand side of Figure 7.16 in addition to including additional components for the specific case of SMEs offering and using PaaS offerings. At the top, the client component for small and medium enterprises provides the UI to access the cloud migration decision marketplace broker. Figure 7.16 shows, on the left-hand side, the four PaaS providers considered in this evaluation scenario: Flexiant, Amazon, CloudBees, and Google Cloud.

The cloud offering models repository stores the models of these PaaS providers and their offerings so that PaaS consumers can migrate their application systems to the PaaS offering that the proposed concept suggests.

The cloud portability tier —on the right-hand side of Figure 7.16— emerged as a natural outcome of customising the proposed concept to the cloud migration to

[601] Own illustration partially based on the first version of the figure published on cf. Surajbali and Juan-Verdejo (2014), p. 279; and PaaSport Consortium (2014b), p. 11

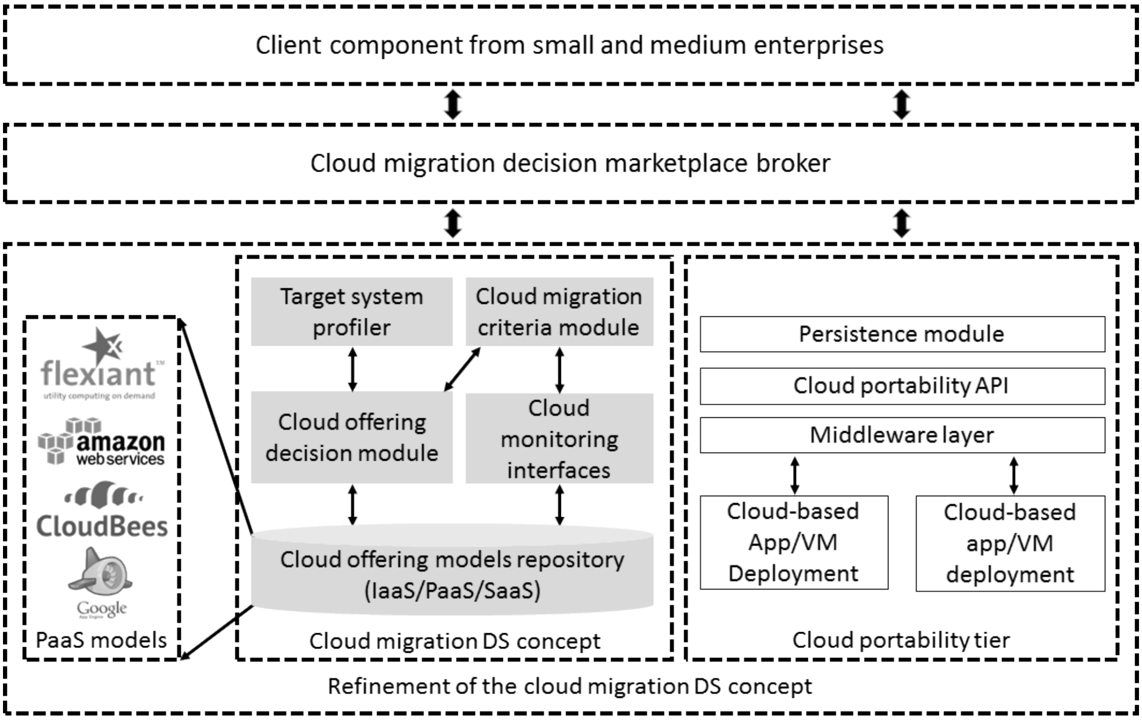

Figure 7.16: Architecture of the proposed cloud migration DS concept integrated into a PaaS offerings marketplace[601]

PaaS offerings. Portability is a very important characteristic when selecting PaaS offerings because the support of different cloud providers differ very much[602]. In addition, it is a fundamental factor for marketplaces of PaaS offerings. The fact that marketplace consumers can shift from one PaaS provider to another makes them more interested in using a marketplace and in taking the leap of selecting one PaaS offering as later on they can change to a different one. The PaaS offering selection is not such a definitive and irreversible decision. Nevertheless, it is an important decision that they take carefully after considering the multiple decision aspects driving their architectural decisions and PaaS selection. At the time of writing this dissertation, portability at the Infrastructure- and Software-as-a-Service are in a better shape than at the Platform level. Therefore, this evaluation scenario is selected as it enables focusing on the research about the open questions of PaaS portability and how this affects the cloud migration decision.

Application portability issues in the PaaS market usually make providers compete at the expense of the application system portability and interoperability. This evaluation scenario adapted the cloud migration concept to the considered semantically-

[602]For more details on portability Sections 6.6] and 6.7 explain portability across all three levels of the cloud stack with respect to cloud migration in Chapter 6

enhanced marketplace of interoperable PaaS offerings. The marketplace allows for the deployment and migration of business application systems coming from small and medium enterprises. The semantically enhanced marketplace of PaaS offerings took a business-driven approach to overcoming the barriers and obstacles that small and medium enterprises might find to enter the cloud offerings European market. Figure 7.16 shows the architecture adapted to this particular scenario so as to help the consumers of a marketplace (SMEs) migrate their business application systems to PaaS offerings according to their different interests, cloud migration criteria, and the architecture of their application systems.

The architecture and functionality of the proposed concept, as Figure 7.16 shows, facilitates the decision behind designing cross-platform architectures and the use of the modules for persistence, execution, and coordination in multi-cloud environments. This evaluation scenario tests the proposed concept and its integration with the aforementioned modules to prototypically initiate the cross-cloud deployment to meet the needs of PaaS providers —especially European SMEs providing cloud platforms— and the cloud migration criteria of PaaS consumers —such as application system developers using PaaS offerings. The proposed cloud migration concept focuses, at the PaaS level, on addressing the interoperability, migration, and scalability challenges in heterogeneous vertical markets using PaaS offerings.

7.3 Scenario 2: cloud migration of business intelligence

Business intelligence and analytics denote integrated approaches to IT-based management and decision support based on multi-layered architectures. The fundamental system in BI is the Data Warehouse (DWH): an integrated, application-spanning data repository that contains data extracted from multiple internal and external data sources.

These data sources feed application-specific data pools, like data marts, so that analysis and reporting application systems can access them. BI and analytics systems range from on-line analytic processing (OLAP), data mining / advanced predictive analysis tools, to budgeting, planning, and balanced scorecard application systems. BI application systems integrate such a heterogeneous IT landscape by leveraging tools for administration and metadata[603].

In addition to the heterogeneous nature of the software components of BI application systems, specific characteristics of BI make its cloud migration a challenge. For example, modern BI application systems might have to cope with large volumes of data or abide by governance frameworks tailored to the BI organisation and the use case. Other systems, like those implementing operational BI, need support to perform timely business decision-making in near real-time. Under the heading operational BI, BI systems support operational decisions; for example, in the realm of process management[604]. The analysis of large volumes of structured and unstructured data becomes increasingly relevant for BI application systems that use big data and, hence, cope with their inherent volume, variety, and velocity, while ensuring veracity in its generation, recording, and publishing[605]. In many companies, specifically-designed BI governance frameworks are tailored to their architectures, processes, and application systems[606]. This heterogeneity needs to be coupled with the cloud migration criteria of BI organisations in order to support the privacy-aware cloud migration decision that takes the architecture of the BI application system.

While some BI software components do not contain much sensitive information —for example for web usage mining—, others deal with strategically relevant data, or need to comply with various legal requirements. Such is the case of (legal) reporting or user data analysis. From this perspective, a complete BI system cannot easily and completely be migrated to cloud environments. Nevertheless, cloud offerings might ease existing agility pressures and be a response to volatile capacity requirements (such as in planning or budgeting) and provide the necessary computing environment to address project-specific BI needs by providing, for example, cloud-powered social media analysis[607].

[603] cf. Ereth and Baars (2020), pp. 4–6; Baars and Kemper (2008), p. 132; Shollo and Kautz (2010), p. 8; Negash (2004), pp. 187–190
[604] cf. Baars and Ereth (2016), pp. 7–10; Golfarelli et al. (2004), p. 1; Marjanovic (2007), p. 215
[605] cf. Zhong et al. (2017), pp. 625–627; Manyika et al. (2011), pp. 1–5; Jacobs (2009), p. 36
[606] cf. Ereth and Baars (2020), pp. 5–7; Baars et al. (2010), p. 1066
[607] cf. Ereth and Baars (2015), pp. 10–11; Zimmer et al. (2012), p. 4189; Baars and Kemper (2010),

The high expectations towards bringing BI to cloud infrastructures[608] make BI systems a relevant area to study how to migrate BI systems to cloud environments. These deployments use software components distributed to both the local and cloud premises. In order to leverage the potential benefits of using cloud environments to run BI application systems, the cloud-based architecture needs to be carefully planned. Organisations need to take many criteria into account in order to adopt an adequate cloud migration alternative that moves the appropriate software components of the application system to suitable target cloud-based infrastructures. Given the complexity of this decision-making process, the proposed cloud migration DS concept assists in the cloud migration of BI systems to cloud deployments in which parts of the application —components, code artifacts, functionalities, or data— are kept at the local premises of the organisation while others are migrated.

The evaluation scenario section firstly explains the setup of the evaluation scenario including the evaluation method followed and the most important aspects of the scenario. Next, the results of the evaluation of the BI scenario are interpreted and used as the cornerstone that guides the qualitative discussion leading to the analysis of several lessons learnt. Finally, a refinement of the cloud migration concept is presented according to the lessons learnt.

7.3.1 Evaluation scenario setup

This evaluation scenario focuses on the migration of typical BI application systems to cloud environments[609]. The proposed cloud migration DS concept assists the BI organisation in cloud migrating its application system according to its cloud migration criteria, the status quo of its BI application system[610], and the particular kind of BI cloud migration project (for example, full versus partial migration) that interests the organisation[611].

pp. 1537–1538; Baars and Kemper (2011), pp. 3–5; Thompson and Walt (2010), pp. 5–7; Baars and Qie (2012), pp. 26–29; Gartner (2012), p. 1, URL see Bibliography

[608] Both market studies and the wide spectrum of BI SaaS services (from almost all large BI vendors) indicate the relevance and interest

[609] This evaluation scenario refers to the second element on the right-hand side of Figure 7.1

[610] The status quo of the BI application system refers to the on-premises application system model prior to the cloud migration

[611] This scenario used for the evaluation relates very closely to previous papers by the author of this

The prototype of the proposed concept is used to model scenarios of typical architectures of BI application systems and the usually observed cloud migration criteria according to the description of practical cloud migration projects and its theoretical study in literature. These are examples of typical BI cloud migration projects that happen in reality as presented in a BI workshop[612]. The examples considered in this section are an spectrum with different BI software components to migrate and different cloud migration criteria that come from the analysis, data preparation, and modelling of cloud migration projects found in literature and BI cloud migration projects. Based on these three kinds of input models —the on-premises model, the cloud migration criteria, and the cloud environments configurations models—, the proposed concept ranked the existing cloud migration alternatives and presented them in order as output. Based on the feedback gathered with the discussions about the cloudification of BI with researchers during the BI workshop, refinements are proposed in two further papers presentations (HotTopics ICPE and, learning from that presentation, Crosscloud INFOCOM)[613]

The evaluation targets of the BI scenario are the cloud migration concept, the partial migration approach, and the highest ranked cloud migration design (that is, the best cloud migration alternative according to the nomenclature of this dissertation) resulting after applying these concepts and prototypes to cloud migrate BI application systems. The resulting cloud migrated BI application systems are studied according to the research collaboration with the chair at the University of Stuttgart and the unstructured discussions that followed the presentation of the cloud-enabled systems in three conferences and workshops.

Motivation

The cloud migration of business intelligence application systems is selected as the second evaluation scenario because it represents a complex cloud migration project that depend on the different cloud migration criteria of BI organisations. In this evaluation scenario, seven cloud migration alternatives with different cloud

dissertation cf. Juan-Verdejo et al. (2014a), pp. 43–48; Juan-Verdejo (2012), pp. 14–27; and Juan-Verdejo and Baars (2013), pp. 35–42

[612] A paper by the author of this dissertation provides examples of BI cloud migration projects and alternatives: cf. Juan-Verdejo (2012), pp. 14–27

[613] Further refinement papers about the cloud migration of BI projects cf. Juan-Verdejo et al. (2014a), pp. 43–48; and Juan-Verdejo and Baars (2013), pp. 35–42

migration criteria and requirements for the on-premises BI application systems according to the needs of each cloud migration project. The cloud migration scenarios target the migration of different software components in complex multi-layered application systems with evolving cloud migration criteria. They provide various testbeds to assess the different cloud migration alternatives in scenarios in which security, synchronisation, and privacy play a very decisive role in the cloud migration due to the strategic nature of the data used in BI.

The conclusions of this evaluation scenario crystallise in lessons learnt that can be reused in other cloud migration projects with multiple software components that might have to run on-premises or on cloud environments in complex architectures. In addition, this evaluation scenario brings about the multi-device nature of current BI with multiple organisations having different views of the data. Data which is continuously updated to be timely analysed in order to support a wide range of business decisions taken by organisations; and enterprises in particular. Finally, the cloud migration criteria play a very fundamental role as, depending on the cloud migration project, organisations have different criteria and priorities related to the adoption of cloud computing as their computational model.

Evaluation method

Similarly to the other two evaluation scenarios (see Sections 7.2 and 7.4), the research frame of reference (see Figure 4.13 in Section 4.7, Chapter 4) identifies the factors that provide a framework to design, build, and evaluate the proposed concept and its composing artifacts according to the principles of design-oriented research[614]. The five components of the research frame of reference systematically structure the evaluation of the concept with this scenario. In particular the first component (cloud migration support) structures the overall cloud migration project and evaluation scenario; in combination with the on-premises application system modelling (second component of the research frame of reference), they help systematically structure the evaluation scenario and the typical architectures for cloud migration projects. Furthermore, the other three components (cloud environments modelling, cloud and inter-cloud architectures, and cloud migration criteria) systematically guide the structure the feedback related to the BI scenarios

[614] Section 1.5 lays, in Chapter 1, the principles of design-oriented research

and the evaluation lessons learnt. The questions used in the unstructured discussions are coherent with these five components of the research frame of reference and are used to gather feedback the assistance provided by the proposed concept and prototype to cloud migrate typical BI application systems. The research frame of reference helps in guiding the discussion about the research artifacts proposed, their functionalities and role in the integrated concept design, and whether they achieve their intended goals.

The feedback was collected from discussions with researchers with experience on running BI in cloud environments and in the field of cloud-based architectures during the three aforementioned conferences. The qualitative analysis showed in this evaluation scenario stems from the feedback collected during three conferences where the results of this evaluation scenario were shown and from academics with expertise on BI. The feedback was collected through discussions about the approach and continuous feedback from members of the department of the Chair of Information Systems I of the University of Stuttgart. Additionally, the cloud migration design was assessed during the peer-review process of publishing the results in three internationally peer-reviewed conferences; in addition to the presentation done and the sub-sequent discussions[615].

The proposed concept was presented as well as its application to the (partial) migration of BI application systems to cloud environments according to cloud migration requirements, which come from the literature analysis and the needs of BI organisations. These requirements include which software components to migrate to cloud environments and why, as well as the cloud migration criteria of BI organisations.

Core evaluation aspects

The evaluation of the proposed concept via this scenario focuses on three aspects. One aspect is analysing how effective the concept is in capturing the architecture of BI application systems to include that in the decision-making process to recommend a partially migrated cloud migration alternatives. Another aspect to analyse

[615]The discussions happened at the *International Workshop Cross Cloud during the INFOCOM conference in 2014)*, the *International Workshop on Hot Topics 2013 in Cloud Services during the ICPE conference in 2013*, and the *4th Workshop in business intelligence in Mainz during the WSBI 2012 conference*

is whether the proposed concept effectively takes into account the requirements of BI organisations that want to design the cloud migration of selected software components. Finally, this evaluation scenario discusses to which extent the concept captures and allows for modelling the cloud migration criteria of BI organisations and especially, those criteria related to data synchronisation and security, which are, according to the gathered feedback, important to cloud migrate BI application systems.

7.3.2 Cloud migration project modelling

The application of the proposed concept to cloud migrate different BI application systems according to their specific particularities is presented in this section. The cloud migration alternatives proposed by the cloud migration concept are then interpreted. The proposed concept assists organisations in migrating their BI application systems[616] by partially migrating them according to the architectural design shown in Figure 7.17. Figure 7.17 is an initial example to cloud migrate a BI system as proposed in a co-authored paper[617]. It shows the usual BI architecture on the left-hand side that includes the Data Warehouse and the BI portal interface that communicate through a broker that mediate multiple cloud-enabled services —big data, OLAP, ETL, and reporting services. An alternative to this architecture might not consider the content management as a so prominent and integrated component of BI for organising unstructured data but use data lakes instead. Data lakes might be used as storage of unstructured (and even structured) data upon which different tasks might be applied including data science approaches, advanced analytics, or machine learning[618]. This instability might reinforce the importance of supporting the cloud migration decision-making processes as BI systems increase in complexity and variety of architectures that might be partially migrated or use hybrid clouds. Subsequently, Figure 7.18 extends the architecture shown in Figure 7.17 to offer a holistic cloud-enabled BI application system that

[616] Figure 6.16 shows, in Chapter 6, the architecture of typical BI application systems that run at the premises of the client organisation

[617] A paper by the author of this dissertation provides examples of BI cloud migration projects and alternatives: cf. Juan-Verdejo (2012), pp. 14–27

[618] The term data lake stems from the collection of raw unstructured data, a source system, that some use as an archive too. Additionally, some authors include the Data Warehouse as part of the data lake by following the relational paradigm. Data lakes might include a metadata layer to organise the different zones such as the sources, transient landing, and raw zones

runs on multiple cloud offerings to overcome some of the limitations of the above architecture. Figure 7.18 is an example used in this section to make it easier to explain the cloud migration alternatives ranked by the concept proposed in this dissertation.

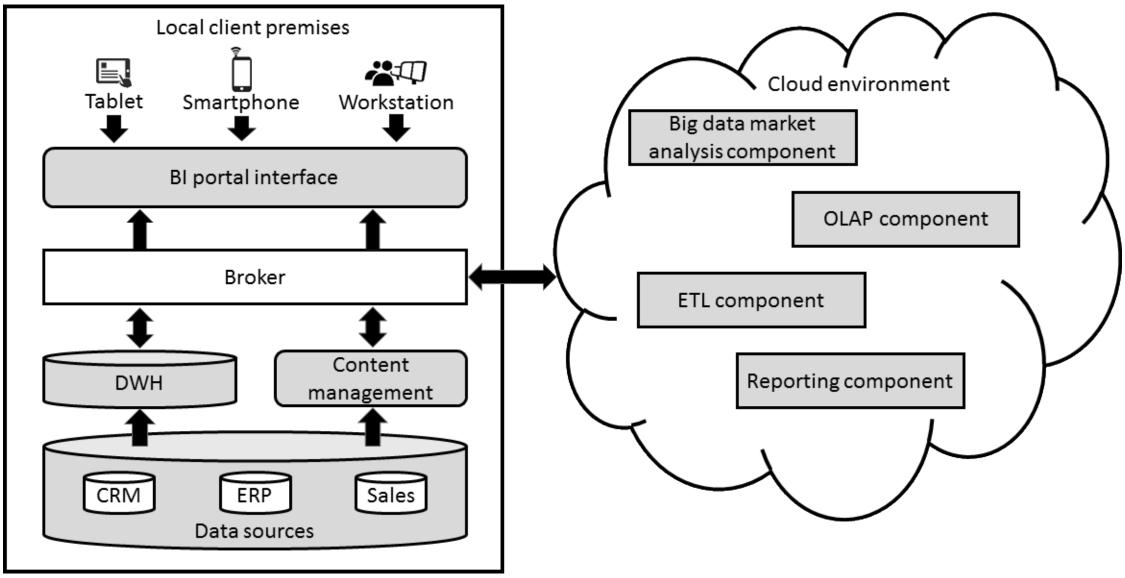

Figure 7.17: Partial cloud migration alternative suggested by the proposed concept to cloud migrate an exemplary BI application system[619]

Typical cloud migration alternatives designed with the assistance of the proposed concept are similar to the architecture presented in Figure 7.17. On its right-hand side, some of the usual software components of BI application systems are migrated to cloud environments, while the data remains locally on the premises of the client organisation —as the bottom left corner shows. These data come from sources such as the CRM and ERP components[620] running at the local premises as well as from the DWH. The broker component interacts with the components which have been newly migrated to the cloud environment. That is, the big data market analysis component, the OLAP component, the ETL component, and the reporting component. This way, the software components on the right-hand side of Figure 7.17 can benefit from using virtualisation in highly scalable environments. The rest of this section describes particular cloud migration alternatives to design the cloud migration of those software components.

[619] Own illustration

[620] CRM stands for Customer Relationship Management and ERP for Enterprise Resource Planning. DWH refers to the data warehouse

7.3.3 Evaluation scenario: cloud migration alternatives

The cloud migration alternatives discussed below are elaborated and extended in three conference papers by the author of this dissertation in collaboration with co-authors from the University of Stuttgart: the *4th Workshop in GI-Fachgruppe business intelligence in Mainz during the WSBI 2012 conference*, the *International Hot Topics Workshop in Cloud Services that is the HotTopiCS 2013 at the ICPE conference*, and the *International Cross Cloud Workshop at the 2014 INFOCOM conference)*[621].

The discussions happened at the *International Workshop Cross Cloud during the INFOCOM conference in 2014)*, the *International Workshop on Hot Topics 2013 in Cloud Services during the ICPE conference in 2013*, and the *4th Workshop in business intelligence in Mainz during the WSBI 2012 conference*

The cloud migration alternatives below respond to the different kind of organisations running their BI systems according to their cloud migration criteria. These are examples of cloud migration scenarios stemming from the literature and practical experience with BI application systems that have been discussed with researchers from three conferences and, according to their feedback, they have been refined to the cases described below.

1.— Inclusion of specialised data analysis functionalities from cloud environments, for example, for big data analysis

BI application systems can source specialised analytic functionalities from cloud environments, especially for temporary projects. For example, when conducting a market analysis using big data solutions on data coming from web sites and social networks. In this particular instance of the evaluation scenario, the proposed concept takes into account the interdependencies of data analysis functionalities with existing data repositories —such as data marts, the DWH, and the underlying data base management application system (DBMS)— to propose suitable cloud migration alternatives. As long as the analysed data is non-sensitive, the proposed concept considers whether migrating data to the selected cloud environment depending on the volume of data that needs to be cloud migrated and the processing

[621] cf. Juan-Verdejo (2012), pp. 14–27; and Juan-Verdejo and Baars (2013), pp. 35–42; Juan-Verdejo et al. (2014a), pp. 43–48;

power needed to analyse those data. Accordingly, the proposed concept can move the data mart to a cloud-based environment which has been selected according to cost rationales.

2.— Cloud migration of OLAP or reporting front-ends

The proposed concept design and its prototype assist here in the design of a less straightforward cloud migration alternative. Tool-based designs migrate the reporting or the on-line analytical processing (OLAP) front-end to cloud environments. Access to data from everywhere and anytime from mobile devices is one of the advantages of having such a BI landscape with software components running on different cloud-based environments. However, as the cloud migration of these components can worsen the reaction times to the user's input and data traffic, the proposed concept factors in these dependencies when ranking the cloud migration alternatives. Dependencies, for example, exist with the device and user management, the web and the application servers, and most importantly, the relevant data repositories. Moving the data mart and the DBMS to cloud environments might alleviate the issues that having different reaction times create but has further implications regarding security —as not only a few reports will be exposed to the web but the whole data repository as well— and due to issues to keep consistency with the DWH. Additionally, when cloud migrating operational reporting, the continuous updates of the data mart running in a cloud environment might induce prohibitive bottlenecks in cost and performance. This might lead to a solution where the DWH or a specifically designed operational data store is cloud migrated. The proposed concept weighs these trade-offs to rank the generated cloud migration alternatives.

3.— Movement of an operational BI solution to the web

The dependencies become even more critical when BI software components trigger events in other application systems. That is the case in the realm of active BI; for example, on the operational level. In this case, the proposed concept needs to assist organisations in deciding where to keep the functionality that triggers the event. Similarly to the case of cloud migrating reporting or OLAP software components, the application system needs to update the operational data in real time. However, in this case the dependencies go in both directions and include transaction management in operational sources and application systems.

4.— Cloud migration of selected ETL procedures; particularly, for data sources already running on cloud environments

A BI application system that includes specialised extract, transform, and load (ETL) routines from a cloud environment can preprocess unstructured data in addition to discovering non-evident duplicate entries —for example, sales for Müller and Mueller, which are the same company or person— in the master data fed to the DWH. The presented cloud migration alternatives should embed these routines into the higher-order ETL process. As the ETL process links several core components, it often involves very large amounts of traffic between components. Hence, the prototype of the proposed concept ranks the alternatives to cloud migration by taking into account how the cloud migration of ETL procedures affect the overall BI application system on various levels.

5.— Holistic cloud-based BI application systems

The proposed concept assists organisations in designing a computing ecosystem like the one shown in Figure 7.18 for typical BI application systems as an evolution of the architecture shown in Figure 7.17. The BI user organisation (see the bottom left corner of the figure) represents the organisation's premises where the BI application system run prior to the cloud migration. The BI user organisation pushes and pulls data to and from the cloud environment marked with (1) in Figure 7.18. Data flow encryption with the Transport Layer Security/Secure Sockets Layer cryptographic protocols (TLS/SSL) secures the push and pull end-to-end communications. Once the BI-User organisation transfers its data to the cloud environments, the cloud-based BI tools analyse and transform those data to generate data analysis and reports.

The DWH stores meaningful and strategic information so that a myriad of different devices —such as the workstation, tablet, or mobile phone at the bottom right corner of Figure 7.18— access them across different layers. These devices use the BI access layer component on the cloud environments' side as (3) marks in Figure 7.18. If the organisation does not entirely trust its sensitive data to a cloud provider, the proposed concept keeps sensitive data locally to the client and pushes the anonymised or pseudo-anonymised data to cloud environments

[622] Own illustration based on Juan-Verdejo et al. (2014a), p. 45

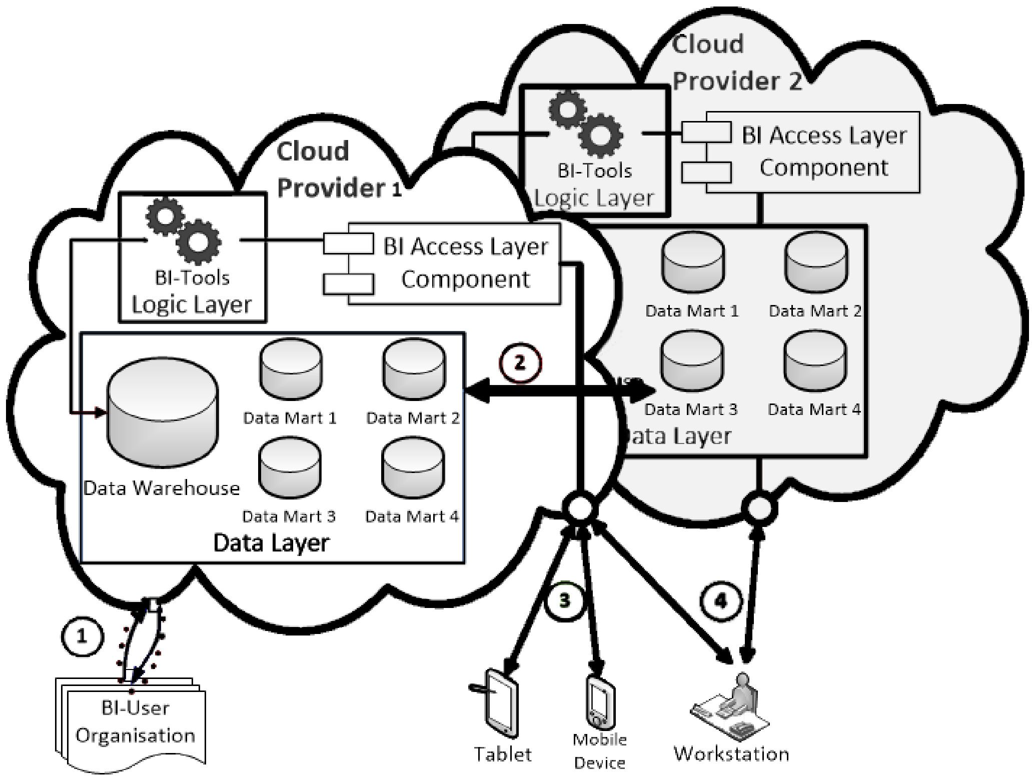

Figure 7.18: Holistic cloud migration of a business intelligence application system to cloud offerings[622]

so that the BI tools can use them —as the (1) in Figure 7.18 shows. Here, the organisation moves their BI tools to Infrastructure-, Platform-, or Software-as-a-Service offerings to use the additional scalable resources and the available BI tools; especially in the case of the Platform- and Software-as-a-Service offerings.

6.— Security and synchronisation in cloud-migrated BI application systems

The proposed concept considers the potential deployment of data and computation across different cloud offerings to benefit from their differences according to the cloud migration criteria of BI organisations, the characteristics of the BI application system, and the cloud offerings attributes. The concept prototype proposes cloud migration alternatives that improve the on-premises BI application system in terms of portability and availability while also focusing on data security. Data security mechanisms that include the use of end-to-end SSL/TLS encryption to secure the communication between the cloud premises over the Internet, as (2) denotes in Figure 7.18.

The proposed concept help organisations in analysing the interactions of the on-premises BI application system with other neighbouring components within the organisation's premises to select the appropriate cloud migration alternative that keeps these interdependencies after the cloud migration. This is a research challenge within the realm of the integration aspect of BI application systems. For example, if an organisation cloud migrates an existing BI software component that heavily interacts with the CRM application system in Figure 7.17, the proposed concept re-engineers the design of the BI application system by taking into account all the dependencies of the CRM. These dependencies describe with which CRM components the BI application system communicates, how often does this communication take place, and the amount of data they interchange. The highest-ranked cloud migration alternative has the BI application system and some CRM components run within the same premises to avoid increased latencies, which could harm the application system's performance beyond acceptance. In the case of migrating a BI application system to a non-trusted cloud provider, the proposed concept helps to design an architecture that uses the connection marked by (2) in Figure 7.18 to synchronise only the relevant migrated data according to their privacy level and the trust on that provider.

7.— BI application systems in multi-cloud ecosystems

Figure 7.18 refers, with (3), to the broad network access the cloud-enabled BI application system offers to heterogeneous mobile devices. The ranked cloud migration alternatives need to transparently postpone the mobile synchronisation when data uploading tasks are interrupted due to mobile devices disconnections from the Internet. Mobile devices will transfer the data once they establish the connection or once the user connects to a trusted and reliable network. Cloud migration alternatives that transparently keep data up-to-date regardless of the used cloud environment —(4) highlights this particular scenario in Figure 7.18— are ranked higher by the proposed concept. Data synchronisation and space computing efficiency use Globally Unique Identifiers (GUIDs) to enforce data consistency between source and target data storage and to harmonise data over time (as (2) shows). In this case, the metadata helps to discover and deliver information while its coordination enables correct data sharing across cloud environments.

The concept prototype, InCLOUDer, assists organisations in designing a cloud migration deployment that keeps a redundant application system that stores harmonised data in a different cloud environment to increase resilience while avoiding vendor lock-in issues. With the use of fault isolation, failures do not spread to healthy software components. Organisations can manage the BI application system from an isolated and safe computing environment in case of failure or decreasing quality of service. The provided quality of service can be assessed using the metrics for the cloud migration criteria or via user intervention. The cloud migration concept ranks higher cloud migration alternatives that tolerate failure by having a redundant BI application system running on a different cloud environment in oder to ensure availability. Likewise, the same applies if a particular cloud provider went out of business or failed to comply with the cloud migration criteria or the agreed Service-Level Agreements.

7.3.4 Discussion and evaluation lessons learnt

The evaluation of the BI scenario is based on the refinement of the seven exemplary BI migration scenarios according to the systematic discussion of the seven cloud migration alternatives (described in the previous sub-section) in the three aforementioned different conferences. This evaluation arguably supports the assumption that organisations need assistance in partially moving their target application systems to local and cloud environments instead of an all-or-nothing cloud migration. The proposed concept intends to support the cloud migration decision by keeping parts of the application system at the local premises of the client organisation while migrating other software components to cloud environments based on the three input models of the proposed concept; that is, the cloud migration criteria model, the application system model, and the cloud environments model. The set of cloud migration criteria is built based on the challenges and drivers that appear when migrating BI application systems and which affect the criteria in different ways and degrees of importance. This fuzziness related to the cloud migration criteria increases the complexity of the decision making. The proposed concept uses security and synchronisation mechanisms, in addition to the partial migration, in order to protect the extra sensitive BI data and to keep its consistency.

Seven lessons can be learnt from the analysis of the effects of each cloud migration alternative recommended for each particular BI cloud migration project. Additionally, the lessons learnt are discussed in the context of how they relate to the requirements explained in Chapter 5 guided by the research frame of reference (as explained in Chapter 4). These seven lessons learnt can be reused in the future in other cloud migration projects with similar characteristics. The main characteristics of the migration of BI relate to its complexity as multiple software components run on the premises of organisations or on cloud environments. The conclusions of these evaluation could also be extrapolated to other application systems that want to benefit from a multi-device environment with fault-tolerance built in cloud-based replicated data infrastructures and with data synchronised across the cloud-based architecture. In addition, cloud migration projects with organisations or enterprises assigning a lot of weight to cloud migration criteria related to cyber-security, privacy, and synchronisation can reuse the lessons learnt via this evaluation scenario. The migration drivers and challenges described below fall into the category of the cloud migration criteria element of the research frame of reference[623]. These criteria are modelled with the use of the proposed concept as the main decision-making factors to select the cloud migration alternatives for the exemplary BI migration scenarios.

Cloud migration criteria: migration drivers

Multiple technical, operational, and financial cloud migration criteria motivate the adoption of cloud computing within BI application systems. The following are six specific cloud migration criteria that are modelled with the proposed concept given their importance to decide how to select from the aforementioned exemplary cloud migration alternatives for BI application systems.

Reduced cost: organisations can convert part of their capital expenditure to operational expenditure through the use of on-demand measured services. Cloud environments let BI organisations increase their financial flexibility by letting them move from capex to opex costs to benefit from the usual cost effective and pay-per-use pricing models that cloud providers offer. In sum, BI organisations lower

[623] See Section 4.7 in Chapter 4 to learn about the fifth element of the research frame of reference, cloud migration criteria

the Total Cost of Ownership (TCO) of their application systems as they move them to cloud environments. Cost belongs to the fundamental and reusable cloud migration criteria modelled with the proposed concept with optional layers of criteria in the context of the fifth element of the research frame of reference.

Broad network access: cloud BI users access relevant BI information from everywhere using any kind of thin and thick clients running on workstations, laptops, tablets, and mobile phones. BI users can remotely access cloud storage with raw data and information or indicators based on those data —such as Key Performance Indicators (KPIs) or Key Risk Indicators (KRIs)— and extracted with well-tested BI tools that run in cloud environments. The research frame of reference also prompts the inclusion of these criteria into the reusable decision-making cloud migration criteria.

Elasticity and increased computational, storage, and networking power: BI application systems can easily scale up and down to accommodate the demand of resources as well as they can use serverless deployments. BI application systems can access large amount of resources that cloud environments pool to let BI organisations perform computing- or storage-intensive tasks when needed. Specifically, the logic layer within common three layered BI architectures can experience a strategic advance resulting from elastically using the apparently infinite resources to run computing-intensive tasks by using advanced analysis components.

Lesson learnt 1: to include in the templates of cloud migration criteria particular sub-criteria specific to BI application systems. Supported requirements: *R5–R9*.

Outsourced configuration, administration, development, and updates to cloud environments: cloud consumers can use cloud environments to more easily and frequently update the code base of BI application systems while managing the execution environments external to their clients. BI application systems' developers have the potential to increase their implementation speed since cloud providers make their execution environments immediately available to them without any external dependence for infrastructure procurement or application system deployment. Cloud providers will usually supply additional capabilities for execution and storage and to support developers to run and orchestrate their containers.

Additionally, providers assist developers in the new DevOps cross-functional mode of working by offering toolchains so that they can shorten the development life cycle and potentially provide continuous delivery. Cloud providers usually offer better security standards, back-up application systems, and the assistance of IT professionals, than on-premises clients. Hence, BI application systems using services outsourced to cloud environments have got the potential to increase the quality of these deployments and organisations using them can now focus on their core business and do it better.

Transformational outsourcing: cloud consumers can add new capabilities that cloud environments offer or structural changes —such as new analytical capabilities— to enhance BI application systems. These capabilities are usually highly standardised, non-core, demanding parts of BI application systems, which can be consumed on demand.

Collaboration: BI application systems' providers can supply added value by integrating information from different clients —such as benchmarks, industry reports, or analytics across enterprise borders— and by making that data accessible from everywhere and to every kind of device. This enhances communication and offers constantly updated BI data.

Lesson learnt 2: to model, with the cloud environments modelling module integrated in the proposed concept, the BI-specific services (such as services to facilitate configuration, (non-)transformational outsourcing, or collaboration) that cloud environments offer to BI organisations. Supported requirements: *R1*, *R2*, *R4*.

Cloud migration criteria: migration challenges

The proposed concept considers the concerns of organisations when adopting cloud environments to run their BI application systems and how to cope with them.

Data security: one of the usual highest-ranking problems associated with cloud migrating BI application systems. Data security includes concerns related to data confidentiality, privacy, and integrity. Hindrances that, in some cases, seem impossible to overcome because of trust issues. However, as cloud computing matures, as cloud providers become more established and implement more advanced security mechanisms, they provide increasingly secure environments. One could argue that cloud environments are even more secure than the local computing environment of some private organisations. In the long term, this might make BI organisations to even regard cloud providers as more trustworthy than their own computing environments. The proposed concept can benefit, therefore, from implementing mechanisms to secure the migration of an application system to cloud environments in addition to security mechanisms to keep the data safe once migrated.

Lesson learnt 3: to integrate (end-to-end) security and synchronisation mechanisms in the proposed concept to guarantee trust in the cloud providers and in the cloudified BI application systems. Supported requirements: *R5–R6*, *R12–R13*.

Trust and cloud provider maturity: in addition to the data security perceived, cloud providers maturity also affects the trust that BI organisations deposit in them because of their reliability, availability, and Quality of Service. Decision makers are reluctant to move their BI application systems to immature and non-trustworthy cloud environments after performing careful risk assessments. They do not know whether the provider will go out of business, which would force them to migrate to another cloud environment. To cope with this problem the proposed concept ranks the cloud migration alternatives that make the target application system more portable higher. Cloud migration alternatives increase their portability when they comply with the Cloud Data Management Interface (CDMI) or when the target application system are deployed to more than one single cloud provider. If a cloud provider went out of business, the application system would still run on a different provider or be ported to it. This way, higher levels of availability can be offered and the proposed concept can be used to port the BI system to an additional cloud environment configuration.

Data ownership: accountability calls for clearly specifying the organisation responsible for the data once the BI data storage is cloud migrated. This way, data ownership risks are identified and measures can be taken to minimise them and assign liability. For example, BI organisations could define data management policies, if their cloud environments went out of business or if their contracts expired. The proposed concept supports designing the deployment of BI application systems to multiple cloud providers to cope with these concerns. In addition, the concept ranks higher those cloud providers offering transparent contracts to negotiate with BI organisations.

Unspecified SLAs and policies: the proposed concept design calls for transparent contracts with cloud providers to overcome this challenge. This way, the concept ranks higher those providers offering transparent SLA, legal, privacy, security, and regulatory policies.

Confusing pricing models: some cloud providers do not clearly state how they will charge for their services and what the final price will be. This hinders organisations when deciding to which cloud environment to migrate their BI application systems. Cloud consumers should be able to clearly assess the economic impact of the migration and show the potential to increase the return on investment (ROI). In addition, the lack of standardised pricing models makes it difficult for BI organisations to compare cloud offerings. The proposed concept includes the cloud environments modelling module that allows for clearly comparing them and for specifying their attributes, including their costs.

Lesson learnt 4: to allow organisations to transparently model cloud environments explicitly defining responsibilities through contracts that delimit data ownership and use standards for cloud resources pricing. Supported requirements: *R4–R7*.

Many Cloud offerings and levels of granularity: in addition to the three usual cloud service levels (IaaS, PaaS, and SaaS), there are different levels of granularity in which cloud providers configure (and let configure) their services. The proposed concept takes into account the particularities of the different cloud offerings configurations to accurately assess the performance, security provision, and even the applicable legislation of the cloud-migrated BI application systems. That is another reason for the proposed concept to enable the clear definition of cloud offering attributes with sufficient level of detail by leveraging the cloud environments modelling module.

Lesson learnt 5: to allow freedom to model cloud environments in different levels of granularity with the use of the cloud environments modelling module. Supported requirements: *R1*, *R2*, *R4*.

Performance erosion: a BI application system might experience increased delays or latencies after the cloud migration, in spite of the intended performance enhancement. Firstly, the data migration requires bandwidth usage, which could affect the performance of the overall application system. Furthermore, the re-distribution of software components among different computing nodes could negatively affect the overall performance if, say, the proposed concept placed a component far from another one with which it frequently and heavily interacts. In that case, performance could erode due to the introduced wide-area communication between components that used to be in the same location. In order to alleviate this problem, the proposed concept assesses the component placement by taking into account coarse-grained performance assessments based on metrics for the cloud migration criteria. Finally, the inability of service providers to properly scale up can cause performance issues as well. The proposed concept assists organisations in selecting the cloud service provider appropriate for their BI application systems and, accordingly, adapt their architecture.

Lesson learnt 6: to let organisations model, in a sufficient level of detail, their BI application systems prior to the cloud migration and to assist them in the specification of metrics to assess performance after the migration. Supported requirements: *R3*, *R8*, *R9*.

Integration: the majority of BI landscapes come in multi-layered architectures of interdependent software components collecting raw data from many sources to extract meaningful information. The proposed concept helps in designing cloud migration alternatives that incrementally cloud migrate BI software components evolving from an on-premises solution to an increasingly cloudified architecture.

Operational BI integrated with other application systems: BI application systems often rely on (near) real-time operational data that is gathered by enterprise application systems, such as: CRM, ERP, SCM, or marketing management systems. Integration is key for operational BI, as it stems from the unification of strategic and tactical decision-making. This way, BI is integrated into frequently changing business processes working in (almost) real time. When moving BI application systems to cloud environments, their dependencies have to remain intact in order to support the adequate functioning of operational BI. However, this cannot happen at the expense of the provided Quality of Service. BI organisations arguably benefit from the assistance of the proposed concept to strike a balance between the additional distance between components and the intensity of the cloud migration. The proposed concept assists in designing an architecture that might cloud migrate software components, such as an ERP system, together with the BI application system to avoid performance erosions due to the introduced geographical distance. The assessment of performance erosion caused by the integration of software components considers the data and computing geo-location and the wide-area communication they cause.

Lesson learnt 7: to perform step-by-step cloud migration to create partially migrated BI ecosystems (that also provide for the integration needs of operational BI application systems) while preventing performance erosion. Supported requirements: *R1–R3*, *R5–R6*, *R8–R9*.

7.3.5 Evaluation-based cloud migration concept refinement

The seven lessons learnt facilitated iteratively refining the proposed concept and prototype by adjusting it to the findings of the BI evaluation scenario. The challenges experienced, when applying the proposed concept to this evaluation scenario, helped revisit the requirements, in Chapter 5, so as to improve the design and implementation of the proposed concept to focus on security provision and synchronisation in cloud-enabled BI application systems. As a result, the proposed concept is able to assess the inclusion of security and synchronisation modules, as shown in Figure 7.19, to rank higher those cloud migration alternatives that include software artifacts that increase security and able to synchronise data sources. That is, given their importance for this evaluation scenario, the proposed concept has been refined by focusing on the third and seventh lessons learnt.

The concept refinement proposes cloud migration alternatives that use security and synchronisation modules to increase the trust of BI organisations in cloud deployments in partially migrated BI landscapes.

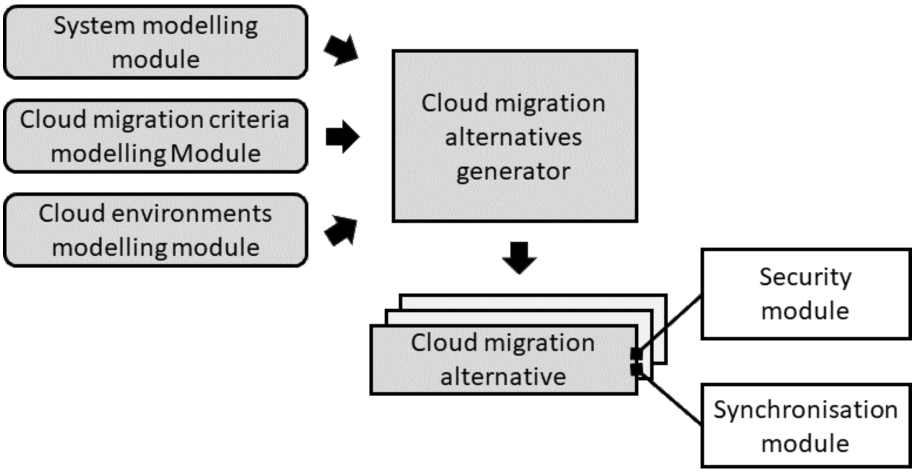

Figure 7.19: Prototype of the cloud migration DS concept for business intelligence application systems[624]

Security modules

Some of the cloud-based BI application systems intend to use a security module to ensure end-to-end encryption to avoid compromising the data at the moment of the cloud migration. Some data cannot leave the premises of the client of the BI application system, due to law or policies on sensitivity, even if the BI tools running on cloud environments need them. Therefore, the cloud migration alternatives that integrate some sort of security module like the one described above are ranked higher as the module is used to substitute sensitive data elements with a token that represents the sensitive or secret data, which are the subjects of the performed operations. The module strives for security with tokenisation[625]. When, as an example, a user accesses client data with a tablet over a non-trusted connection, some of the cloud migration alternatives offer a security module that tokenises the original sensitive personal data. A person's Social Security Number could never leave the premises local to the client and is replaced with randomly generated values instead. The module keeps the original format of the data and preserves the functionality running on the cloud premises by translating from token to real data at the cloud premises. This approach is compatible with the proposed use

[624] Own illustration based on Juan-Verdejo and Baars (2013), p. 46
[625] cf. Bomar and Harper (2019), pp. 4–5; Cutillo et al. (2012), p. 1

of encryption to guarantee data residency for some cases while encrypting data for others according to the requirements. The proposed concept assesses the different cloud migration alternatives according to the security mechanisms they use.

As the previous discussion explains, growing privacy and sensitivity concerns hinder both data sharing and integration. The proposed concept takes them into account in combination instead of independently. The concept ranks higher those cloud migration alternatives that use a security module that annotates data with privacy-related metadata to specify the applicable privacy policies, which move along with the data to which they refer and change as those data do.

The proposed concept ranks cloud migration alternatives not migrating sensitive data to non-trusted cloud providers. These might data stay at the local premises of the BI organisation according to the data access rules, which depend on the roles of the BI users and the business process in which they participate. As a result, users get different levels of authorisation to access, manipulate, and store depending on which data those operations are performed[626].

Synchronisation modules

Some BI cloud migration alternatives in this evaluation scenario use a synchronisation module to track the state of records and their relations to transfer data that has changed[627]. At the beginning of a synchronisation session, the module obtains the state from each of the two synchronised application systems and compares them with the state known from the last session.

The connection bridge module, as Figure 7.20 shows, controls the communication between a synchronisation module and the BI cloud-based application system, which is the target for synchronisation. To ensure an independent and reusable implementation, the bridge encapsulates communication technology between the application system and a synchronisation module. As the connection bridge has no domain model-based dependencies, the same implementation (of the required methods for the synchronisation process) works for different application systems

[626] cf. GloNet Consortium (2012), p. 32

[627] Figure 7.20 shows how some cloud migration alternatives integrate a synchronisation module, which builds on top of a previous work: cf. GloNet Consortium (2012), p. 33

that support the implemented communication technology. The synchronisation module leveraged in some of the BI cloud migration alternatives uses mapping and link definitions to work with the differences in the domain model structure of various application systems. The new application system, the cloud migrated BI system, needs to implement the interfaces of the connection bridge (as connection bridge A and B do in Figure 7.20) to fulfil the requirements of the synchronisation process and, hence, be able to use it. This intends to reduce the cost of supporting multiple external application systems.

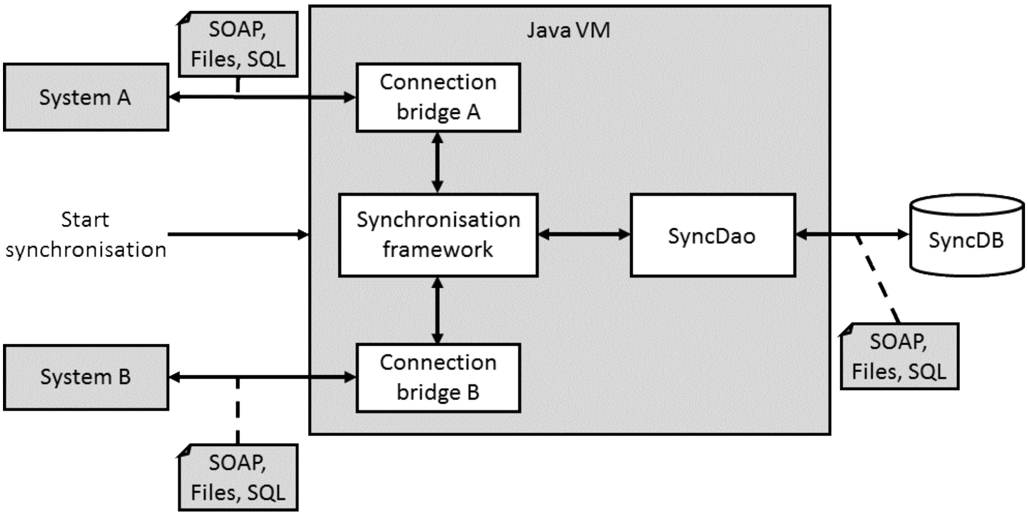

Figure 7.20: Description of the implementation requirements for synchronisation sessions[628]

Synchronisation modules, like the one used by some cloud migration alternatives, manage the workflow of the synchronisation session and the communication to other connection bridges. Likewise, it contains the logic of the synchronisation to determine the states of the records that the module has to synchronise. Based on the record pair state and the selected sync strategy, the sync operations are created. A synchronisation module arguably delivers updated data in a transparent manner and enforces their availability by defining mappings and link definitions in configuration files.

[628] Own illustration based on Juan-Verdejo and Baars (2013), p. 47

7.4 Scenario 3: cloud migration of collaborative networks systems

This evaluation scenario revolves around the architecture, concepts, and steps needed to adapt collaborative networks application systems to a cloud-based deployment. The cloud-based platform facilitates online collaboration by letting organisations deliver and operate service-enhanced products (SEPs) by leveraging collaborative networks during their entire product life-cycle. Networked organisations share business operations among them as well as resources, competences, and knowledge. Organisations can create this way new competitive environments enabling them to enhance their ability to quickly respond to market demands and therefore increase their business agility. One of the key challenges in establishing effective networks for collaboration to operate the service-enhancedproducts that they offer lies in its complexity stemming from the wide variety of business operations that organisations offer by using multiple involved resources. Collaborating organisations, if they have access to the complete shared cloud-based view of their business ecosystem, can analyse how working with multiple stakeholders affect their products' business life-cycle. Collaboration alters their global performance and cost-versus-quality effectiveness while changing their emerging behaviours, business agility, and resilience. Software solutions that support this sort of collaboration come with effort-intensive design changes to let collaborative application systems cost-effectively run on cloud environments while being ready to be tailored to multiple scenarios presented by organisations and members of the involved networks.

This evaluation scenario closely relates to three scientific papers published in three peer-reviewed international conferences that document the evaluation and two technical reports[629]. The cloud migration concept help organisations design the cloud-enabled platform for collaborative systems to build highly customised service-enhanced products and operate them in collaborative networks. The transition to highly customised products in manufacturing can even reach the

[629] Several papers co-authored by the author of this dissertation document this evaluation Surajbali et al. (2014b), pp. 1–8; Surajbali et al. (2014a), pp. 1295–1299; Camarinha-Matos et al. (2015), pp. 185–190; GloNet Consortium (2015a), pp. 1–43; GloNet Consortium (2015b), pp. 1–30

point of delivering one-of-a-kind products; that is, mass customisation[630]. As the creation of these products usually requires a range of skills and resources that may not be readily available within a single organisation. Organisations are therefore incentivised to collaborate to deliver service and products. In fast changing market conditions, multiple organisations can build up dynamic virtual organisations to pool their resources, competencies, and services to quickly react to new collaboration opportunities[631]. Enterprises[632] benefit from using cloud environments by using virtual collaborative alliances that employ distributed value-added services to help enterprises overcome obstacles to collaboration. That is, barriers to dynamic and regular collaboration that could hinder the creation of business value on a global scale through offering their products and services.

This scenario evaluates the benefits for organisations of being assisted by the proposed cloud migration DS concept to design the cloud migration of a collaborative platform for networked organisations[633] and assesses the effects of running its different collaborative mechanisms in cloud environments. The cloud-based collaboration platform helps collaborative networks —of manufacturers, customers, and local suppliers— to support and deliver complex, highly customised, and service-enhanced products. The motivation to run this collaborative platform on cloud environments is to better support co-creation and mass customisation of service-enhanced products (SEP) with enterprise networks working across the entire life-cycle of those products. The targeted collaborative system provides multiple functionalities for virtual organisations to form goal-oriented networks or new collaborative networks, co-design complex products, or collaborate in offering services or products. In addition, other functionalities help organisations share competences and knowledge to undertake maintenance tasks in those SEPs[634].

[630] Mass customisation is a co-design process driven by the customer to enable products and services to meet the unique needs and preferences of each individual customer with regards to desired product features

[631] cf. Taleb et al. (2017), pp. 1657–1662; Stergiou et al. (2018), pp. 964–975; Alonso et al. (2017), pp. 400–403; Graça and Camarinha-Matos (2019), pp. 254–257; Camarinha-Matos et al. (2008), pp. 15–20; Noran (2004), p. 71

[632] In particular, glocal enterprises benefit from the use of cloud environments to collaborate. *Glocal* refers to enterprises that are able to work with local clients and stakeholders because they know the local context but at the same time act in a global manner by forming virtual organisations

[633] cf. GloNet Consortium (2016), p. n/a, URL see Bibliography

[634] cf. Graça and Camarinha-Matos (2019), p. 257; Camarinha-Matos and Afsarmanesh (2007), p. 119; Camarinha-Matos et al. (2008), pp. 15–20

7.4.1 Evaluation scenario setup

The last evaluation scenario (see the research frame of reference[635]) focuses on how the proposed concept assists organisations to migrate their collaborative networks system and entails designing the migration of the collaborative networks platform and its sub-systems. These sub-systems implement collaborative concepts, models, and methodologies. The CAS Open cloud environment was selected as a design constraint to migrate the application system to it. This is due to the ability of this IaaS environment to ensure the availability of application systems running on it as well as the the expertise of the organisation in developing software for this computing environment. The cloud migration concept assists the migration to CAS Open according to the architecture of the collaborative networks system and their set of cloud migration criteria.

Motivation

The selection of collaborative networks systems as the third evaluation scenario is motivated by the fact that one of the advantages of cloud-enabling application systems strives in the potential to improve collaboration. A very relevant cloud migration criterion for multiple organisations. In this evaluation scenario, the focus lies on advanced collaboration concepts in multi-stakeholder networks. Collaborative networks systems represent cloud migration projects with a strong focus on cloud-based collaboration applied to complex business processes that act during a relatively long period of time. The conclusions of this evaluation scenario are abstracted in order to open the door to applying them to the migration of other application systems that greatly benefit from cloud-based collaboration. That is, application systems like those that offer cloud storage, cloud database, or distributed collaboration; in addition to other application systems offering collaborative office automation, e-mail, teleconferencing, social networks, different document editing suites, or presentation software.

This evaluation scenario is selected given the relevance for SMEs and larger organisations of the collaboration cloud migration criterion. Cloud-based collabo-

[635]The cloud-based collaborative networks system is, as Figure 7.1 shows on its right-hand side, the third element of the research frame of reference. It focuses on cloud-enabling collaborative networks systems and how the proposed concept assists organisations in migrating such systems to cloud environments

rative networks systems belong to the category of cloud collaboration application systems acting in a scenario as complex as the kind of application systems that might potentially need the assistance of the proposed concept to be migrated to cloud environments.

Evaluation method

Two workshops are designed to let eighteen respondents evaluate the functionalities, architecture, and properties of the collaborative networks system after cloud migrating it based on the design proposed by the cloud migration concept. A workshop with iPLON and Prolon —the main end users, leaders in the solar and building automation business fields respectively, who were closely involved in the requirements gathering and cloudification of the collaborative networks system—- to present them with questionnaires to assess how effectively the cloud-enabled collaborative networks system fulfils their cloud migration criteria and those of their networks. They assessed the architecture of the selected cloud migration alternative and the associated changes with respect to their cloud migration criteria. Additionally, another evaluation workshop in Chennai (India) have sixteen members of the iPLON solar energy collaborative network contribute by responding, after the training session, to questionnaires about the cloud-enabled collaborative system in comparison to the on-premises functionalities. They are mainly companies that provide services to iPLON to manage and operate service-enhanced products.

All eighteen participants have prior experience with the on-premises application system they used to co-create, co-design, and operate complex products without modern cloud-enabled collaborative concepts. During the workshops they were explained how the cloud-migrated collaborative networks system works and its related concepts/functionalities in addition to seeing the system in action and using it. Based on that experience, they assessed different aspects of the cloud-enabled evaluation scenario on a Likert 4-point ordinal scale as: 1) inadequate, 2) limited, 3) good fit, or 4) very good fit. They were assigned the task of operating service-enhanced products by using the collaborative networks platform once migrated with the cloud migration concept. They serviced complex products by using the cloud-migrated collaborative tools —such as the product-service specification tool, the product portfolio, and the virtual organisations management tool—, the cloud-migrated collaborative networks system, and their now migrated collaborative functionalities (functionalities with which they have worked in on-premises deployments) applied to their fields of expertise. The participants reflect on the problem at hand, perform hand-on operations with the cloud-enabled system, and fill in the questionnaires. The questionnaires are used to gather feedback to extract subjective and objective measurements in addition to being designed to collect several responses to open-ended questions. Additionally, informal discussions about the cloud-migrated collaborative networks systems followed.

The two organisations are the leading organisations of their collaborative networks —iPLON and Prolon— have a more prominent role in the evaluation as they beta-test the iteratively applied cloud migration alternatives suggested by the proposed concept and provided feedback to the migration process to the CAS Open cloud. Their input form a knowledge base which is employed to model, with the help of the proposed concept, the on-premises collaborative network systems, the collaboration functionalities, and the cloud migration criteria of iPLON and PROLON. In addition, their input guide the selection of one cloud migration alternative for its implementation as well as helping to manually fine-tune the cloud migration alternative according to the feedback of iPLON and PROLON. Their input is not needed to use the cloud environments modelling module because the CAS Open cloud environment configuration has already been selected and modelled as part of the input for the cloud migration decision-making process. The CAS Open cloud is a design decision; it is the cloud infrastructure that the involved stakeholders selected prior to the cloud migration project. Next, the cloud migration alternatives were implemented in two pilot demonstrators for intelligent buildings and solar energy. Based on that, the two leading organisations analysed how fit for purpose the cloud-based deployment is for running their different collaborative networks to service their complex products during their entire product life-cycle. Both pilots have two different perspectives as iPLON operates solar energy service-enhanced products and networks, while Prolon services intelligent building automation SEPs in collaboration with its networks. In addition to improving the collaborative networks concepts, the cloud migration also added new cloud-based functionalities. The evaluation in the case of the cloud-based solar energy collaborative network is more comprehensive as the additional sixteen respondents belong to the network partners of iPLON.

Core evaluation aspects

The evaluation of the proposed concept via this scenario focuses on three aspects. One aspect entails evaluating the support that the cloud migration DS concept offers to recommend a cloud migration alternative. Specifically, the evaluation scenario analyses the application of the proposed concept to cloud migrate the operation of the collaborative networks of two leading organisations working on the solar energy and building automation business (iPLON and Prolon). The second aspect entails analysing to which extent the cloud-enabled collaborative system improves the collaboration of sixteen partners of the iPLON solar energy collaborative network in the co-creation, co-design, and operation of solar energy service-enhanced products. The third aspect assesses the degree of fulfilment of the organisations' cloud migration criteria with the use of collaborative functionalities[636] migrated to the CAS Open cloud environment configuration and which cloud migration criteria are of particular relevance for this evaluation scenario.

7.4.2 Cloud migration project modelling

The proposed cloud migration concept assists in selecting the most appropriate cloud migration alternative for the collaborative networks system according to the architecture of the on-premises collabrorative networks system and the cloud migration criteria modelled with the help of the two leading organisations[637]. The cloud migration of the collaborative networks system was subject to a design decision dictating the use of the CAS Open cloud environment configuration according to organisational decisions. The following describe the on-premises application system model and the cloud migration criteria modelled as input of the proposed cloud migration concept. Finally, the selected cloud migration alternative is described.

[636] Collaboration functionalities to let organisations manage virtual breeding environments (VBEs), to improve the operation of their service-enhanced products, and to co-create their complex products

[637] iPLON and Prolon are the two leading organisations, in the the solar and building automation networks respectively, closely involved in the requirements gathering and cloudification of the collaborative networks system

On-premises application system model

Emerging from an available virtual breading environment (VBE)[638], dynamic collaborative networks jointly design, engineer, and manufacture the physical product. Likewise, they collaboratively design and execute the integrated business services that multiple suppliers provide. The members of collaborative networks require that the cloud-migrated collaborative networks system and its underlying collaborative platform provide the eight functionalities previously offered on-premises.

Functionalities to set up collaborative spaces. Collaborative networks systems let organisations select VBE members and create virtual organisations (VOs) to collaboratively offer service-enhanced products. For this collaboration to happen, the collaborative platform should let virtual organisations share competences and knowledge —such as documents, specifications, or emails— among themselves and new VBE members. Furthermore, it must allow VOs to create and dissolve their goal-oriented collaboration networks with different VBE members.

Functionalities to support role-based access in collaborative networks. Different roles are allowed particular permissions to access information and knowledge in order to protect the confidentiality of the information shared in VOs. Some VOs members collaborating in a goal-oriented network (VO) could find themselves competing to achieve other goals and use that gained information. The collaborative networks system should grant VO members access to resources fitting their particular roles. In addition it should be able to allow the identification and management of property rights for each resource.

Functionalities for data and knowledge sharing across VOs. In order to effectively collaborate, VO members need to share information on products and services in addition to software tools or lessons learned. The collaborative platform system must let organisations share resources and help them to search, retrieve, and update information and knowledge —templates, standard processes, or ontologies— within VOs as well as implement policies to encourage organisations to share.

[638]cf. Graça and Camarinha-Matos (2019), pp. 254–257; Camarinha-Matos et al. (2008), pp. 15–16

Functionalities for undertaking tasks related to groupware. These are functionalities that allow VO members to manage the tasks needed to attain the goals of their projects and to set appointments among them.

Functionalities to keep data consistency between collaboration spaces and existing information systems. VO members and their own systems share products and services. Therefore, a goal-oriented VO collaboration space should consistently synchronise information across the collaboration space and the information system. For instance, if a bidding document changed in a collaboration space, the platform should update the document in the VO information system to ensure data consistency.

Functionalities to maintain knowledge of the VO during its life-cycle. These functionalities support VOs to collaboratively design their services and products and facilitate the appearance of solutions increasingly innovative. The platform should let VOs define workflows and information that they will maintain throughout the entire life-cycle of the VO.

Functionalities to ease the access to collaborative networks. The platform should let VOs participate in more than one collaboration and transparently switch between the collaborative network easily without logging in again every time they shift from one collaboration space to another one.

Functionalities to implement interoperability. VBE members will have different information systems —such as ERP and CRM ISs— that should seamlessly interoperate with the collaboration network platform. For example, VBE members might require exchanging offers of their products and services over time among themselves and with the collaborative network.

Cloud migration criteria

The cloud migration criteria cover the essential design principles and important dimensions driving how to migrate the collaborative networks system to the CAS Open cloud environment. The cloud migration of the collaboration system as a centralised platform allows any authorised collaborator to access it over the Internet from anywhere, at any time, and using any kind of device.

Reduced initial costs and flexible cost models. Running the platform on cloud environments provides cost reduction from which small and medium enterprises or organisations particularly benefit. They reduce costs as they do not own the computing capabilities —computing, networks, and storage resources— for which they pay. Cloud flexible cost models let organisations use the pay-per-use model therefore avoiding investing large amounts upfront in software or hardware equipment. Organizations that opt for the cloudifying their systems for maintaining collaborative networks pay fees on a monthly or yearly basis according to resources they consume or the number of user licenses they require.

Internet-powered ubiquitous and simple access to the collaboration infrastructure. With the use of cloud environments, companies can eliminate geographical obstacles and promote dynamic collaboration on a continuous manner. This leads to creating business value composed globally via virtual partnerships. Any virtual organisation can gain access to the central collaboration platform running in cloud infrastructures as if it were an integration hub. This way, end users may access collaboration functionalities over the Internet via web-based user interfaces or web services with programmatic access.

Modularity and extensibility. Cloud-based software solutions run a consolidated code-base of a software sub-system that works for all the consumers. However, provider and user organisations have to adapt their collaboration platform to be run in cloud environments while accommodating the requirements of collaborative organisations. That could entail supporting customisable features that are easy to extend. These features could include configurations and data models tailored to end users and their workflows and business rules. This enables organizations to easily add or remove modules to fit their specific needs.

Multi-tenancy. A software system able to work for multiple tenants (or customers) comprising multiple users is a multi-tenant system[639]. Multi-tenancy accommodates two potentially conflicting requirements of cloud-based solutions to offer a consolidated architecture while customising the software solution to the needs of the user organisation. The common architecture handles all customers uniformly

[639] Chapter 4 explains multi-tenancy more in detail in Section 4.6 under the technical cloud migration criteria sub-section

and reduces cost by delivering software that scales as economies of scale. At the same time, organisations need software that matches their requirements and highly individual business processes. Hence, both data and customisations should remain isolated per tenant on the cloud-based collaboration platform while the system should synchronise data across collaborating organisations.

Scalability and availability. Any cloud-based collaborative environment, like the target system to enable collaborative networks to work, arguable need a scalable architecture able to concurrently handle thousands of users. That is, the number of customers that have licensed the software multiplied by the average number of users per customer. To ensure its smooth functioning, the system requires the ability to scale up to servers with either additional processing power and RAM or with faster storage. Likewise, these systems need to scale out with typically load balancers distributing the workload of the collaborative networks system among several computing, network, or storage resources identical to the original one. Additionally, the ability of scaling up and out has to come hand in hand with their opposites to scale down and in. Application systems used in collaborative networks adopt the use of standard and affordable computing resources instead of costly high-end servers. This leads to increased availability as multiple instances of the same application server can be deployed.

Trust and security. Data security[640] is one of the main issues that organisations have to overcome to adopt cloud environments to run their systems and, specially, their collaborative networks systems. This is because the collaboration platform needs security mechanisms that strike the right compromise between security and data sharing among collaborating organisations.

Selected cloud migration alternative

Based on the on-premises application system model and the cloud migration criteria, the proposed concept generates multiple cloud migration alternatives using the CAS Open cloud environment configuration. This section explains the cloud migration alternative suggested by the proposed concept for the collaborative networks system and fine-tuned with the help of the two leading organisations.

[640] Section 4.6 in Chapter 4 explains data security in depth

The collaborative networks application system still follows the three-layered architecture once cloud migrated as Figure 7.21. This way, each layer or tier can be independently developed, maintained, and deployed. The presentation layer uses a graphical interface to provide provides user interaction displaying information while gathering user commands and input. Meanwhile, the business or application logic layer executes the core businesss processes and operations. The data layer provides data storage and access to persistent data, usually stored in a database management system. The figure shows the interplay, through the interface layer, between the collaborative networks platform, the collaboration space, and their integration with the six additional modules. These modules implement functionalities for single sign-on, user permissions, synchronisation, service-enhanced products (SEPs), process workflow, and VBE and VO management. In addition, Figure 7.21 shows the module for the integration with the external systems on the right-hand side. The proxy-based and mash-up integration are two alternative mechanisms that mediate between the interface layer and the external systems. The external systems might be an ERP or CRM system as well as other systems specific to the customised scenario. The mediation with the interface layer allows for connecting such external systems to the eight software modules of the cloud-based collaborative networks system via the interface layer.

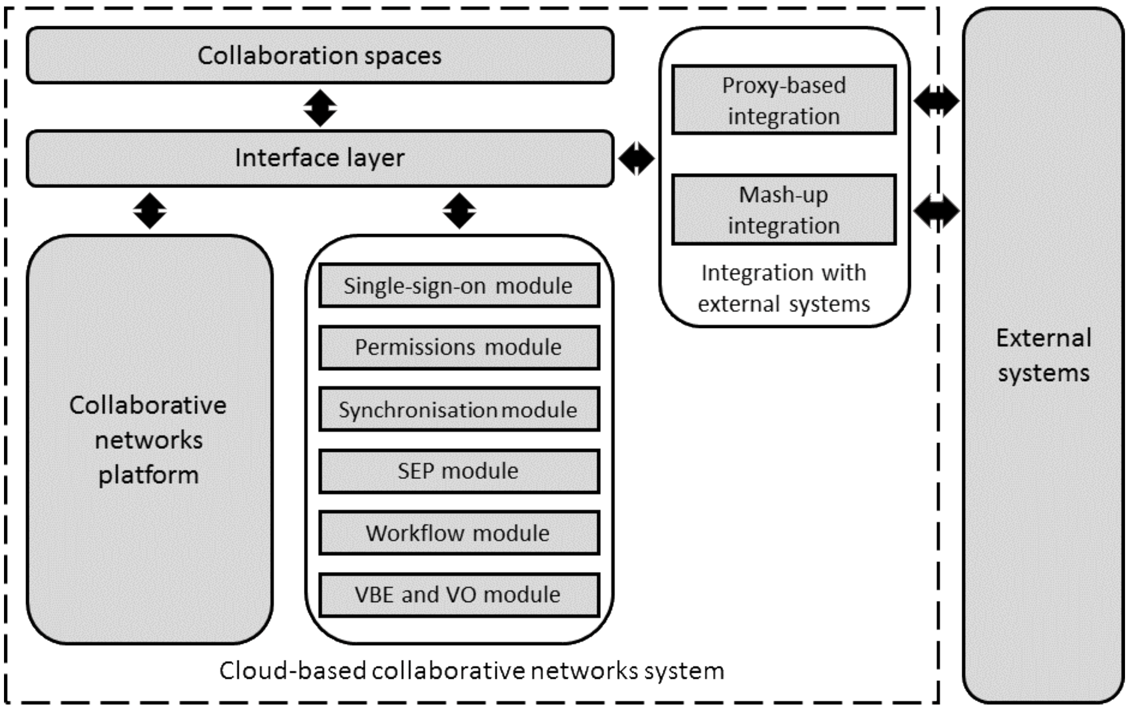

Figure 7.21: Cloud-migrated collaborative networks system[641]

The collaborative networks platform, on the left-hand bottom corner of Figure 7.21, offers the architecture of the application system, which includes components that implement the fundamental functionalities in components of the architecture. Organisations are offered mechanisms to customise the platform to extend it to their specific needs. To that end integration and extension mechanisms allow them to complement the features of the platform with additional services and modules. The cloud-based collaborative networks platform includes five basic modules contributing to the overall system. The data access module, the presentation module, and the publicly-defined interfaces to access the other two modules: the business operations module and the server core module. The business operations module implements basic services to manage users, tenants, permissions, documents, calendars, and customised operations added with the server core. The back-end implementation of the platform is encapsulated in the server core module and includes different mechanisms. Those mechanisms allow users of the platform to conceive modules providing new operations that they can register. They can also help in data objects management or user- and tenant-based access according to the context of the executed operation. The rest of the software components access these two modules with interfaces such as in-process method calls, remote method invocations (RMIs), and SOAP (Simple Object Access Protocol) or RESTful (representational state transfer) web services.

There are seven other software components and modules in Figure 7.21. The collaboration spaces module manages the collaboration spaces[642] that help groups of stakeholders so that they collaborate in order to achieve their common goal in both the physical and virtual worlds. Furthermore, the six modules on the right-hand of the collaborative networks platform are its fundamental components. The SEP module helps maintain, overhaul, or make service-enhanced products evolve. The synchronisation framework and the workflow module ensure proper collaboration among systems and stakeholders by enabling the integration of the collaborative networks platform with external stakeholders systems via services. Finally, the VBE and VO module help manage the long-term base networks so that

[641] Surajbali et al. (2014b), p. 3

[642] cf. Surajbali et al. (2014a), p. 1295; Surajbali et al. (2014b), p. 1; Camarinha-Matos et al. (2015), p. 185

its members can quickly form goal-oriented networks with the most suitable set of partners to collaborate in achieving the goal of the VO.

7.4.3 Evaluation results and interpretation

The evaluation results stem from the responses of two organisations —iPLON and Prolon— after the cloud migration of their collaborative network systems for networks involved in solar energy, in the case of iPLON, and in building automation, for Prolon. The two respondents provide input and work in the cloud migration of components of their collaborative networks system. Their aim is to improve how they collaborate with their networks in the fields of building automation and solar energy provision to offer service-enhanced products.

The two organisations adopt the cloud-based collaborative networks system adjusted to their needs after adapting some of their business processes and their systems software architecture. In addition to the feedback of these two organisations, sixteen network partners of iPLON participate in the evaluation in accordance to their role collaborating with iPLON to deliver service-enhanced products, solar plants, in the field of solar energy provision. Sixteen network members and iPLON provide feedback of the use of the cloud-based collaborative networks system to offer the complex and highly customised photovoltaic solar plant product over its long life-cycle of twenty to twenty-five years. The Charanka solar power plant in Gujarat (India) requires business services throughout its operation phase as well as for monitoring and maintenance —preventive, corrective, and condition-based maintenance— tasks that benefit from using more knowledge across the entire product life-cycle in a cost-effective manner.

Two leading organisations of the solar energy and building automation collaboration networks.

The collaborative work with the two leading organisations, iPLON and Prolon, to cloud migrate their customised collaborative networks systems help putting into play the cloud migration alternatives that the the proposed concept suggests. The subsequent evaluation workshop with these two organisations focuses on evaluating the fitness for work of the cloud-enabled collaborative networks system tailored to the needs of their collaborative networks and according to their field of expertise. This entails assessing how well the cloud-based collaborative functionalities and the cloud-based collaborative platform fit their purpose. Likewise, the two respondents are asked in questionnaires and informal discussions with regard to how well the adapted architecture and application system fulfil their cloud migration criteria.

The assessment of how fit the cloud-migrated collaborative systems of the two leading organisations are for their purpose, is extracted from the responses of the evaluators which rate the cloud-based collaborative systems from inadequate to very good fit through limited and good fit, as Figure 7.22 shows. The questionnaires that both iPLON and Prolon fill as well as the discussion with them show that they see a very good fit (4 Likert points out of 4: 4/4) of the cloud-based design with components and functionalities running on the CAS cloud environment to effectively collaborate with their networks. They both acknowledge the very good potential for information sharing due to using the designed cloud-based storage. Likewise, they value as a very good fit for their purposes the architecture designed with the proposed concept with regard to integration and multi-tenancy support (4/4). At the same time, they consider that the security provision, customisation support, reliability, and efficiency are a good fit (3/4) to their cloud migration criteria.

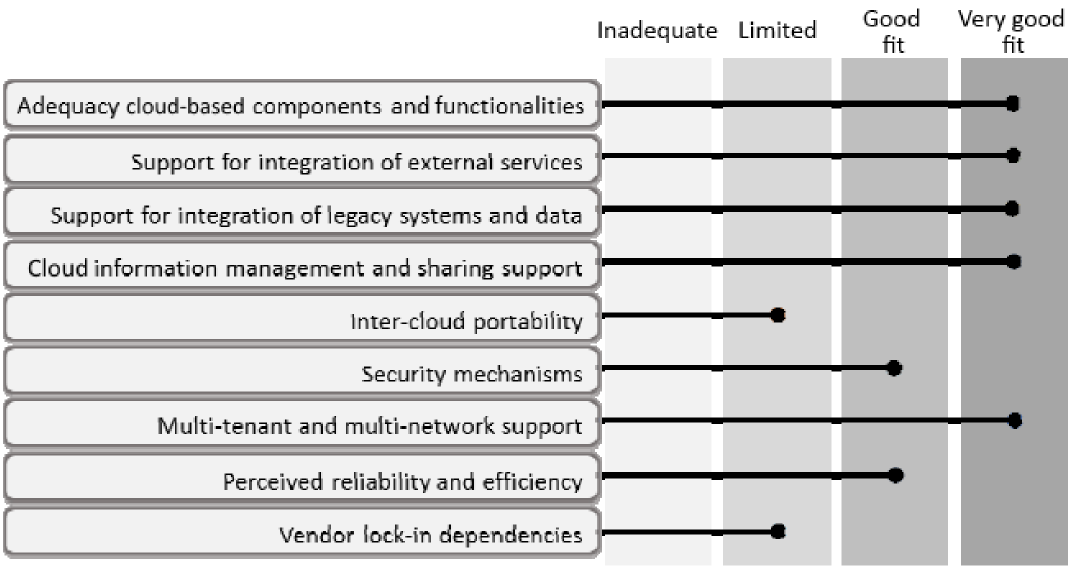

Figure 7.22: Evaluation of the cloud migration of the collaborative networks system[643]

[643]GloNet Consortium (2015b), p. 11

iPLON and Prolon find the proposed cloud migration alternative a good fit in the extent it fulfils their cloud migration criteria and a very good fit of the proposed cloud-based architecture. The two leading organisations rate as limited fit the functionalities to improve inter-cloud portability and to overcome vendor lock-in dependencies (2/4). Discussions with the two organisations shed light upon the fact that as the selected cloud environment was unique, the CAS Open, they could not the potential to showcase how the proposed concept designs the cloud migration for inter-cloud portability and to prevent vendor lock-in issues. The architecture is designed to use a single cloud environment rather than to migrate the collaborative system to multiple cloud environments. The limitations and challenges of the Platform-as-a-Service deployment model —namely, vendor lock-in and inter-cloud portability— make them fear that the lack of standards they adopt might increase their chances to be locked-in to a specific cloud provider.

The assessment of how well the adapted architecture and application system fulfil their organisational cloud migration criteria relate to collaboration requirements and how the use of cloud infrastructures enhance collaboration. The two organisations report that their collaborative networks and interdependencies work better in a cloud-enabled manner that lets all network members collaborate in the same centralised version of the data. In addition, they mention that cloud-based collaboration fits their collaboration criteria for the management of the entire life cycle of their solar energy and building automation products. Likewise, long-term and goal-oriented strategic networks can be more conveniently managed and improve how service-enhanced products are operated.

The two respondents found a very good fit how the proposed concept designed the cloud migration alternative to handle public and confidential information and exert access control on who can access what. This way, the network can access data stored in the cloud according to the stored sharing policies that let other network members use data access mechanisms. iPLON and Prolon value as positive that the network members and stakeholder's competences are stored now in the CAS Open cloud environment in addition to their roles and core values. As a result, as collaboration improves, long-term healthy solar energy and building automation VBEs could be used for years to let their participants join forces in trusted VOs.

Sixteen network partners of the solar energy organisation.

iPLON and sixteen partners of its partners of the solar energy network, which iPLON leads, participate in the final evaluation in Chennai (India) to evaluate the cloud-based collaborative networks system. They all exchange information on how their collaboration improves by implementing a cloud-based architecture. The feedback of the network partners is collected via questionnaires with four-star Likert scales and informal discussions.

After some short presentations of the cloud-based collaborative networks system tailored to the solar energy field, the iPLON network partners participate in a hands-on trial of the system to then evaluate the effects of the cloudification on the four key functionalities. The network positively evaluates the cloud-migrated functionalities to model sub-products, services, and complex product portfolios as they could then link product and services that are always in sync with real-time data that improve current monitoring. The network acknowledges the positive effects on collaboration of dynamically conforming cloud-enabled enterprise consortia operating at the levels of VBEs or VOs. Participating organisations can as a result collaboratively co-create and co-design working on the same set of data according to their access rights. The cloud migration of workflow monitoring functionalities, that Figure 7.21 shows, could in the opinion of the network facilitate business service execution whereby different organisations act at different stages of the service execution while directly sharing their data. The network members share that the cloud-migrated workflow engine is an appropriate mechanism to supervise the operation of their business services. At the same time,they advocate adding services for smoother distributed process execution and for alert and email notifications. Some network partners suggest adding remote communication services to improve the cloud-based collaborative systems. One network partner highly appreciates feature is the integration design of all the collaborative functionalities in a single system for fluid business processes and practices.

7.4.4 Discussion and evaluation lessons learnt

This discussion concludes by analysing the evaluation results and its interpretation with the aim of deriving lessons learnt during the process of cloudifying collaborative networks systems and via the discussion with the eighteen respondents. The following six lessons learnt abstract from the evaluation scenario and can be applied to other cloud migration projects in which organisations consider collaboration as one of the main cloud migration criteria driving the cloud migration decision. This evaluation scenario belongs to the category of application systems that might target adopting cloud environments to increment their collaboration capabilities: cloud storage, cloud database, or distributed collaboration for: office automation, e-mail, teleconferencing, social networks, different document editing suites, or presentation software. These are complex application systems that could benefit from being assisted the assistance of the proposed concept to be migrated to cloud environments.

The first lesson learnt abstracts from this evaluation scenario given that cloud collaboration tools and mechanisms are needed in order to operate and maintain service-enhanced products over their entire life cycle.

Lesson learnt 1: to integrate cloud collaboration tools in collaborative settings to let different parties collaboratively operate on updated data sets, product designs, and product portfolios with their software tools that do not run in data silos any more. Supported requirements: *R5*, *R10*

Small and medium enterprises (SMEs) encounter business opportunities to provide innovative business services that combine advances in communications, brought by cloud infrastructures, with sensor networks. They combine both to apply the Internet of Things to deliver intelligent supervision systems that leverage data mining and other support technologies. The cloud-enabled system for collaborative networks runs on top of the selected cloud environment —hence, CAS open— in a decentralised manner. Any authorised VO member can use cloud infrastructures to access the infrastructure at any time any device connected to the Internet. This way, SMEs experience increased ease of use and availability of resources.

For cloud-based collaborative network systems to integrate different highly specialised services with high quality, they need integration tools such as mash-up and proxy integration. The integration tools allow VOs to integrate their existing data and share them through cloud-based collaboration spaces. Knowledge and data such as documents, know-how, specifications, tasks, and historical information of products and services. The cloud-based collaborative network system needs to deliver secured collaboration that helps VOs maintain their information and knowledge with collaborative tools that interoperate with existing VO platforms using the integration modules.

The mechanisms to integrate external systems and services let cloud-based collaborative network systems providers adapt their systems to the cloud migration criteria of collaborative networks such as the solar plant and building automation networks. This helps in cases of goal-oriented VO-based collaboration around particular service-enhanced products.

Lesson learnt 2: to provide integration tools to include different services from different providers of, usually highly specialised, services to improve the migration to the cloud of collaborative networks systems.
Supported requirements: *R2*, *R11*

Best practices in virtual enterprise management recommend not compromising privacy for the sake of synchronisation functionalities. The effective management of the supply chains is crucial for complex production and operating units together with the sharing of knowledge and the functionalities to track processes involved in the delivery of complex products. Collaboration spaces enabled by cloud infrastructures may assist SMEs in their collaborative processes through services to operate and deliver service-enhanced products by managing their VOs. The synchronisation module[644] synchronises data between collaboration spaces and VOs while the collaboration space assigns rights and permissions to each VO member. A VO member can take part in multiple collaboration spaces according to their goals while joining and leaving the collaboration space over time. The cloud-based collaborative network system needs synchronising the data shared by the different sub-systems used to deliver collaborative design and operation of complex SEPs.

[644] Figure 7.21 shows the synchronisation module on the centre

Lesson learnt 3: to enhance advanced collaboration by synchronising data between collaboration spaces and virtual organisations with the collaboration space assigning rights and permissions to each virtual organisation.
Supported requirements: *R3*, *R6*, *R8*

The evaluation compared the preconceived idea that iPLON had before the pilot demonstrator regarding the benefits they could experience due to the use of collaborative networks system with the actual improvement they assessed after adopting the cloud-based platform. The organisations considered additional services acting on the collaboration spaces as a fundamental factor with added value that motivates it to adopt the new concepts and methodologies around collaboration and the cloud-based collaborative networks system.

Lesson learnt 4: to incentivise the adoption of cloud-based collaboration by supporting added-value functionalities (like the cloud-based collaboration spaces) that act on the shared data.
Supported requirements: *R2*, *R11*

The evaluation helps reach conclusions on how to improve the business processes and operations of cloud-based collaborative network systems to increase their commercial potential. Performance and collaboration indicators can help assessing this commercialisation potential of these systems. Performance indicators help comparing the needs for product development with financial and operational dimensions but also with concerns about environmental impact and energy consumption. The evaluation shows that iPLON experienced more benefits than expected from adopting the platform and that it is prone to move to the exploitation phase. iPLON assesses a forty percent improvement in how it collaborates with its VO in comparison to the situation prior to cloud migrating its collaboration system. Regardless, the performance indicators were less promising and were within a margin of ten to thirty percent, probably due to the prototypical implementation. In order to overcome this limitation, the transition from research to product might require training, workshops, and feedback sessions to bring together system providers and consumers to jointly tailor the collaboratively concepts, models, and methodologies to the application field. One of the sixteen network partners of iPLON distrusted to some extent the collaboration tools, even after shaping their business models for their optimal adoption and recognising the benefits of cloud-based collaboration. This might be due to certain opposition to cloud-based collaborative practices that might be overcome with more intensive training sessions and by customising the collaborative tools to its specific needs.

Lesson learnt 5: to improve the adoption of cloud-based collaboration by increasing and extending the training and workshop sessions to explain the collaboration concepts and to customise the cloud-based collaborative network system to the needs of the organisations which are part of the network. Supported requirements: *R5*, *R8*, *R10*

The cloud-based collaborative networks system present SMEs with the opportunity to implement new collaborative business models to operate and offer their SEPs. This evaluation shows the importance of providing appropriate training so that organisations overcome the usual fears of the uncertain and of losing control that come with change. Organisations sticking to traditional sub-contracting strategies might miss the chance to increase their agility in responding to dynamic markets with changing circumstances. The glocal enterprise concept is embedded in the cloud-based collaborative networks platform to help SMEs, such as iPLON, expand into new markets in different regions.

Lesson learnt 6: to better accommodate how multiple collaborative networks interplay with interoperability standards as well as the behavioural issues in collaborative networks in holistic approaches combining networks of organisations, people, and machines. Supported requirements: *R5*, *R8*, *R10*

7.4.5 Evaluation-based cloud migration concept refinement

The iterative cloud migration of the collaborative networks system takes into account the evaluation lessons learnt and discussion to improve the cloud-based architecture recommended by the cloud migration concept. The concept refinement of the cloud-based collaborative networks system explained here focuses on supporting functionalities to incentivise the adoption of the cloud-based collaborative network system by selecting cloud migration alternatives that facilitate the integration of existing services and the data synchronisation with different software components. Integration and synchronisation mechanisms make it easier to attach different advanced or customised functionalities to the collaboration spaces —which is one of the fundamental building blocks of cloud-based collaborative networks systems— by integrating various associated software modules.

Cloud-based collaborative network system architecture

The cloud migration concept selects as a cloud migration alternative the architecture that Figure 7.23 shows. That is, an architecture whereby the software components described in the on-premises application system input model are moved to the CAS Open cloud, which is the cloud environment offering.

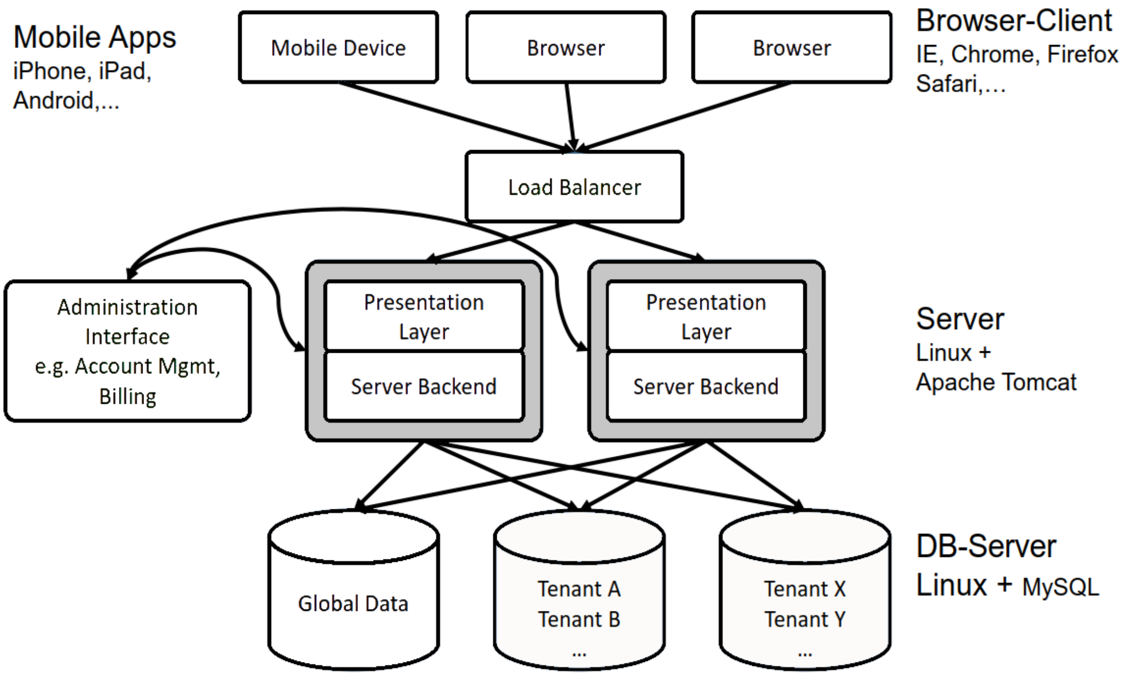

Figure 7.23: Deployment architecture of the cloud-based collaborative network system[645]

The selected cloud migration alternative uses industry standards, when possible, in addition to off-the-shelf components such as the load balancer, application servers, and database management systems. The basic infrastructure of the cloud-based collaborative networks system illustrated in Figure 7.23 uses a group of servers based on Linux to run an Apache Tomcat Java application server or a MySQL database management system. At the top, the presentation layer takes care of the visual interface with a server-side framework that renders AJAX-style user interfaces with the use of the Vaadin and Eclipse RAP software frameworks.

[645] Surajbali et al. (2014b), p. 7

The migration of these standard blocks is a bit lighter given that they are quite standard in the Java-based technology stack after adapting the load balancing and the VM administration interface to the selected cloud offering. An instance of the collaborative networks system can run on virtually any cloud infrastructure provided that it runs Java-based application servers (like Apache Tomcat) and that it supports the deployment of standard VM images.

Based on the evaluation lessons learnt, the cloud migration concept positively assesses cloud migration alternatives that provide mechanisms for the integration of collaborative services, cloud-based collaboration spaces, and other collaborative software modules.

Mechanisms for the integration of collaborative services

One of the evaluation lessons learnt[646] motivated fine-tuning the selection of cloud migration alternative to prioritise those alternatives providing tools to integrate external components or systems that add value to the cloud-migrated collaborative network system. As a result, cloud migration alternative are prioritised if they connect external services via two integration mechanisms based on either proxies (see Figure 7.25) or mashups (see Figure 7.24).

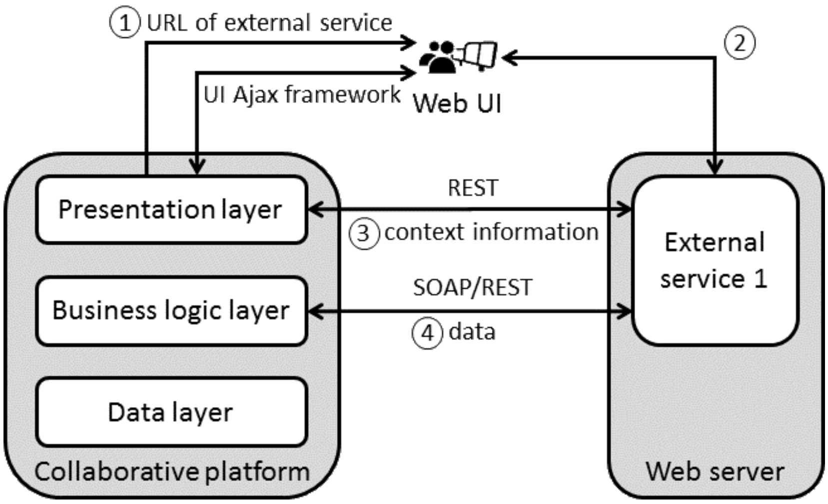

Figure 7.24: Mashup-based service integration for cloud-based collaboration[647]

Proxy-based integration mechanism. The left-hand side of Figure 7.25 shows service proxy plug-ins directing each external service invocation to the cloud-based

[646]Lesson learnt 2: to provide integration tools to include different services from different providers of, usually highly specialised, services to improve the cloud-based collaborative networks system

[647]Surajbali et al. (2014b), p. 7

collaborative networks platform according to the service registry and an additional generic plug-in. In this exemplary cloud migration alternative, a user might trigger SOAP or REST web service calls via the UI or the platform might do it due to a condition or business rule in its data model. The service plug-in collects, transforms, assembles, and distributes input and output data. In addition, plug-ins update the UI or the data model depending on whether the user or the platform invoked them. External services can be synchronously or asynchronously invoked. Synchronous invocations usually provide data to an application for display. Asynchronous calls notify a callback servlet upon completion to trigger workflows involving human interaction according to the transaction ID that identifies the invocation context such as the corresponding user or authentication information. Generic plug-ins can be used if the external service implements its specific interfaces.

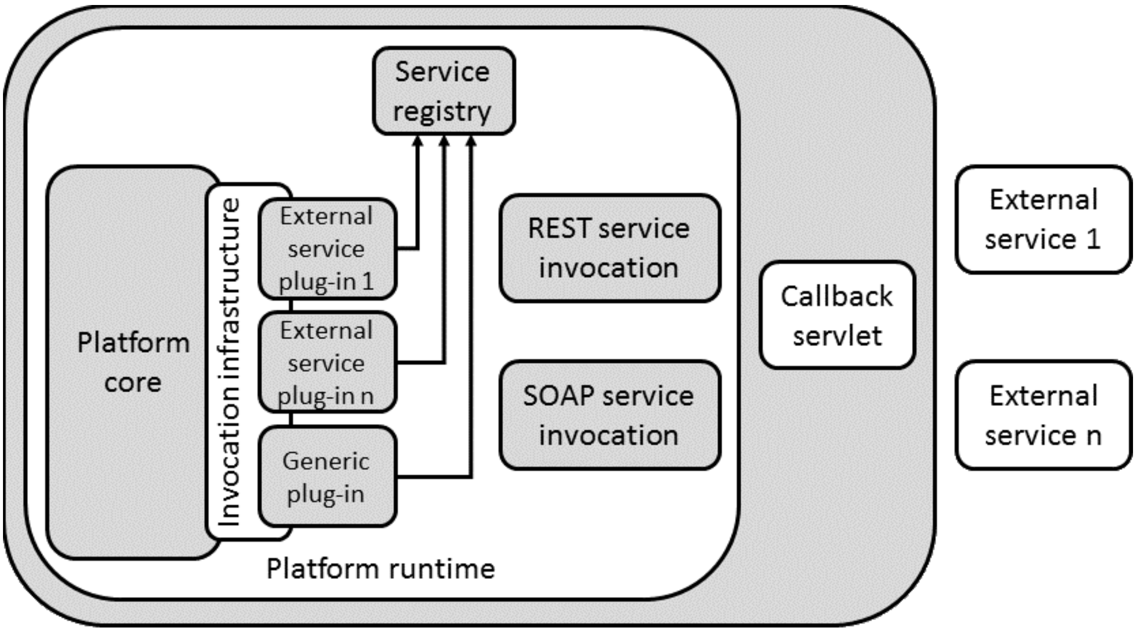

Figure 7.25: Proxy-based service integration for cloud-based collaboration[648]

Mashup-based integration mechanism. Cloud migration alternatives implementing this kind of mechanism have external services provide their UIs as a plug-in that uses web service RESTful APIs to exchange data with the cloud-based collaborative networks system[649]. Figure 7.24 shows at the top the Web UI, an embedded browser widget (such as iFrame) displaying the UI of the external service and using metadata with basic integration properties (for example, the

[648]Surajbali et al. (2014b), p. 6
[649]cf. Vesyropoulos et al. (2018), p. 239; Hoyer et al. (2008), pp. 601–602

external service's base URL). The presentation layer of the collaborative platform, on the left-hand side, sets the URL of the external service —(1) in Figure 7.24— and the context information needed to render the form. The browser invokes a request to the external service, (2) in Figure 7.24, with context data as HTTP GET parameters. Now, the external service can query the UI context information for the user's session, like the open data object or the selection of a known table: (3) in Figure 7.24. The user interaction with the UI triggers interaction with the REST or SOAP web services of the platform: (4) in Figure 7.24, and queries context information too, (3) in Figure 7.24.

Cloud-based collaboration spaces and other collaborative software modules

An additional lesson learnt[650] motivates offering cloud-based collaboration spaces functionalities with additional functionalities working on the data they store. The concept ranks higher those cloud migration alternatives providing this kind of functionalities. The cloud-based collaboration spaces follow a three-layer architecture with additional functionalities attached to them via the EIMInterface[651]. The EIMInterface uses context mechanisms to let the platform identify the business operation applied in a collaboration space. A plug-in checks the user permission and synchronises the associated data between the collaboration space, the cloud-based collaborative networks system, and external information systems. Collaboration spaces use context mechanisms to restrict the data they expose to the users who they authorise to access data in particular collaboration spaces by using key-value pairs connecting the collaboration space through the EIMInterface with the key identifying the collaboration space and the value representing its name. The business layer contains three main components to process data objects (DAOs), SQL data, and business objects. In addition, each of these three components provides corresponding plug-ins to transparently process data objects. The DAO component receives information through the EIMInterface and processes the meta information in this call with five plug-ins that help clients connect to the five workflow modules shown in the figure.

[650] Lesson learnt 4: to deliver cloud-based collaboration spaces and the cloud-based collaborative network system with additional functionalities to incentivise their adoption with added-value functionalities that act on the shared data

[651] cf. GloNet Consortium (2016), p. n/a, URL see Bibliography

[652] Surajbali et al. (2014b), p. 4

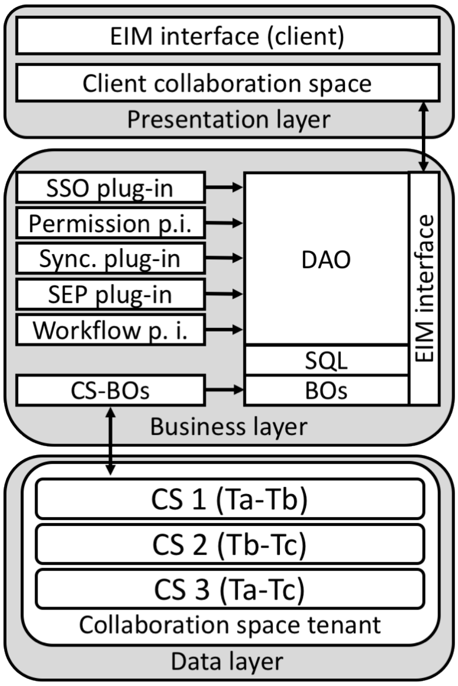

Figure 7.26: Collaboration space[652]

The proposed concept assesses the provision of other software modules offering functionalities for advanced collaboration such as single sing-on, permissions, synchronisation, and workflows. The single sign-on (SSO) module connects users to collaboration space environments using a unique login mechanism (which the following lines below explains). The permissions plug-in connects to the permissions module, which extends the server's native user access rights management to the collaboration space. The synchronisation plug-in connects to the synchronisation module, which synchronises data across multiple tenants. The service-enhanced products plug-in connects to the service-enhanced products module, which associates this module to the particular collaboration space. The process workflow plug-in connects to the process workflow module, which maintains all product business and process information. Those modules belong to the business logic layer, as for the data layer, it maintains the information a collaboration space shares. The collaboration space functionality uses a single tenant to hold all the collaboration spaces, which ensures low administrative overhead to create and delete collaboration spaces.

8. Conclusion

This dissertation concludes with Section 8.1 summarising the findings and contribution of this dissertation —the cloud migration concept— which result after following the design-oriented research design explained with Figure 8.1. The concept assists organisations in the cloud migration of their complex application systems by considering multiple criteria according to the priorities that organisations set for the migration. Further, Section 8.2 reflects on the application of the proposed concept to other situational contexts. Furthermore, Section 8.3 analyses the concept and its prototypical implementation in a critical manner in order to identify its limitations and how to overcome them with two potential extensions. Finally, on a subjective note, Section 8.4 presents the most relevant open challenges that future research works could address by extending this dissertation, the proposed concept, and the prototype.

8.1 Summary

This dissertation proposes the cloud migration DS concept, which the InCLOUDer prototype implements, to arguably guide organisations in designing how to migrate their application systems to cloud environments. The cloud migration concept was conceived and designed by allowing for weighing the cloud migration criteria that organisations might differently prioritise. The heterogeneous cloud migration criteria —which include technical, organisational, provider-related, business-related, economic, security, and privacy aspects— motivate and guide the cloud migration decision[653] supported by the proposed concept. The concept applies cognitive processes in the decision-making step to migrate application systems composed of different software components to several cloud environments. A particular stance taken in the design of the proposed concept is to enable it to offer partial cloud migration alternatives whereby sensitive data is kept at the local premises of the organisation to overcome privacy-related issues. In addition, partially migrated architectures can utilise both the local and cloud premises to overcome security and trust issues when executing their computing tasks or they can use other security approaches such as encryption or tokenisation.

The motivation to write this dissertation stems from the need to address the challenges of formalising the cloud migration criteria of organisations on a systematic and accurate manner that can be used to assist them in the partial cloud migration of their application systems. Hence, the aim of this dissertation is to provide a concept to support organisations in their cloud migration projects. The concept supports the decision-making process involved in finding the best adaptation to organisations' application systems in order to cloud migrate them according to the particularities of: the application systems, the migrating organisation, and the cloud environment configurations. Finally, the cloud migration concept aims at lowering the design effort by offering reusable cloud migration criteria and metrics to use them in conjunction to automate the cloud migration decision, as much as possible. Still, the concept offers the necessary mechanisms to include organisations in the loop to let them manually drive or adjust the cloud migration decision.

Derived from the aim of this dissertation[654], four research questions frame, in Chapter 1, the investigated research problems. The answers to the research questions takes place throughout the entire dissertation and it is not limited to one specific research goal or any result after pursuing those four goals (see Figure 8.1). The answer to the first research question —how can a decision support concept assist organisations in deciding how to migrate their application systems to cloud environments?— is answered in this dissertation by proposing the cloud migration DS concept, which uses the AHP to help organisations select the

[653] See Section 5.1 for more details criteria driving the cloud migration project
[654] Chapter 1 describes, in Section 1.2, the research questions of this dissertation and its aim

appropriate cloud migration alternative that runs on both their premises and cloud environments. The second research question —what are the decision-making criteria driving organisations to decide how to migrate their application systems to cloud environments?— is answered with the thorough analysis of cloud migration from both literature and business practices; see Chapters 3 and 4. A set of cloud migration criteria classified as: business- and economics-related, technical, provider-related, and security and privacy criteria. Chapter 5 also describes the needs of cloud environments modelling to partially answer the third research question —what are the criteria or requirements organisations consider to select cloud environments to migrate their application systems?— in addition to the cloud modelling module proposed in Chapter 6. The answer to the last research question —which requirements and characteristics of the system prior to the cloud migration do organisations consider in the decision-making process?— is documented in Chapter 6 as well, with the approach taken with the on-premises modelling module.

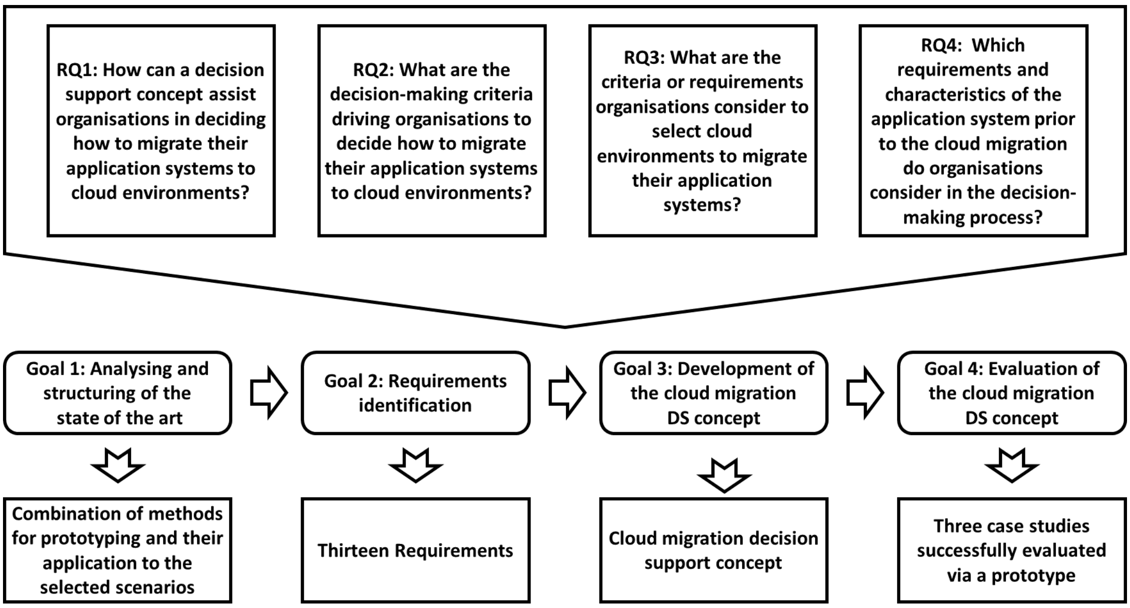

Figure 8.1: Goals of this dissertation and the produced artifacts after achieving them[655]

This research work adopts a design-oriented research stance to achieving the four specific goals. These goals are derived from the aim of this dissertation and answer the above research questions. The achievement of each goal requires building on top of the previous ones. The goals are iteratively and incrementally pursued to construct and evaluate the cloud migration DS concept by, for example,

[655]Own illustration

going back from goal 4 to goals 3, 2, and 1. The performed research focuses on the outcomes of constructing concepts and prototypes[656]. Four artifacts are the outcomes of achieving each goal, as different parts of this dissertation explain[657] and as Figure 8.1 shows.

The research frame of reference, informed by these goals, guides the execution of the research design in order to answer this dissertation's four research questions. The concept is realised on the basis of the frame of reference, which also guides the scenario-based evaluation and the refinement of the cloud migration concept in line with the lessons learnt during the evaluation[658].

8.2 Application of the concept to other situational contexts

The process of adapting information systems to a new technological change cannot be considered a new research field. It usually entails the migration of application systems to a new technological solution or their adaptation to a new computing infrastructure, which has been a common practice in the field of information systems for decades. The cloud migration concept was designed with this in mind and it is accordingly conceived to be future-proof via its capacity to be easily extended and updated with new features. Even if predicting future technological developments can be difficult, combining the flexibility and extendibility of the concept and its prototypical implementation combined with being aware of trends and new technologies will help the concept to remain relevant and useful.

The concept can be applied to other situational contexts including DevOps, the use of microservices, or other computing paradigms such as quantum computing or MapReduce. The models used to describe the target computing environment are extensible by design and the idiosyncrasies of the new computing paradigm can that way be included in the cloud migration decision-making process. In

[656] For more details on design-oriented information systems research and its application to this dissertation, refer to Section 4.7 in Chapter 4

[657] Section 4.7 describes, in Chapter 4, the result of Goal 1; Chapter 5 explains the resulting artifact of achieving Goal 2; Goal 3 produces an artifact documented in Chapter 6; and the result of Goal 4 is described in Chapter 7

[658] Again, see Section 4.7, Chapter 4, for more details on the research design of this dissertation

addition, the cloud migration criteria and its sub-criteria thrive when extended and tailored to the migration project which include the shifting priorities of organisations in line with technological changes. The following section, Section 8.3 explains potential extensions that benefit from the future-proof orientation of the concept and prototype to having 1) additional cloud deployment capabilities to be ready for Devops; and 2) new concept requirements and cloud migration projects to accommodate microservices and other computing paradigms. In the case of Devops, continuing the development of the concept so that it can move from the design time to the run-time could facilitate the on-line decision-making process that reacts to changes in the computing environment and computing problem in real time. For the case of microservices, quantum computing, and MapReduce, the concept will use the same software modernisation approaches (like Architecture-Driven Modernisation as well as reverse and forward engineering practices) tailored to the new computing context. In addition the problem to re-scatter parts of the application system will remain the same. That is, using microservices would entail breaking down the software components to micro- or miniservices to profit from the self-contained execution of them in clusters using Kubernetes or Docker containers running (or not) in the cloud. In the case of quantum computing and admitting that classical computing will stay with us for a while, the initial steps to adopt the computing paradigm could entail migrating parts of the application system. That is, the same approach followed by the concept. This scenario lets at the same time use classical computing in parts of the application system (similar to the local premises in the proposed concept) while allowing for the migration of some parts of the application system that could benefit from properties such as quantum entanglement and the superposition of states to achieve faster computing and parallelisation that can deliver fundamental changes in simulations, machine learning, or cryptography. That way, the concept stays current and caters to new applications such as drug discovery, financial modeling, supply chain optimisation, artificial intelligence, or cryptography.

8.3 Critical analysis: extensions to the concept

The evaluation of the cloud migration concept happens in the context of the three cases shown in Chapter 7. Although these are cloud migration projects of particular interest and relevance, the evaluation is limited to these three evaluation scenarios and would benefit from additional and larger evaluation scenarios. The concept could thereby be used in other application domains with settings differing to those of the three presented scenarios. The evaluation scenarios are carefully selected because they represent kind of application systems similar to other potential target application systems but this dissertation could benefit from additional research that would extend the scope of the evaluation. In particular, the evaluation scenarios are selected due to: the relevance of cloud standardisation, portability, and interoperability in PaaS environments (see Section 7.2); the complexity of migrating heterogeneous and multi-layered application systems such as BI systems (see Section 7.3); and the particular focus on cloud migration criteria very relevant for organisations migrating to the cloud such as collaboration in collaborative networks systems (see Section7.4).

The proposed cloud migration concept aims to holistically support the decision of how to cloud migrate application systems and to which cloud environments. It considers multiple and heterogeneous cloud migration criteria defined by organisations. The cloud migration concept aims at respecting data privacy, security, and trust criteria by cloud migrating those software components of application systems which store or process less sensitive data or that are appropriately secured. The variance in the offerings that cloud environments provide, combined with the different target application system and the various interests of organisations in migrating to cloud environments, create a huge solution space for the cloud migration problem. The concept provides decision support to cloud migrate application systems by using three input models to propose an output model.

As input models, the cloud migration concept takes into consideration the description of the application system target for migration, the models of existing cloud environments, and the organisations' cloud migration criteria. The incremental and iterative analysis of the developed prototypes and the evaluation of the proposed concept by applying that prototype to three realistic cloud migration scenarios —the cloud migration to PaaS offerings, the cloud migration of business intelligence, and the cloud migration of collaborative networks systems— show its applicability to cloud migration projects similar to those that organisations will face. Despite the successful use of the concept to three different scenarios, this dissertation presents, in the opinion of the author, two limitations that could be addressed with two potential extensions:

8.3.1 Extension 1: additional cloud deployment capabilities

An extension incorporating full-blown cloud deployment would imply that organisations could integrate the extended concept in their continuous integration, delivery, and deployment cycles. The concept for partial cloud migration could be included in the operation of traditional development, testing, and production environments. Additionally, the lessons learnt by undertaking this extension could be reused for the migration to future computing environments. Extending the cloud migration concept with full deployment capabilities could transcend the research status of the concept and increase its technology readiness level to exploit it in operational industrial settings. Incorporating the cloud migration concept into the continuous integration (CI) would entail merging the changes in the cloud-enabled architecture in the main code branch as often as possible. A step further could take the concept to make selected changes available to the users of the application system while the cloud-enabled application system is able to scale to timely react to the computation, storage, and networking needs (continuous delivery CD). Finally, the inclusion of the concept in the continuous deployment (likewise, CD) would have every change in the cloud architecture that has passed all stages of the production pipeline ready for the users without a particular release day, which is particularly interesting in enterprise systems to use the user as the tester.

Architecture-Driven Modernisation (ADM) initiatives propose applying knowledge discovery to software modernisation in order to understand the software artifacts that compose application systems. The InCLOUDer prototype can already use the source code of the target application system, if it exists, to extract its architecture[659]. This step that uses source code and architectural descriptions can

[659] InCLOUDer implements Requirement 3 (see Section 5.1 in Chapter 5) via two design decisions —FDD-01 and FDD-02— explained in 7.1 of Chapter 7

be automatically executed, without human intervention, in order to build the on-premises application system model. The model which describes the architecture, properties, and requirements of the target application system. The on-premises model combined with the models of the cloud environment configurations and of the cloud migration criteria allow the proposed concept to automatically rank all cloud migration alternatives if each cloud migration criterion is associated at least to one metric for its assessment. Therefore, the inclusion of automatic deployment capabilities in the concept would have the potential to: automate the entire process in ideal cloud migration projects or increase automation in less ideal cases. Of course, boosting the levels of automation in cloud migration would reduce the costs and efforts needed to modernise an application system in addition to reducing the amount of errors made in the process.

The proposed concept paves the way for improving its automatic deployment and cloud portability capabilities. Portability libraries that could be used are mentioned in the PaaS migration scenario of the evaluation chapter (see Chapter 7). Although, the cloud migration concept and prototype incorporate the first building blocks to deploy application systems to cloud environments, it currently focuses on the holistic design of cloud-enabled application systems. Future extensions to this dissertation could complete the entire cloud migration process —from the conception of a cloud-enabled application system to its operation— by extending the concept and prototype so that they fully deploy the target application system to cloud environments.

Containerisation is a current trend in computing that can be used to deliver new cloud-based application systems built as containerised microservices. Organisations break complex application systems into a series of smaller services, which are more specialised and manageable. The cloud migration concept could be extended to assist the repackaging of application systems into containers or containerised microservices. With containerisation, computing resources are used more effectively and services run uniformly and consistently on any infrastructure. The cloud migration concept could be reused to assist in encapsulating an application as a single executable package of software that bundles (in a container) application code with its related configuration files, libraries, and dependencies.

Containerised application systems can provide higher degrees of isolation by not bundling in a copy of the operating system. In addition, bundling application systems as a collection of smaller services or components arguably make them more portable, scalable, efficient and easier to manage. The isolation provided by containers potentially increase the agility of organisations, fault isolation, and the facilitates providing higher security levels. In addition, the speed and efficiency of application systems could increase due to its lightweight character.

8.3.2 Extension 2: new concept requirements and cloud migration projects

The development of the cloud migration concept targeted the fulfilment of thirteen requirements identified by analysing research literature after gaining hands-on experience through three research projects related to the cloud migration problem[660]. Despite the process followed to deduct these requirements, which materialised in design decisions affecting the implementation of the prototype[661], the InCLOUDer prototype might require being extended for it to be able to cover future requirements. Due to the huge variety and complexity of cloud migration scenarios and to unforeseeable technological changes in cloud computing, it is not possible to implement the prototype so that it covers all potential scenarios. However, the concept is designed to provide the flexibility to accommodate future extensions.

This dissertation had to define a clear scope not to become too broad and to avoid risking not accomplishing the intended breakthrough in the holistic cloud migration decision support. Its scope is limited to taking into account cloud migration criteria to cloud migrate application systems in such a way that optionally keeps software components at the local premises of the migrating organisation for data sensitivity and trust reasons. The concept tested the coverage of the specific requirements of the three evaluation scenarios. Therefore, it would be beneficial to extend the evaluation to test the requirements coverage in further cloud migration projects which are independent of this dissertation. With all this in mind, the proposed concept was designed according to an extensible architecture that lets others

[660] Chapter 5 explains the thirteen requirements of the proposed concept

[661] The design decisions that affect the InCLOUDer prototype are explained in Chapter 7, Section 7.1

seamlessly extend its prototypical implementation to include new requirements by adding modules to the Eclipse RCP-based prototype[662]. This adds flexibility to add new models to be used in the decision making. In addition, the concept offers out-of-the-box modelling flexibility to describe the cloud environments, target application system, and cloud migration as needed. The need to extend the concept would respond to new requirements discovered by using the InCLOUDer prototype in new scenarios, case studies, or cloud migration projects. Additionally, potential extensions could aim to accommodate technological changes in cloud computing or at tackling challenges when migrating application systems to related computing technologies.

8.4 Open challenges

This section explains how researchers and business practitioners could build on top of this dissertation to go beyond its scope and address the challenges that remain open after the completion of this dissertation. The research performed to publish this dissertation sometimes touches upon the topics explained below but that represent a new dissertation on their own. Often times, this research opens the door to extending the cloud migration concept in upcoming research trends given that cloud computing is a very transversal technology that can be used as the infrastructure used to optimise different computing systems or in combination with other research strands.

In addition to investigating cloud migration projects in new application fields, the lessons learned via the evaluation scenarios —in PaaS, BI, or Industry 4.0— to exploit the proposed concept in those fields by increasing its maturity so that it can be used in operational environments. Within the field of Industrial Intelligence the concept could be used for decision support in industrial contexts or to study Analytics Architectures. Further open challenges would entail equipping the cloud migration concept to work at run time (instead of at design time) or improving its user interfaces, user experience, or quality of experience. The concept considers soft reservations and cloudlets in terms of how these mechanisms affect the cloud migration decision if cloud environments support them. More advanced concepts

[662] Chapter 6 explains the extensible architecture of the cloud migration concept and Section 7.1 describes in Chapter 7 its implementation in the InCLOUDer prototype

related to soft reservations (such as advanced soft and hard reservations) and cloudlets (such as fog and edge computing) could be studied to see how they affect the cloud migration and how to equip the concept to assess those effects. Finally, new viewpoints could be added to this dissertation by considering multi-cloud and cloud bursting orchestration in addition to the implications of GDPR with regards to data security, privacy, and trust in cloud environments.

8.4.1 Run-time orientation

As already mentioned, increasing the level of automation of the decision making would require adding functionalities to the proposed concept. For example, new research could add capabilities to partially deploy the target application system according to the selected deployment scheme. Higher levels of automation would result in saving time, costs, and effort when cloud migrating an application system and, at the same time, the migration process would become less error-prone. An open challenge relates to making the cloud migration concept go from working at design time as it currently does, to acting at run time by leveraging automatic cloud deployment capabilities. In that case, the cloud migration concept would adapt application systems and their behaviour at run time with little or no human intervention (self-adaptation). The concept would react at run time to dynamic changes happening in the application systems and their environments. These changes could trigger on-line adaptations in the cloud migration alternatives to react to alterations in the software or deployment architecture. Likewise, adaptations could respond to changes in the roles taken by stakeholders and to the inclusion or removal of computing devices in the IT ecosystem. The proposed concept could quickly react to changes in the cloud migration criteria as well as being able to orchestrate a different deployment upon quality of service degradations caused by the cloud environment.

8.4.2 Standardisation of cloud migration criteria and metrics

The cloud migration concept proposes different cloud migration criteria that are evaluated in the fields of PaaS offerings, business intelligence, and collaborative networks[663]. Different metrics are also proposed for specific cloud migration

[663] See Chapter 7 for more details about the evaluation

criteria. This dissertation explains how sets of cloud migration criteria and their metrics can be reused for the cloud migration of different application systems by tailoring them, if necessary, to the organisation carrying out the cloud migration and its particular project.

An open challenge entails extending the templates of cloud migration criteria and their metrics that could even lead to the successful standardisation of the cloud migration criteria. These extensions could happen in parallel to the evaluation of their re-usability, easiness of use, and fit for purpose. Likewise, future research work activities might extend the proposed metrics[664] with metrics for new cloud migration criteria, multiple metrics for an existing cloud migration criterion, or metrics to assess the fit for purpose of different cloud environments according to their configuration models.

8.4.3 Multi-cloud orchestration

The open challenge here involves automating and continuing the work done in muti-cloud by adding the orchestration of (micro)services. Services would be deployed to multi-cloud environments to exploit the benefits of each cloud environment in a very fine-grained manner with services deployed to interoperable multi-cloud ecosystems. Within such deployments, the automatic deployment could use state-of-the-art cloud management systems to orchestrate the workloads between the cloud environments and local premises. In addition, a fourth evaluation scenario could be added to assess the performance of the designed multi-cloud application systems.

8.4.4 Cloud bursting

This dissertation identified the open challenge of how to consider cloud bursting in the cloud migration alternatives generated by the cloud migration concept. Cloud bursting could be used to accommodate peaks in the demand of computation resources beyond the in-house capabilities. Cloud bursting is often used in private

[664] Section 6.3 proposes, in Chapter 6, different metrics to assess how well the cloud migration alternatives deliver a cloud migration criterion

[664] Flexera (2023), p. n/a, URL see Bibliography; Enstratius (formerly enStratus), EnStratius, Inc. (2013), p. n/a, URL see Bibliography; and Scalr, SCALR (2022), p. n/a, URL see Bibliography

cloud environments or local premises to direct overflow traffic to public cloud infrastructures to avoid interruptions in the service. The concept could let organisations benefit from the economic savings of using cloud bursting (that is, from using the local premises when no extra computation is needed) while increasing the flexibility of their application systems. Cloud consumers would, after the implementation of the cloud bursting model, pay for the public cloud resources in cases of peaking demand without compromising the Quality of Service or over-provisioning, which results in extra costs. Additionally, the proposed concept with cloud bursting functionalities could free up local resources for business-critical application systems as they would use public cloud resources instead. The use of cloud bursting poses similar issues as the concept tries to address; in terms of security, compliance, latency, load balancing, and platform compatibility.

8.4.5 Mobile cloud

With some adjustments, the proposed concept could fully support the mobile cloud computing scenario. This open challenge has mobile devices outsource computation to cloud environments and the local or near premises to overcome its limitations in terms of storage, battery, and computing power. Given the current advances in mobile computing and the huge amount of resources mobile devices have nowadays, the amount of scenarios potentially benefiting from the computation offloading might have decreased or even become irrelevant for some of the traditional applications of mobile outsourcing. However, it could become more relevant than ever for other scenarios that use augmented reality, virtual reality, or mixed reality due to the amount of constantly updated data they need.

Additionally, cloud environments provide a collaboration platform that mobile users could share to work on the same files, knowledge base, or models.

8.4.6 Fostering the adoption of the concept

An open challenge is to scale the proposed concept so that it can be exploited after the completion of this dissertation. The first step to increase adoption would be to improve the InCLOUDer prototype in terms of usability: its user interface, user experience, and the overall quality of experience. In all, this would improve the practicality of the concept.

In addition, the concept could be extended so that it is offered in marketplaces offering cloud-based services or offerings in order to incentivise its adoption by cloud consumers and providers[665]. In this scenario, the cloud migration concept offers standard cloud migration criteria and user-generated cloud migration templates with their associated metrics. More cloud consumers could turn to the marketplace if it offered a big cloud ecosystem with higher levels of interoperability and cloud portability. In turn, this would motivate additional cloud consumers to participate in the marketplace to access cloud migration criteria which proved to work in the past, as well as updated cloud environments models that accurately describe cloud offerings. Additionally, cloud providers could be attracted to the marketplace given the number of cloud consumers and be motivated to model their cloud offerings with the cloud environments modelling module. This would liberate organisations from the burden of modelling the potential target cloud environments.

8.4.7 Soft reservations

Soft reservations might be offered in combination with hard reservations by an increasing number of cloud providers, which would affect the cloud migration decision making, with regards to pricing and availability, as Section 6.8 describes in Chapter 6. The proposed concept considers soft reservations when ranking cloud migration alternatives according to whether the cloud offerings in them use soft and hard reservations. Cloud consumers are more certain of their resource requirements when they softly reserve resources and that allows cloud providers to better plan the use of their infrastructures. In turn, cloud providers offer cheaper rates (while respecting the privacy of their consumers) to cloud consumers in order to: on the one hand, give them incentives to accurately forecast their resource needs while, on the other hand, only charging them once they issue a hard reservation. An open challenge remains to more accurately consider the effects of using advanced mechanisms to reserve and pay for cloud resources in the design of cloud-migrated software architectures and in the rearrangement of software components to the selected cloud environments; hence, the effects of the advanced reservations mechanisms in the cloud migration decision making.

[665] Fostering the adoption of the proposed concept via its integration into a marketplace is discussed in Chapter 7, Section 7.2

Soft and hard reservations are mechanisms that help to better forecast resource utilisation as estimations from both cloud providers and consumers are more accurate. Cloud consumers can provide resources faster and in a more effective manner as they can exactly plan when to provide cloud resources and how many are needed. Reservations lead to win-win situations whereby cloud consumers lower their costs, by booking in advance at cheaper rates, and providers can efficiently manage their resources. For example, providers can minimise their consumption by forcing physical machines to sleep or by orchestrating a different architecture. The role of the proposed concept would include optimising the placement of software components by taking into account the costs of the different reconfiguration options[666]. Like mentioned above, software architecture adaptations could also take place at run time in order to accommodate dynamic changes in the computing environment, as well as changes in its context of operation.

8.4.8 Cloudlet-based, edge, and fog computing

The work done on cloudlets[667] could be extended to address open challenges related to the particularities of using nearer computation; such as the infrastructure geographically closer to the cloud consumer. As none of the evaluation scenarios incorporate cloudlets in the decision making, an extension to this dissertation might evaluate the proposed concept in a cloudlet-based scenario (or any of the related concepts such as fog or edge computing) that would evaluate the components placement and cloud migration decision, as suggested in the aforementioned papers.

The study of the effects of adding fog and edge computing as target computing environments, in addition to cloud environments, pose interesting research questions related to the distribution of software components and its effects in the cloud migration decision. Fog and edge computing place intelligence and processing closer to where data originates; that is, pumps, motors, sensors, or relays. Fog computing pushes intelligence down to the local area network, thus processed

[666] cf. Mohammadi et al. (2013), pp. 226–228

[667] The effects of cloudlets in the cloud migration decision are explained in Section 6.9, Chapter 6, in addition to the details given in two published papers: cf. Al Ali et al. (2014), pp. 1–4; Tesgera et al. (2014), pp. 253–253

data is stored in a fog node or IoT gateway. Whereas, edge computing pushes the intelligence, processing power, and communications of edge gateways or appliances into devices such as programmable automation controllers (PACs). The proposed concept could focus on the data governance of these systems and on how to distribute processing between cloud services and nearer gateways or devices.

8.4.9 Regulations for cyber-security and privacy: GDPR, Cloud Act, and more

Data security, privacy, and trust issues are fundamental aspects that the proposed cloud migration decision support concept takes into account. In fact, this is one of its unique selling points as it enacts partial migration to overcome them. An open challenge is to deliver a cloud migration concept that generates cloud migration alternatives that comply with GDPR to a reasonable extent[668]. GDPR is of particular relevance given that this dissertation was done within the EU but the same could be said for other existing regulations, such as the CLOUD Act and the Do Not Track Act, or upcoming ones such as the EU ePrivacy Regulation[669]. These regulations and laws do not remain at the technical level but imply changes in the structure of organisations and the business or legal processes they follow. Accordingly, the extension to the proposed concept should also recommend technical and organisational measures to re-engineer organisational processes in order to comply with the applicable regulation and law.

8.4.10 Cloud computing as horizontal technology applied to other research fields

Cloud computing is a horizontal technology that can be applied to multiple research and technological fields that typically build on top of cloud-based infrastructures. The cloud migration concept could be extended to cater to the specific needs of domains that include: Industry 4.0, bioinformatics, chemistry, augmented reality

[668] The BPR4GDPR researched on GDPR: cf. Lioudakis et al. (2019) pp. 72–78, a paper co-authored by the author of this dissertation

[669] For more details on the regulatory and privacy-related implications of some of the existing regulations and laws please refer to Chapter 4 Section 4.6, under the regulatory provider-related cloud migration criteria

settings, and to deployments using near computation and soft reservations. The cloud migration criteria will be different as each domain might benefit from using cloud computing for different reasons. Some fields of application will give more weight to the fact of using an ecosystem whereby on-demand resources work on the presented computational problems while others will prioritise improving multi-stakeholder collaboration. New evaluation scenarios specific to one of the research and technological fields might even bring forward, as a lesson learnt, the need to extend the proposed concept with additional modules.

Within the already explored cyber-physical systems domain[670], Industry 4.0 services can be outsourced so that other organisations re-use them to offer high value-added products. In that manner, the Internet of Things joins the Internet of Services to enact the fourth industrial revolution. The digitalisation of manufacturing techniques and logistics capabilities have the potential to deliver massively customised products adapted to the individualised requirements of the customer[671]. The proposed concept has explored this research avenue but future undertakings might extend it by studying additional particularities of Industry 4.0 settings such as Shopfloor Management Systems[672]. Likewise, scenarios for remote presence across physical locations benefit from assisted cloud migration. These scenarios present particularities to be taken into account to include the sensors coming from Interconnected Cyber-Physical Systems (CPSs) and imply using internal and cross-organisational services that cooperate in real time. CPSs can use cloud computing architectures to combine different providers and use various service interfaces to share services and data.

Open challenges in line with this dissertation relate to tackling cloud interoperability and collaboration by enacting appropriate access control and designing privacy-aware trusted cloud infrastructures[673]. These scenarios extend to settings whereby augmented reality is used to assist human workers so that they learn or operate in

[670] Section 6.13 explains in Chapter 6 the work done in this dissertation in the field of Industry 4.0: cf. Juan-Verdejo and Surajbali (2016), pp. 11–23

[671] cf. Müller et al. (2018) pp. 2–5; Lasi et al. (2014) pp. 239–242

[672] The DiGAP project worked on Shopfloor Management Systems for Industry 4.0 settings: cf. Pfeiffer et al. (2018) pp. 113–129

[673] The Trust 4.0 project worked on privacy-aware trusted cloud infrastructures: cf. Al-Ali et al. (2019) pp. 277–284; Al-Ali et al. (2018b) pp. 1–4; Al-Ali et al. (2018a) pp. 4–40

dynamically changing production environments[674]. With AR, users get spatially registered information on the task they are performing directly in their field of view so that this information guides them through new tasks, such as assembling new products and operating or repairing new machines. Virtual, augmented, or mixed reality systems only show information in the relevant spatial (virtual or real) context. Modern bioinformatics and cheminformatics usually need lots processing and could be additional evaluation scenarios to test the efficacy of their cloud migration with the proposed concept. In these fields, heavy processing happens on demand to handle lots of data to get timely solutions while coping with computation costs.

8.4.11 A personal note: now starts the fun...

The path towards completing this dissertation, while challenging, has been an uplifting one due its potential to contribute to scientific and technical progress. My major concern would be that the findings and lessons learnt through this research would be unused and not be either integrated into organisational processes or systems going forward. I however, have faith in this dissertation, the cloud migration concept, and the InCLOUDer prototype trusting they will be used for cloud migration or adapted for related undertakings after the lifetime of this dissertation.

Firstly, the pace at which the software development evolves is accelerating and becoming more complex. As a result, organisation might increasingly benefit from being assisted when readying their application systems for new computational paradigms. In addition, the concept could be used to support the migration to cloud architectures that use containerisation or other technologies. The concept could be adapted to assist organisations so that they more effectively migrate to related computing infrastructures that implement new technologies or concepts.

Finally, and on a more practical note, my new job after this publication entailed applying the knowledge summarised in this dissertation. In particular, the lessons learnt during this research were applied to the digital transformation of how IT services are provided in the organisation. This challenge included leading hybrid cloud transition programmes for the migration of corporate e-mail and electronic messaging services to cloud environments.

[674] cf. Liu and Xu (2017) p. 1; Paelke (2014) pp. 1–4

Bibliography

Abbas, A. S., Jeberson, W. and Jeberson, K. (2012), The need of re-engineering in software engineering, in: International Journal of Engineering and Technology, vol. 2, 2012, iss. 2, pp. 292–295

Abdulateef, A. A., Mohammed, A. H. and Abdulateef, I. A. (2020), Cloud Computing Security For Algorithms, in: 2020 4th International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT), IEEE, pp. 1–5

Abts, D. and Mülder, W. (2009), Grundkurs Wirtschaftsinformatik: Eine kompakte und praxisorientierte Einführung, 6th Edition, Wiesbaden 2009

Aebi, D. (1997), A process model for re-engineering, migration and multi-use of business data, in: Proceedings of the 1st Euromicro Conference on Software Maintenance and Reengineering (CSMR 1997), pp. 305–310

Agutter, C. (2012), ITIL foundation handbook, London 2012

Ahmad, A. and Babar, M. A. (2014a), A framework for architecture-driven migration of legacy systems to cloud-enabled software, in: Proceedings of the Working IEEE/IFIP Conference on Software Architecture (WICSA 2014), pp. 1–7

Ahmad, A. and Babar, M. A. (2014b), Towards a pattern language for self-adaptation of cloud-based architectures, in: Proceedings of the Working IEEE/IFIP Conference on Software Architecture (WICSA 2014), pp. 1–6

Akinrolabu, O., New, S. and Martin, A. (2017), Cyber supply chain risks in cloud computing–bridging the risk assessment gap, in: Open Journal of Cloud Computing, vol. 5, 2017, iss. 1, pp. 1–19

Al-Ali, R., Bures, T., Hartmann, B.-O., Havlik, J., Heinrich, R., Hnetynka, P., Juan-Verdejo, A., Parizek, P., Seifermann, S. and Walter, M. (2018a), Use Cases in Dataflow-based Privacy and Trust Modeling and Analysis in Industry 4.0 Systems, Technical Report Karlsruhe Reports in Informatics 2018/9, Karlsruhe 2018

Al Ali, R., Gerostathopoulos, I., Gonzalez-Herrera, I., Juan-Verdejo, A., Kit, M. and Surajbali, B. (2014), An Architecture-Based Approach for Compute-Intensive Pervasive Systems in Dynamic Environments, in: Proceedings of the 2nd International Workshop on Hot Topics in Cloud service Scalability (HotTopiCS 2014), pp. 1–3

Al-Ali, R., Heinrich, R., Hnetynka, P., Juan-Verdejo, A., Seifermann, S. and Walter, M. (2018b), Modeling of dynamic trust contracts for industry 4.0 systems, in: Proceedings of the 12th European Conference on Software Architecture (ECSA 2018), pp. 45–53

Al-Ali, R., Hnetynka, P., Havlik, J., Krivka, V., Heinrich, R., Seifermann, S., Walter, M. and Juan-Verdejo, A. (2019), Dynamic security rules for legacy systems, in: Proceedings of the 13th European Conference on Software Architecture (ECSA 2019), vol. 2, 2019, pp. 277–284

Al-Dhuraibi, Y., Paraiso, F., Djarallah, N. and Merle, P. (2017), Elasticity in cloud computing: state of the art and research challenges, in: IEEE Transactions on Services Computing (TSC 2017), vol. 11, 2017, iss. 2, pp. 430–447

Alcala Alvarez, M. d. C., Fernández-Montes, A., Ortega, J. A. and Gonzalez-Abril, L. (2012), The CICA GRID-A Cloud Computing Infrastructure on Demand with Open Source Technologies, in: Proceedings of the International Conference on Enterprise Information Systems (ICEIS 2012), vol. 2, 2012, pp. 301–304

Aleem, S., Ahmed, F., Batool, R. and Khattak, A. (2019), Empirical Investigation of Key Factors for SaaS Architecture Dimension, in: IEEE Transactions on Cloud Computing (TCC 2019), pp. 1–14

Alegre, U., Augusto, J. C. and Clark, T. (2016), Engineering context-aware systems and applications: A survey, in: Journal of Systems and Software, vol. 117, 2016, pp. 55–83

Alfraihi, H. and Lano, K. (2017), The Integration of Agile Development and Model Driven Development, in: Proceedings of the 5th International Conference on Model-Driven Engineering and Software Development (MODELSWARD 2017), pp. 451–458

Ali, M., Khan, S. U. and Vasilakos, A. V. (2015), Security in cloud computing: Opportunities and challenges, in: Information sciences, vol. 305, 2015, pp. 357–383

Allan, K., Loot, M. and Esterhuyse, M. P. (2016), The Business Value of Cloud Computing in South Africa, in: African Journal of Information Systems (AJIS 2016), vol. 8, 2016, iss. 2, pp. 1–20

Alonso, J., Orue-Echevarria, L., Escalante, M. and Benguria, G. (2017), DECIDE: DevOps for Trusted, Portable and Interoperable Multi-Cloud Applications towards the Digital Single Market, in: Proceedings of the 7th International Conference on Cloud Computing and Services Science (CLOSER 2017), pp. 397–404

Alonso Juncal, M., Orue-Echevarria, L., Escalante, M., Gorroñogoitia, J. and Presenza, D. (2013), Cloud modernization assessment framework: Analyzing the impact of a potential migration to cloud, in: Proceedings of the IEEE 7th International Symposium on the Maintenance and Evolution of Service-Oriented and Cloud-Based Systems (MESOCA 2013), pp. 64–73

Aloran, M., Eid, H. and Al-Sarayreh, K. T. (2015), A High Quality Software after Maintenance Depend on Effectiveness measures, in: Proceedings of the International Conference on Intelligent Information Processing, Security and Advanced Communication (IPAC 2015), pp. 64–68

Alpar, P. and Polyviou, A. (2017), Management of Multi-cloud Computing, in: Proceedings of the International Workshop on Global Sourcing of Information Technology and Business Processes (ITBP 2017), pp. 124–137

Althani, B., Khaddaj, S. and Makoond, B. (2016), A quality assured framework for cloud adaptation and modernization of enterprise applications, in: Proceedings of the International IEEE Conference on Computational Science and Engineering (CSE 2016), the International IEEE Conference on Embedded and Ubiquitous Computing (EUC 2016) and the 15th International Symposium on Distributed Computing and Applications for Business Engineering (DCABES 2016), pp. 634–637

Amazon.com, Inc. (2021a), Amazon Elastic Compute Cloud (Amazon EC2), on the website of Amazon, http://aws.amazon.com/ec2/, last updated 2021

Amazon.com, Inc. (2021b), Amazon Relational Database Service (RDS), on the website of Amazon, http://aws.amazon.com/rds/, last updated 2021

Anandasivam, A. (2010), Consumer preferences and bid-price control for cloud services, PhD Dissertation at the Karlsruhe Institut für Technologie (KIT), Karlsruhe 2010

Anastasi, G. F., Carlini, E., Coppola, M. and Dazzi, P. (2014), Qbrokage: A genetic approach for qos cloud brokering, in: Proceedings of the 7th International IEEE Conference on Cloud Computing (IEEE CLOUD 2014), pp. 304–311

Anderson, D., Korpela, E., Werthimer, D., Adb Bowyer, W. and Cobb, J. (2021), SETI@home: the search for extraterrestrial intelligence, on the website of SETI@home, https://setiathome.berkeley.edu/, last updated 2021

Anderson, D. J. (2010), Kanban, Berlin 2010

Anderson, D. J. and Carmichael, A. (2016), Essential kanban condensed, Berlin 2016

Anderson, R. (2008), Security Engineering: A Guide to Building Dependable Distributed Systems, 3rd Edition, Boston and New York 2008

Andrikopoulos, V., Binz, T., Leymann, F. and Strauch, S. (2013a), How to adapt applications for the Cloud environment: Challenges and solutions in migrating applications to the Cloud, in: Computing, vol. 95, 2013, iss. 6, pp. 493–535

Andrikopoulos, V., Song, Z. and Leymann, F. (2013b), Supporting the Migration of Applications to the Cloud through a Decision Support System, in: Proceedings of the 6th International IEEE Conference on Cloud Computing (CLOUD 2013), pp. 565–572

Anthony, B. M. and Cantu Jr., V. (2018), Modernizing the mythical man-month, in: Journal of Computing Sciences in Colleges, vol. 34, 2018, iss. 2, pp. 110–116

Apache Software Foundation (2015), Apache Delta Cloud, on the website of Apache, http://deltacloud.apache.org/, last updated 20.07.2015

Apache Software Foundation (2021), Apache Nuvem, on the website of Apache, http://wiki.apache.org/incubator/Nuvem/, last updated 10.04.2019

Apache Software Foundation (2022a), Apache Brooklyn, on the website of Apache, https://brooklyn.incubator.apache.org/, last updated 2022

Apache Software Foundation (2022b), Apache Hadoop, on the website of Apache, http://hadoop.apache.org/, last updated 08.08.2022

Apache Software Foundation (2022c), Apache Libcloud, on the website of Apache, http://libcloud.apache.org, last updated 08.01.2022

Apache Software Foundation (2023), Apache Tomcat, on the website of Apache, http://tomcat.apache.org/, last updated 19.01.2023

Arcelli Fontana, F., Raibulet, C. and Zanoni, M. (2017), Alternatives to the knowledge discovery metamodel: An investigation, in: International Journal of Software Engineering and Knowledge Engineering, vol. 27, 2017, iss. 7, pp. 1097–1128

Ardagna, D., Ciavotta, M., Lancellotti, R. and Guerriero, M. (2018), A Hierarchical Receding Horizon Algorithm for QoS-driven control of Multi-IaaS Applications, in: IEEE Transactions on Cloud Computing, pp. 1–32

Ardagna, D., Ciavotta, M. and Passacantando, M. (2017), Generalized nash equilibria for the service provisioning problem in multi-cloud systems, in: IEEE Transactions on Services Computing, vol. 10, 2017, iss. 3, pp. 381–395

Arjuna (2010), Arjuna Agility: Removing the Barriers to Business Agility, Technical Report Arjuna Department of Informatics University of Zurich, Newcastle upon Tyne 2010

Armbrust, M., Fox, A., Griffith, R., Joseph, A. D., Katz, R., Konwinski, A., Lee, G., Patterson, D., Rabkin, A. and Stoica, I. (2010), A view of Cloud Computing, in: Communications of the ACM, vol. 53, 2010, iss. 4, pp. 50–58

ARTIST Consortium (2021), ARTIST Advanced software-based service provisioning and migration of legacy software

Assunção, W. K., Lopez-Herrejon, R. E., Linsbauer, L., Vergilio, S. R. and Egyed, A. (2017), Reengineering legacy applications into software product lines: a systematic mapping, in: Empirical Software Engineering, vol. 22, 2017, iss. 6, pp. 2972–3016

Ateniese, G., Burns, R., Curtmola, R., Herring, J., Kissner, L., Peterson, Z. and Song, D. (2007), Provable data possession at untrusted stores, in: Proceedings of the 14th ACM conference on Computer and Communications Security (CCS 2007), pp. 598–609

Atkinson, C. and Kühne, T. (2002), Rearchitecting the UML infrastructure, in: ACM Transactions on Modeling and Computer Simulation (TOMACS 2002), vol. 12, 2002, iss. 4, pp. 290–321

Atkinson, C. and Kühne, T. (2003), Model-driven development: a metamodeling foundation, in: IEEE software, vol. 20, 2003, iss. 5, pp. 36–41

Atteslander, P. (2003), Methoden der empirischen Sozialforschung, 10th Edition, Berlin and New York 2003

Autili, M., Di Ruscio, D., Paola, I., Massimo, T., Athanasopoulos, D., Zarras, A., Vassiliadis, P., Lockerbie, J., Maiden, N., Bertolino, A., Angelis, G. D., Ben Amida, A., Silingas, D., Bartkevicius, R., Gerosa, M. A., Oliva, G. A., Leite, L., M. Nogueira, G., Goldman, A. and Ngoko, Y. (2011), CHOReOS Dynamic Development Model Definition (D2.1), Technical Report CHOReOS consortium, Rocquencourt 2011

Avetisyan, A. I., Campbell, R., Lai, K., Lyons, M., Milojicic, D. S., Lee, H. Y., Soh, Y. C., Ming, N. K., Luke, J.-Y. and Namgoong, H. (2010), Open cirrus: A global cloud computing testbed, in: Computer, vol. 43, 2010, iss. 4, pp. 35–43

Avison, D. E., Lau, F., Myers, M. D. and Nielsen, P. A. (1999), Action research, in: Communications of the ACM, vol. 42, 1999, iss. 1, pp. 94–97

Baars, H., Arnold, T.-M. and Kemper, H.-G. (2010), Ansätze für eine differenzierte Business Intelligence Governance, in: Multikonferenz Wirtschaftsinformatik 2010, pp. 1065–1076

Baars, H. and Ereth, J. (2016), From Data Warehouses to Analytical Atoms–The Internet of Things as a Centrifugal Force in Business Intelligence and Analytics, in: Proceedings of the 24th European Conference on Information Systems (ECIS 2016), pp. 1–18

Baars, H. and Kemper, H.-G. (2010), Business intelligence in the cloud?, in: Pacific Asia Conference on Information Systems (PACIS 2010), pp. 1528–1539

Baars, H. and Kemper, H.-G. (2011), Ubiquitous Computing—an Application Domain for Business Intelligence in the Cloud?, in: A Renaissance of Information Technology for Sustainability and Global Competitiveness, 17th Americas Conference on Information Systems (AMCIS 2011), pp. 1–9

Baars, H. and Kemper, H.-G. (2008), Management support with structured and unstructured data? An integrated business intelligence framework, in: Information Systems Management, vol. 25, 2008, iss. 2, pp. 132–148

Baars, H. and Qie, L. (2012), BI in the Cloud–Die Cloud als neuer Ansatz zur Erhöhung der BI-Agilität, in: BI Spektrum, vol. 7, 2012, iss. 2, pp. 26–29

Babar, M. A. and Chauhan, M. A. (2011), A tale of migration to cloud computing for sharing experiences and observations, in: Proceedings of the 2nd international workshop on software engineering for cloud computing, pp. 50–56

Balobaid, A. S. (2020), Cloud Provisioning and Migration Strategies for Small and Medium Enterprises, PhD Dissertation at Oakland University, Oakland 2020

Baranwal, G., Kumar, D., Raza, Z. and Vidyarthi, D. P. (2018), A negotiation based dynamic pricing heuristic in cloud computing, in: International Journal of Grid and Utility Computing, vol. 9, 2018, iss. 1, pp. 83–96

Bass, L., Clements, P. and Kazman, R. (2021), Software Architecture in Practice, SEI Series in Software Engineering, Boston, San Francisco et al. 2021

Bauer, S., Bernroider, E. W. N. and Chudzikowski, K. (2017), Prevention is better than cure! Designing information security awareness programs to overcome users' non-compliance with information security policies in banks, in: Computers & Security, vol. 68, 2017, pp. 145–159

Beck, K. (2003), Test-driven development: by example, Boston, San Francisco et al. 2003

Becker, J., Niehaves, B., Olbrich, S. and Pfeiffer, D. (2009a), Forschungsmethodik einer Integrationsdisziplin: Eine Fortführung und Ergänzung zu Lutz Heinrichs „Beitrag zur Geschichte der Wirtschaftsinformatik" aus gestaltungsorientierter Perspektive, in: Wissenschaftstheorie und gestaltungsorientierte Wirtschaftsinformatik, pp. 1–22

Becker, S., Koziolek, H. and Reussner, R. (2009b), The Palladio component model for model-driven performance prediction, in: Journal of Systems and Software, vol. 82, 2009, iss. 1, pp. 3–22

Bellman, K., Botev, J., Diaconescu, A., Esterle, L., Gruhl, C., Landauer, C., Lewis, P. R., Nelson, P. R., Pournaras, E., Stein, A. and Tomforde, S. (2021), Self-improving system integration: Mastering continuous change, in: Future Generation Computer Systems, vol. 117, 2021, pp. 29–46

Bello, A. and Mahadevan, V. (2019), A Cloud Based Conceptual Identity Management Model for Secured Internet of Things Operation, in: Journal of Cyber Security and Mobility, vol. 8, 2019, iss. 1, pp. 53–74

Benavides, D., Trinidad, P. and Ruiz-Cortés, A. (2005), Automated reasoning on feature models, in: Advanced Information Systems Engineering, pp. 491–503

Bennasar, H., Bendahmane, A. and Essaaidi, M. (2017), An overview of the state-of-the-art of cloud computing cyber-security, in: International Conference on Codes, Cryptology, and Information Security, pp. 56–67

Bennett, K. (1995), Legacy systems: coping with stress, in: IEEE software, vol. 12, 1995, iss. 1, pp. 19–23

Bennett, K. and Rajlich, V. T. (1999), A new perspective on software evolution: the staged model, in: IEEE software, pp. 1–7

Bennett, K. and Rajlich, V. T. (2000), Software maintenance and evolution: a roadmap, in: Proceedings of the Conference on the Future of Software Engineering (FOSE 2000), pp. 73–87

Benslimane, D., Dustdar, S. and Sheth, A. (2008), Services Mashups, in: New Generation of Web Applications (NGWA 2008), pp. 13–15

Bergmayr, A., Breitenbücher, U., Ferry, N., Rossini, A., Solberg, A., Wimmer, M., Kappel, G. and Leymann, F. (2018), A systematic review of cloud modeling languages, in: ACM Computing Surveys (CSUR), vol. 51, 2018, iss. 1, pp. 1–22

Bergmayr, A., Bruneliere, H., Canovas Izquierdo, J. L., Gorronogoitia, J., Kousiouris, G., Kyriazis, D., Langer, P., Menychtas, A., Orue-Echevarria, L. and Pezuela, C. (2013), Migrating legacy software to the cloud with ARTIST, in: 17^{th} European Conference on Software Maintenance and Reengineering (CSMR 2013), pp. 465–468

Bernstein, D. and Demchenko, Y. (2013), The IEEE Intercloud Testbed–Creating the Global Cloud of Clouds, in: Proceedings of the 5^{th} International IEEE Conference on Cloud Computing Technology and Science (CloudCom 2013), pp. 45–50

Bernstein, D., Ludvigson, E., Sankar, K., Diamond, S. and Morrow, M. (2009), Blueprint for the intercloud-protocols and formats for cloud computing interoperability, in: Proceedings of the 4^{th} International Conference on Internet and Web Applications and Services (ICIW 2009), pp. 328–336

Bernstein, D. and Vij, D. (2010a), Intercloud directory and exchange protocol detail using XMPP and RDF, in: Proceedings of the 6th World Congress on Services (SERVICES 2010), pp. 431–438

Bernstein, D. and Vij, D. (2010b), Simple storage replication protocol (SSRP) for intercloud, in: Proceedings of the 2nd International Conference on Emerging Network Intelligence, pp. 30–37

Bernstein, D. and Vij, D. (2010c), Using Semantic Web Ontology for Intercloud Directories and Exchanges. In: International Conference on Internet Computing (ICIC 2010), pp. 18–24

Bernstein, D., Vij, D. and Diamond, S. (2011), An intercloud cloud computing economy-technology, governance, and market blueprints, in: SRII Global Conference (SRII 2011), pp. 293–299

Bhardwaj, A. and Krishna, C. R. (2021), Virtualization in cloud computing: Moving from hypervisor to containerization—a survey, in: Arabian Journal for Science and Engineering, vol. 46, 2021, iss. 9, pp. 8585–8601

Billings, J.-J., Bennett, A. R., Deyton, J., Gammeltoft, K., Graham, J., Gorin, D., Krishnan, H., Li, M., McCaskey, A. J. and Patterson, T. (2018), The eclipse integrated computational environment, in: SoftwareX, vol. 7, 2018, pp. 234–244

Binz, T., Breitenbücher, U., Haupt, F., Kopp, O., Leymann, F., Nowak, A. and Wagner, S. (2013), OpenTOSCA: A Runtime for TOSCA-based Cloud Applications, in: Service-Oriented Computing, pp. 692–695

Binz, T., Leymann, F. and Schumm, D. (2011), CMotion: A Framework for Migration of Applications into and between clouds, in: Proceedings of the International IEEE Conference on Service-Oriented Computing and Applications (SOCA 2011), pp. 1–4

Birman, K., Chockler, G. and Renesse, R. van (2009), Toward a cloud computing research agenda, in: ACM SIGACT News, vol. 40, 2009, iss. 2, pp. 68–80

Bisbal, J., Lawless, D., Wu, B. and Grimson, J. (1999a), Legacy Information System Migration: A Brief Review of Problems, Solutions and Research Issues, Recommendation ITU-T H.262, ISO/IEC Video, Technical Report International Telecommunication Union, Geneva 1999

Bisbal, J., Lawless, D., Wu, B. and Grimson, J. (1999b), Legacy information systems: Issues and directions, in: IEEE software, vol. 16, 1999, iss. 5, pp. 103–111

Bisbal, J., Lawless, D., Wu, B., Grimson, J., Wade, V., Richardson, R. and O'Sullivan, D. (1997), A survey of research into legacy system migration, in: Technique report, pp. 1–39

Blasi, L., Jensen, J. and Ziegler, W. (2013), Expressing quality of service and protection using federation-level service level agreement, in: European Conference on Parallel Processing (Euro-Par 2013), pp. 146–156

Blau, B., Neumann, D., Weinhardt, C. and Lamparter, S. (2008), Planning and pricing of service mashups, in: 10^{th} IEEE Conference on E-Commerce Technology and the 5^{th} IEEE Conference on Enterprise Computing, E-Commerce and E-Services (CEC/EEE 2008), pp. 19–26

Boehm, B. W. (1976), Software Engineering, in: IEEE Trans. Computers, vol. 25, 1976, iss. 12, pp. 1226–1241

Bomar, K. B. and Harper, G. E. (2019), Tokenized data security, Technical Report US Patent Application, Washington 2019

Botto-Tobar, M. and Insfran, E. (2017), A Proposal for Migrating SOA Applications to Cloud Using Model-Driven Development, in: International Conference on Technology Trends (CITT 2017), pp. 171–184

Bouguettaya, A., Singh, M., Huhns, M., Sheng, Q. Z., Dong, H., Yu, Q., Neiat, A. G., Mistry, S., Benatallah, B. and Medjahed, B. (2017), A service computing manifesto: the next 10 years, in: Communications of the ACM, vol. 60, 2017, iss. 4, pp. 64–72

Bourque, P., Dupuis, R., Abran, A., Moore, J. W. and Tripp, L. (1999), The guide to the software engineering body of knowledge, in: IEEE software, vol. 16, 1999, iss. 6, pp. 35–44

Bowers, K. D., Van Dijk, M., Juels, A., Oprea, A. and Rivest, R. L. (2011), How to tell if your cloud files are vulnerable to drive crashes, in: Proceedings of the 18th ACM Conference on Computer and Communications Security (CCS 2011), pp. 501–514

BPR4GDPR Consortium (2021), BPR4GDPR Business Process Re-engineering and functional toolkit for GDPR compliance, EU H2020 Project, tech. rep.

Brambilla, M., Cabot, J. and Wimmer, M. (2017), Model-driven software engineering in practice, in: Synthesis Lectures on Software Engineering, vol. 3, 2017, iss. 1, pp. 1–207

Brandenburger, M., Cachin, C. and Knežević, N. (2015), Don't trust the cloud, verify: Integrity and consistency for cloud object stores, in: Proceedings of the 8th International ACM Systems and Storage Conference (SYSTOR 2015), pp. 1–16

Brandenburger, M., Cachin, C. and Knežević, N. (2017), Don't trust the cloud, verify: Integrity and consistency for cloud object stores, in: ACM Transactions on Privacy and Security (TOPS 2017), vol. 20, 2017, iss. 3, pp. 1–8

Brandtzæg, E., Parastoo, M. and Mosser, S. (2012), Towards a domain-specific language to deploy applications in the clouds, in: Proceedings of the 3rd International Conference on Cloud Computing, GRIDs, and Virtualization (CLOUD COMPUTING 2012), pp. 213–218

Brataas, G., Becker, S., Lehrig, S., Huljenić, D., Kopcak, G. and Stupar, I. (2016), The CloudScale method: A white paper, Technical Report SINTEF ICT, Oslo 2016

Brataas, G., Stav, E., Lehrig, S., Becker, S., Kopčak, G. and Huljenic, D. (2013), CloudScale: scalability management for cloud systems, in: Proceedings of the 4th ACM/SPEC International Conference on Performance Engineering (ICPE 2013), pp. 335–338

Brettel, M., Friederichsen, N., Keller, M. and Rosenberg, M. (2014), How virtualization, decentralization and network building change the manufacturing landscape: An Industry 4.0 Perspective, in: International Journal of Mechanical, Industrial Science and Engineering, vol. 8, 2014, iss. 1, pp. 37–44

Brewster, E., Griffiths, R., Lawes, A. and Sansbury, J. (2012), IT service management: a guide for ITIL foundation exam candidates, 2^{nd} Edition, Swindon 2012

Brinkkemper, S. (2016), Classifying information systems education by method engineering, in: Informatics in Higher Education, pp. 1–235

Brinkkemper, S., Lyytinen, K. and Welke, R. (1996), Method Engineering: Principles of Method Construction and Tool Support, London 1996

Broberg, J., Venugopal, S. and Buyya, R. (2008), Market-oriented grids and utility computing: The state-of-the-art and future directions, in: Journal of Grid Computing, vol. 6, 2008, iss. 3, pp. 255–276

Brodie, M. L. and Stonebraker, M. (1995), Migrating legacy systems: gateways, interfaces & the incremental approach, 1^{st} Edition, Burlington 1995

Bromberg, Y.-D., Grace, P. and Réveillère, L. (2011), Starlink: runtime interoperability between heterogeneous middleware protocols, in: 31^{st} International Conference on Distributed Computing Systems (ICDCS 2011), pp. 446–455

Brooks, F. P. (1975), The mythical man-month, Boston 1975

Bruneliere, H. (2011), MoDisco in a Nutshell! How to Deal with your IT Legacy? Reverse Engineering using Models, in: Java TECH, pp. 21–24

Bruneliere, H. (2018), Generic Model-based Approaches for Software Reverse Engineering and Comprehension, PhD Dissertation at the Université de Nantes of the Comue Université Bretagne Loire, Nantes 2018

Bruneliere, H., Cabot, J., Dupé, G. and Madiot, F. (2014), Modisco: A model driven reverse engineering framework, in: Information and Software Technology, vol. 56, 2014, iss. 8, pp. 1012–1032

Bukowski, L. (2016), System of systems dependability–Theoretical models and applications examples, in: Reliability Engineering & System Safety, vol. 151, 2016, pp. 76–92

Buyya, R., Abramson, D., Giddy, J. and Stockinger, H. (2002), Economic models for resource management and scheduling in grid computing, in: Concurrency and computation: practice and experience, vol. 14, 2002, pp. 1507–1542

Buyya, R., Broberg, J. and Goscinski, A. M. (2010a), Cloud computing: Principles and paradigms, Hoboken 2010

Buyya, R., Ranjan, R. and Calheiros, R. N. (2010b), Intercloud: Utility-oriented federation of cloud computing environments for scaling of application services, in: International Conference on Algorithms and Architectures for Parallel Processing, pp. 13–31

Buyya, R., Yeo, C. S. and Venugopal, S. (2008), Market-oriented cloud computing: Vision, hype, and reality for delivering it services as computing utilities, in: 10th International IEEE Conference on High Performance Computing and Communications (HPCC 2008), pp. 5–13

Buyya, R., Yeo, C. S., Venugopal, S., Broberg, J. and Brandic, I. (2009), Cloud Computing and emerging IT platforms: Vision, hype, and reality for delivering Computing as the 5th utility, in: Future Generation computer systems, vol. 25, 2009, iss. 6, pp. 599–616

Byrne, E. J. (1991), Software reverse engineering: a case study, in: Software: Practice and Experience, vol. 21, 1991, iss. 12, pp. 1349–1364

Byrne, E. J. (1992), A conceptual foundation for software re-engineering, in: Proceerdings of the Conference on Software Maintenance (ICSM 1992), pp. 226–235

Cachin, C., Keidar, I. and Shraer, A. (2009), Trusting the cloud, in: ACM Special Interest Group on Algorithms and Computation Theory (SIGACT), vol. 40, 2009, iss. 2, pp. 81–86

Cachin, C., Shelat, A. and Shraer, A. (2007), Efficient fork-linearizable access to untrusted shared memory, in: Proceedings of the 26th annual ACM symposium on Principles of distributed computing, pp. 129–138

Calheiros, R. N., Ranjan, R., Beloglazov, A., De Rose, C. A. and Buyya, R. (2011), CloudSim: a toolkit for modeling and simulation of Cloud Computing environments and evaluation of resource provisioning algorithms, in: Software: Practice and Experience, vol. 41, 2011, iss. 1, pp. 23–50

Calheiros, R. N., Toosi, A. N., Vecchiola, C. and Buyya, R. (2012), A coordinator for scaling elastic applications across multiple clouds, in: Future Generation Computer Systems, vol. 28, 2012, iss. 8, pp. 1350–1362

Calinescu, R., Weyns, D., Gerasimou, S., Iftikhar, M. U., Habli, I. and Kelly, T. (2018), Engineering trustworthy self-adaptive software with dynamic assurance cases, in: IEEE Transactions on Software Engineering (TSE), vol. 44, 2018, iss. 11, pp. 1039–1069

Camarinha-Matos, L. M. and Afsarmanesh, H. (2007), A framework for virtual organization creation in a breeding environment, in: Annual Reviews in Control, vol. 31, 2007, iss. 1, pp. 119–135

Camarinha-Matos, L. M., Afsarmanesh, H. and Ollus, M. (2008), Methods and Tools for Collaborative Networked Organizations, New York 2008

Camarinha-Matos, L. M., Juan-Verdejo, A., Alexakis, S., Bar, H. and Surajbali, B. (2015), Cloud-based collaboration spaces for enterprise networks, in: International Conference on Computing and Communications Technologies (ICCCT 2015), pp. 185–190

Campbell, R., Gupta, I., Heath, M., Ko, S. Y., Kozuch, M., Kunze, M., Kwan, T., Lai, K., Lee, H. Y. and Lyons, M. (2009), Open cirrus™ cloud computing testbed: federated data centers for open source systems and services research, in: Proceedings of the 1st USENIX conference on Hot topics in cloud computing (HotCloud 2009), pp. 1–5

Canfora, G. and Di Penta, M. (2007), New frontiers of reverse engineering, in: Proceedings of the Conference on the Future of Software Engineering (FOSE 2007), pp. 326–341

Canfora, G., Di Penta, M. and Cerulo, L. (Apr. 2011), Achievements and Challenges in Software Reverse Engineering, in: Communications of the ACM, vol. 54, 2011, iss. 4, pp. 142–151

Canfora, G., Di Penta, M., Esposito, R. and Villani, M. L. (2005), An approach for QoS-aware service composition based on genetic algorithms, in: Proceedings of the 7th annual conference on Genetic and evolutionary computation (GECC0 2005), pp. 1069–1075

Canfora, G., Fasolino, A. R., Frattolillo, G. and Tramontana, P. (2008), A wrapping approach for migrating legacy system interactive functionalities to service oriented architectures, in: Journal of Systems and Software, vol. 81, 2008, iss. 4, pp. 463–480

Cantor, S., Moreh, J., Philpott, S. R. and Maler, E. (2005), Metadata for the OASIS Security Assertion Markup Language (SAML) V2. 0, Technical Report Standard OASIS, Burlington 2005, pp. 1–46

Cao, K., Liu, Y., Meng, G. and Sun, Q. (2020), An overview on edge computing research, in: IEEE access, vol. 8, 2020, pp. 85714–85728

Cao, L. and Ramesh, B. (2008), Agile requirements engineering practices: An empirical study, in: IEEE Software, vol. 25, 2008, iss. 1, pp. 60–67

Carlini, E., Coppola, M., Dazzi, P. and Mordacchini, M. (2016), Challenges to be addressed for realising an Ephemeral Cloud Federation, Technical Report Istituto di Scienza e Tecnologie dell'Informazione-Consiglio Nazionale delle Ricerche, Pisa 2016

Carlini, E., Coppola, M., Dazzi, P., Ricci, L. and Righetti, G. (2012), Cloud federations in contrail, in: European Conference on Parallel Processing (Euro-Par 2011), pp. 159–168

Carr, N. G. (2008), The big switch: Rewiring the world, from Edison to Google, New York 2008

Catteddu, D. (2009), Cloud Computing: benefits, risks and recommendations for information security, in: Iberic Web Application Security Conference (IBWAS 2009), pp. 17–17

Chang, F., Dean, J., Ghemawat, S., Hsieh, W. C., Wallach, D. A., Burrows, M., Chandra, T., Fikes, A. and Gruber, R. E. (2008), Bigtable: A distributed storage system for structured data, in: ACM Transactions on Computer Systems (TOCS 2008), vol. 26, 2008, iss. 2, pp. 1–4

Chang, V., Kuo, Y.-H. and Ramachandran, M. (2016a), Cloud computing adoption framework: A security framework for business clouds, in: Future Generation Computer Systems, vol. 57, 2016, pp. 24–41

Chang, V. and Ramachandran, M. (2015), Towards achieving data security with the cloud computing adoption framework, in: IEEE Transactions on Services Computing, vol. 9, 2015, iss. 1, pp. 138–151

Chang, V., Ramachandran, M., Yao, Y., Kuo, Y.-H. and Li, C.-S. (2016b), A resiliency framework for an enterprise cloud, in: International Journal of Information Management, vol. 36, 2016, iss. 1, pp. 155–166

Chang, V., Walters, R. J. and Wills, G. B. (2015), Cloud Computing and Frameworks for Organisational Cloud Adoption, in: Delivery and Adoption of Cloud Computing Services in Contemporary Organizations, vol. 1, 2015, pp. 1–28

Chapin, N., Hale, J. E., Khan, K. M., Ramil, J. F. and Tan, W.-G. (2001), Types of software evolution and software maintenance, in: Journal of software maintenance and evolution: Research and Practice, vol. 13, 2001, iss. 1, pp. 3–30

Chen, Q. (2011), Towards energy-aware VM scheduling in IaaS clouds through empirical studies, Master Thesis at the University of Amsterdam, Amsterdam 2011

Chen, Q., Grosso, P., Van Der Veldt, K., Laat, C. de, Hofman, R. and Bal, H. (2011), Profiling energy consumption of VMs for green cloud computing, in: Proceedings of the 9th International IEEE Conference on Dependable, Autonomic and Secure Computing (DASC 2011), pp. 768–775

Cheung, A. S. and Weber, R. H. (2015), Data protection regulation and cloud computing, in: Privacy and Legal Issues in Cloud Computing, vol. 2, 2015, pp. 26–42

Chikofsky, E. J. and Cross, J. H. (1990), Reverse engineering and design recovery: A taxonomy, in: IEEE software, vol. 7, 1990, iss. 1, pp. 13–17

Chokhani, S., Ford, W., Sabett, R., Merrill, C. and Wu, S. (2003), Internet X. 509 public key infrastructure certificate policy and certification practices framework, Technical Report at the Internet Society, Reston 2003

Chong, F. and Carraro, G. (2006), Architecture strategies for catching the long tail, Technical Report MSDN Library Microsoft Corporation, Seatle 2006

Christensen, E., Curbera, F., Meredith, G. and Weerawarana, S. (2001), Web services description language (WSDL) 1.1, Technical Report World Wide Web Consortium (W3C), Cambridge MA 2001

Chun, B.-G., Maniatis, P., Shenker, S. and Kubiatowicz, J. (2007), Attested append-only memory: Making adversaries stick to their word, in: ACM SIGOPS Operating Systems Review, vol. 41, 2007, iss. 6, pp. 189–204

Ciriello, R. F., Richter, A. and Schwabe, G. (2018), Digital innovation, in: Business & Information Systems Engineering, vol. 60, 2018, iss. 6, pp. 563–569

Claycomb, W. R. and Nicoll, A. (2012), Insider threats to cloud computing: Directions for new research challenges, in: Proceedings of the 36th International Annual IEEE Computer Software and Applications Conference (COMPSAC 2012), pp. 387–394

Clearwater, S. H. and Huberman, B. A. (2005), Swing options: a mechanism for pricing IT peak demand, in: International Conference on Computing in Economics and Finance (ICCEF 2005), pp. 1–21

Cloud Management Initiative (2021), Cloud Infrastructure Management Interface (CIMI) DMTF Standards, on the website of DMTF, http://dmtf.org/standards/cloud, last updated 14.12.2012

Cloud-Security-Alliance (2009), Security guidance for critical areas of focus in cloud computing v3.0, Technical Report Cloud Security Alliance (CSA), St. Petersburg FL 2009

Cloudscale Consortium (2015), CloudScale project, FP7 ICT research project SINTEF, Olso 2015

CloudScale-Consortium (2021), CloudScale Environment, FP7 ICT research project result SINTEF, Olso 2015

Cohn, J. M., Nair, S. P., Panikkar, S. B. and Pureswaran, V. S. (2018), Device self-servicing in an autonomous decentralized peer-to-peer environment, Technical Report US Patent Application, Washington 2018, pp. 1–17

Cohn, M. (2004), User stories applied: For agile software development, Boston, San Francisco et al. 2004

Congress-US, 115th (2018), Clarifying Lawful Overseas Use of Data Act or CLOUD Act, US federal law, Washington 2018

Contrail Consortium (2011), Contrail open-source release 2.0: Open Computing Infrastructures for Elastic Services, on the website of the contrail project, https://contrail.ow2.org/view/Documentation/InstallationContrailSecurityServices, last updated 2011

Contrail Consortium (2012), Overview of the Contrail System, Components and Usage, Technical Report White Paper Contrail consortium, Bucharest 2012

Contrail-Consortium (2014a), Contrail: Assembly of Large Genomes using Cloud Computing, on the website of sourceforge, http://sourceforge.net/projects/contrail-bio/, last updated 30.04.2014

Contrail-Consortium (2014b), Contrail Project: Open Computing Infrastructures for Elastic Services, FP7 ICT research project INRIA, Rennes 2014

Cook, S., He, J. and Harrison, R. (2001), Dynamic and static views of software evolution, in: Proceedings of the International IEEE Conference on Software Maintenance (ICSM 2001), pp. 592–601

Cooper, H. M. (1998), Synthesizing research: A guide for literature reviews, New York 1998

Coppola, M., Dazzi, P., Lazouski, A., Martinelli, F., Mori, P., Jensen, J., Johnson, I. and Kershaw, P. (2012), The Contrail approach to cloud federations, in: Proceedings of the International Symposium on Grids and Clouds (ISGC 2012), vol. 2, 2012, pp. 1–14

Coppolino, L., D'Antonio, S., Mazzeo, G. and Romano, L. (2017), Cloud security: Emerging threats and current solutions, in: Computers & Electrical Engineering, vol. 59, 2017, pp. 126–140

Corcho, O., Fernández-López, M. and Gómez-Pérez, A. (2006), Ontological engineering: principles, methods, tools and languages, in: Ontologies for software engineering and software technology, vol. 1, 2006, pp. 1–48

Costanzo, A. d., De Assunção, M. D. and Buyya, R. (2009), Harnessing cloud technologies for a virtualized distributed computing infrastructure, in: IEEE internet computing, vol. 13, 2009, iss. 5, pp. 24–33

Coutinho, E. F., Carvalho Sousa, F. R. de, Rego, P. A. L., Gomes, D. G. and Souza, J. N. de (2015), Elasticity in cloud computing: a survey, in: annals of telecommunications-annales des télécommunications, vol. 70, 2015, pp. 289–309

Creese, S., Goldsmith, M. and Hopkins, P. (2013), Inadequacies of current risk controls for the cloud, in: Privacy and Security for Cloud Computing, vol. 1, 2013, pp. 235–255

Creswell, J. W. (2013), Research design: Qualitative, quantitative, and mixed methods approaches, 5^{th} Edition, New York 2013

Crosby, S., Doyle, R., Gering, M., Gionfriddo, M., Grarup, S., Hand, S., Hapner, M. and Hiltgen, D. (2010), Open virtualization format specification: DSP0243, in

Crotty, J. and Horrocks, I. (2017), Managing legacy system costs: A case study of a meta-assessment model to identify solutions in a large financial services company, in: Applied computing and informatics, vol. 13, 2017, iss. 2, pp. 175–183

Cutillo, L. A., Molva, R. and Önen, M. (2012), Privacy preserving picture sharing: Enforcing usage control in distributed on-line social networks, in: Proceedings of the 5^{th} Workshop on Social Network Systems, pp. 1–6

Czarnecki, K., Helsen, S. and Eisenecker, U. (2005), Formalizing cardinality-based feature models and their specialization, in: Software process: Improvement and practice, vol. 10, 2005, iss. 1, pp. 7–29

De Cerqueira Leite Duboc, A. L. (2010), A framework for the characterization and analysis of software systems scalability, PhD Dissertation at the University College London (UCL), London 2010

De Lemos, R., Giese, H., Müller, H. A., Shaw, M., Andersson, J., Litoiu, M., Schmerl, B., Tamura, G., Villegas, N. M. and Vogel, T. (2013), Software engineering for self-adaptive systems: A second research roadmap, Technical Report Research Roadmap, Daghstuhl Wadern 2013

De Lucia, A., Di Lucca, G. A., Fasolino, A. R., Guerra, P. and Petruzzelli, S. (1997), Migrating legacy systems towards object-oriented platforms, in: Proceedings of the International Conference on Software Maintenance, (ICSM 1997), pp. 122–129

Denzin, N. K. and Lincoln, Y. S. (2011), The SAGE handbook of qualitative research, New York 2011

Deshpande, P., Sharma, S. C., Peddoju, S. K. and Abraham, A. (2018), Security and service assurance issues in Cloud environment, in: International Journal of System Assurance Engineering and Management, vol. 9, 2018, iss. 1, pp. 194–207

Devlin, B., Gray, J., Laing, B. and Spix, G. (1999), Scalability terminology: Farms, clones, partitions, packs: RACS and RAPS, Technical Report Microsoft research, Redmond WA 1999

Di Nitto, E., Casale, G. and Petcu, D. (2016), On MODAClouds' Toolkit Support for DevOps, in: Advances in Service-Oriented and Cloud Computing: Workshop of the 4th ACM European Conference on Service-Oriented and Cloud Computing (ESOCC 2015), vol. 567, 2016, pp. 1–430

Dias Canedo, E., Oliveira Albuquerque, R. de and Sousa Junior, R. Timóteo de (2011), Review of Trust-based File Sharing in Cloud Computing, in: The 4th International Conference on Advances in Mesh Networks (MESH 2011), pp. 44–50

Dick, J., Hull, E. and Jackson, K. (2017), Requirements engineering, New York 2017

Diehl, S. (2007), Software visualization: visualizing the structure, behaviour, and evolution of software, New York 2007

Dintzner, N., Deursen, A. van and Pinzger, M. (2018), FEVER: An approach to analyze feature-oriented changes and artefact co-evolution in highly configurable systems, in: Empirical Software Engineering, vol. 23, 2018, iss. 2, pp. 1–48

DOD (2013), Cyber Security and Reliability in a Digital Cloud by the US Department of Defense, Arlington VA 2013

Dolata, M. and Schwabe, G. (2016), Design Thinking in IS Research Projects, in: Design Thinking for Innovation, vol. 3, 2016, pp. 67–83

Draper, P. J., Galloway, K. D., Hyde, J., Li, Y., Liu, L., Ma, Q., Ratakonda, K. C. and Sun, K. (2013), Method for optimizing migration of software applications to address needs, Technical Report US Patent Application, Washington 2013

Dresch, A., Lacerda, D. P. and Antunes, J. A. V. (2015), Design science research, in: Design Science Research, vol. 3, 2015, pp. 67–102

Duan, Y. (2012), Value Modeling and Calculation for Everything as a Service (XaaS) based on Reuse, in: 13th International ACIS Conference on Software Engineering, Artificial Intelligence, Networking and Parallel & Distributed Computing (SNPD 2012), pp. 162–167

Duboc, L., Rosenblum, D. and Wicks, T. (2007), A framework for characterization and analysis of software system scalability, in: Proceedings of the 6th joint meeting of the European software engineering conference and the ACM SIGSOFT symposium on the foundations of software engineering, pp. 375–384

Duncan, A. J., Creese, S. and Goldsmith, M. (2012), Insider attacks in cloud computing, in: 11th International IEEE Conference on Trust, Security and Privacy in Computing and Communications (TrustCom 2012), pp. 857–862

Duncan, B. (2018), Can EU general data protection regulation compliance be achieved when using cloud computing?, in: The 9th International Conference on Cloud Computing, GRIDs, and Virtualization (CLOUD COMPUTING 2018), IARIA, pp. 1–6

Durkee, D. (2010), Why cloud computing will never be free, in: Queue, vol. 8, 2010, iss. 4, pp. 1–20

Dzombeta, S. (2016), Compliance framework for change management in cloud environments, PhD Dissertation at the Universidad Carlos III de Madrid, Leganés 2016

Easterbrook, S., Singer, J., Storey, M.-A. and Damian, D. (2008), Selecting empirical methods for software engineering research, in: Guide to advanced empirical software engineering, vol. 5, 2008, pp. 285–311

Eberlein, A. and Prado Leite, J. C. Sampaio do (2002), Agile requirements definition: A view from requirements engineering, in: Proceedings of the International Workshop on Time-Constrained Requirements Engineering (TCRE 2002), pp. 4–8

Eder, J., Olivotto, G. E. and Gruber, W. (2002), A data warehouse for workflow logs, in: Engineering and Deployment of Cooperative Information Systems, vol. 1, 2002, pp. 1–15

Eisa, M., Younas, M., Basu, K. and Awan, I. (2020), Modelling and simulation of qos-aware service selection in cloud computing, in: Simulation Modelling Practice and Theory, vol. 103, 2020, pp. 102–108

ElMalah, K. and Nasr, M. (2019), Cloud business intelligence, in: International Journal of Advanced Networking and Applications, vol. 10, 2019, iss. 6, pp. 4120–4124

Elson, J. and Howell, J. (2009), Refactoring human roles solves systems problems, in: Proceedings of the 1st USENIX conference on Hot topics in cloud computing (HotCloud 2009), pp. 1–5

EnStratius, Inc. (2013), EnStratius: Commercial solution keynote, on the website of Youtube, https://www.youtube.com/watch?v=G0w4aXGJjpQ, last updated 02.05.2013

Ereth, J. and Baars, H. (2015), Cloud-Based Business Intelligence and Analytics Applications-Business Value and Feasibility. In: Pacific Asia Conference on Information Systems (PACIS 2015), pp. 1–36

Ereth, J. and Baars, H. (2020), A Capability Approach for Designing Business Intelligence and Analytics Architectures, in: Proceedings of the 53rd Hawaii International Conference on System Sciences, pp. 1–10

ETSI (2016), Cloud Standards Coordination Phase 2; Interoperability and Security in Cloud Computing, Technical Report ETSI, Sophia-Antipolis 2016

European Union (2016), On the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation, Regulation EU 2016/679 of the European Parliament and of the Council), EU regulation, Brussels 2016

European Union (2019), Directive on Copyright in the Digital Single Market, Official Journal of the European Union, L 130, EU directive, Brussels 2019

Fazli, A., Sayedi, A. and Shulman, J. D. (2018), The effects of autoscaling in cloud computing, in: Management Science, vol. 64, 2018, iss. 11, pp. 5149–5163

Fehling, C., Leymann, F. and Mietzner, R. (2010), A framework for optimized distribution of tenants in Cloud applications, in: Proceedings of the 3rd International IEEE Conference on Cloud Computing (CLOUD 2010, pp. 252–259

Fellows, W. (2008), Partly Cloudy, Blue-Sky Thinking About Cloud Computing, Technical Report White paper 451 group, London 2008

Ferdiana, R. and Putra, G. D. (2018), A Review of Cloud Migration Strategies in the Developing Country, in: Proceedings of the 4th International Conference on Science and Technology (ICST 2018), pp. 1–6

Ferrer, A. J., Hernández, F., Tordsson, J., Elmroth, E., Ali-Eldin, A., Zsigri, C., Sirvent, R., Guitart, J., Badia, R. M. and Djemame, K. (2012), OPTIMIS: A holistic approach to cloud service provisioning, in: Future Generation Computer Systems, vol. 28, 2012, iss. 1, pp. 66–77

Ferri, L., Maffei, M., Mangia, G. and Tomo, A. (2017), Analyzing Cloud-based Startups: Evidence from a Case Study in Italy, in: International Business Research, vol. 10, 2017, iss. 5, pp. 73–85

Ferry, N., Chauvel, F., Rossini, A., Morin, B. and Solberg, A. (2013a), Managing multi-cloud systems with CloudMF, in: Proceedings of the 2nd Nordic Symposium on Cloud Computing & Internet Technologies (CCIT 2013), pp. 38–45

Ferry, N., Chauvel, F., Song, H., Rossini, A., Lushpenko, M. and Solberg, A. (2018), Cloudmf: Model-driven management of multi-cloud applications, in: ACM Transactions on Internet Technology (TOIT), vol. 18, 2018, iss. 2, pp. 1–16

Ferry, N., Rossini, A., Chauvel, F., Morin, B. and Solberg, A. (2013b), Towards Model-Driven Provisioning, Deployment, Monitoring, and Adaptation of Multi-cloud Systems, in: Proceedings of the 6th International Conference on Cloud Computing (CLOUD 2013), pp. 887–894

Ferry, N., Song, H., Rossini, A., Chauvel, F. and Solberg, A. (2014), Cloud MF: Applying MDE to Tame the Complexity of Managing Multi-cloud Applications, in: Proceedings of the 7th International IEEE/ACM Conference on Utility and Cloud Computing (UCC 2014), pp. 269–277

Ferstl, O. K. and Sinz, E. J. (2012), Grundlagen der Wirtschaftsinformatik, 5th Edition, München 2012

Fleurey, F., Steel, J. and Baudry, B. (2004), Validation in model-driven engineering: testing model transformations, in: Proceedings of the 1st International Workshop on Model, Design and Validation (MoDeVa 2004), pp. 29–40

Flexera (2017), Alerts, Alert Escalations, and Server Arrays, White paper

Flexera (2023), Flexera: Cloud management platform, Commercial Solution's Main Website, on the website of Flexera, https://www.flexera.com/products/cloud-management-platform, last updated 2023

Folkerts, E., Alexandrov, A., Sachs, K., Iosup, A., Markl, V. and Tosun, C. (2013), Benchmarking in the cloud: What it should, can, and cannot be, in: Selected Topics in Performance Evaluation and Benchmarking, vol. 775, 2013, pp. 173–188

Foster, I. and Kesselman, C. (1999), The Grid: Blueprint for a New Computing Architecture, Berlin 1999

Foster, I. and Tuecke, S. (2005), Describing the elephant: The different faces of IT as service, in: Queue, vol. 3, 2005, iss. 6, pp. 26–29

Foster, I., Zhao, Y., Raicu, I. and Lu, S. (2008), Cloud computing and grid computing 360-degree compared, in: Grid computing Environments Workshop (GCE 2008), pp. 1–10

Fox, A. and Brewer, E. (1999), Harvest, yield, and scalable tolerant systems, in: Proceedings of the 7th Workshop on Hot Topics in Operating Systems (HotOS 1999), pp. 174–178

Fox, A., Griffith, R., Joseph, A., Katz, R., Konwinski, A., Lee, G., Patterson, D., Rabkin, A. and Stoica, I. (2009), Above the clouds: A Berkeley view of cloud computing, Report UCB/EECS, Technical Report at the Department of Electrical Engineering and Computer Sciences University of California, Berkeley 2009

Frey, S. (2014), Conformance Checking and Simulation-based Evolutionary Optimization for Deployment and Reconfiguration of Software in the Cloud, PhD Dissertation at the Christian-Albrechts University, Kiel 2014

Frey, S., Fittkau, F. and Hasselbring, W. (2018), CDOXplorer: simulation-based genetic optimization of software deployment and reconfiguration in the cloud, Technical Report Kiel University Library, Kiel 2018

Fuhr, A., Horn, T., Riediger, V. and Winter, A. (2013), Model-driven software migration into service-oriented architectures, in: Computer Science-Research and Development, vol. 28, 2013, iss. 1, pp. 65–84

Fuhr, A., Winter, A., Erdmenger, U., Horn, T., Kaiser, U., Riediger, V. and Teppe, W. (2012), Model-Driven Software Migration: Process Model, Tool Support, in: Migrating Legacy Applications: Challenges in Service Oriented Architecture and Cloud Computing Environments, vol. 1, 2012, pp. 1–153

Ga, D., Djuric, D. and Deved, V. (2009), Model driven engineering and ontology development, 2nd ed., Heidelberg London New York 2009

Ganek, A. G. and Corbi, T. A. (2003), The dawning of the autonomic computing era, in: IBM systems Journal, vol. 42, 2003, iss. 1, pp. 5–18

Gangwar, H., Date, H. and Ramaswamy, R. (2015), Understanding determinants of cloud computing adoption using an integrated TAM-TOE model, in: Journal of Enterprise Information Management, vol. 28, 2015, iss. 1, pp. 107–130

Ganti, N. and Brayman, W. (1995), The transition of legacy systems to a distributed architecture, Michigan 1995

Gao, Q., Li, J., Xiong, Y., Hao, D., Xiao, X., Taneja, K., Zhang, L. and Xie, T. (2016), High-confidence software evolution, in: Science China Information Sciences, vol. 59, 2016, iss. 7, pp. 1–19

García, Á. L., Castillo, E. F. del and Fernández, P. O. (2016), Standards for enabling heterogeneous IaaS cloud federations, in: Computer Standards & Interfaces, vol. 47, 2016, pp. 19–23

García, A. G. and Blanquer, I. (2015), Cloud Services Representation using SLA Composition, in: Journal of Grid Computing, vol. 13, 2015, iss. 1, pp. 35–51

García-Galán, J., Trinidad, P., Rana, O. F. and Ruiz-Cortés, A. (2016), Automated configuration support for infrastructure migration to the cloud, in: Future Generation Computer Systems, vol. 55, 2016, pp. 200–212

Garg, S. K., Versteeg, S. and Buyya, R. (2011), Smicloud: A framework for comparing and ranking cloud services, in: Proceedings of the 4^{th} International IEEE Conference on Utility and Cloud Computing (UCC 2011), pp. 210–218

Garousi, V., Felderer, M. and Mäntylä, M. V. (2019), Guidelines for including grey literature and conducting multivocal literature reviews in software engineering, in: Information and Software Technology, vol. 106, 2019, pp. 101–121

Gartner (2012), Gartner Says Nearly One Third of Organizations Use or Plan to Use Cloud Offerings to Augment Business Intelligence Capabilities, Technical Report, Stammford CT 2012

Gerrard, C., Noble, J., Robinson, P. and Wheater, S. (2011), Analysis of power-saving techniques over a large multi-use cluster, in: Proceedings of the 9^{th} International IEEE Conference on Dependable, Autonomic and Secure Computing (DASC 2011), pp. 364–371

Gheyas, I. A. and Abdallah, A. E. (2016), Detection and prediction of insider threats to cyber security: a systematic literature review and meta-analysis, in: Big Data Analytics, vol. 1, 2016, iss. 1, pp. 1–6

Gholami, M. F., Daneshgar, F., Beydoun, G. and Rabhi, F. (2017), Challenges in migrating legacy software systems to the cloud—an empirical study, in: Information Systems, vol. 67, 2017, pp. 100–113

Gholami, M. F., Daneshgar, F., Low, G. and Beydoun, G. (2016), Cloud migration process—A survey, evaluation framework, and open challenges, in: Journal of Systems and Software, vol. 120, 2016, pp. 31–69

Giboney, J. S., Briggs, R. and Nunamaker, J. J. (2019), Engineering Artifacts and Processes of Information Systems, in: Journal of Management Information Systems, vol. 36, 2019, iss. 1, pp. 11–13

Gilbert, S. and Lynch, N. (2002), Brewer's conjecture and the feasibility of consistent, available, partition-tolerant web services, in: ACM SIGACT News, vol. 33, 2002, iss. 2, pp. 51–59

GloNet Consortium (2012), GloNet Platform Design: D3.1, Technical Report Deliverable, Lisboa 2012

GloNet Consortium (2015a), GloNet Deliverable D6.2 – Implementation of the pilot demonstrator, Technical Report Deliverable, Lisboa 2015

GloNet Consortium (2015b), GloNet Deliverable D6.3 – Pilot assessment results and lessons learned, Technical Report Deliverable, Lisboa 2015

GloNet Consortium (2016), GloNet Project Glocal Enterprise Network Focusing on Customer-Centric Collaboration, Technical Report Deliverable Final Report, Lisboa 2016

Godfrey, M. W. and German, D. M. (2008), The past, present, and future of software evolution, in: Proceedings of the Frontiers of Software Maintenance (FoSM 2008), pp. 129–138

Goeken, M. (2007), Entwicklung von data-warehouse-systemen: Anforderungsmanagement, modellierung, implementierung, New York 2007

Gogouvitis, S. V., Kousiouris, G., Vafiadis, G., Kolodner, E. K. and Kyriazis, D. (2012), OPTIMIS and VISION cloud: how to manage data in clouds, in: European Conference on Parallel Processing (Euro-Par 2011), pp. 35–44

Goldberg, R. P. (1974), Survey of virtual machine research, in: Computer, vol. 7, 1974, iss. 6, pp. 34–45

Golfarelli, M., Rizzi, S. and Cella, I. (2004), Beyond data warehousing: what's next in business intelligence?, in: Proceedings of the 7th international ACM workshop on Data warehousing and OLAP, pp. 1–6

Gonidis, F. (2015), A Framework Enabling the Cross-Platform Development of Service-based Cloud Applications, PhD Dissertation at the University of Sheffield, Sheffield, 2015

Gonidis, F., Simons, A. J., Paraskakis, I. and Kourtesis, D. (2013), Cloud application portability: an initial view, in: Proceedings of the 6th Balkan Conference in Informatics, pp. 275–282

Gonzalez-Perez, C. and Henderson-Sellers, B. (2008), Metamodelling for software engineering, 1st Edition, New York 2008

González-Torres, A., García-Peñalvo, F. J., Therón-Sánchez, R. and Colomo-Palacios, R. (2016), Knowledge discovery in software teams by means of evolutionary visual software analytics, in: Science of Computer Programming, vol. 121, 2016, pp. 55–74

Gouvas, P., Kalaboukas, K., Ledakis, G., Dimitrakos, T., Daniel, J., Ducatel, G. and Dominguez, N. R. (2015), A Cloud Orchestrator for Deploying Public Services on the Cloud–The Case of STRATEGIC Project, in: IFIP International Conference on Trust Management, pp. 217–225

Graça, P. and Camarinha-Matos, L. M. (2019), A Model of Evolution of a Collaborative Business Ecosystem Influenced by Performance Indicators, in: Working Conference on Virtual Enterprises, pp. 245–258

Grozev, N. and Buyya, R. (2014), Inter-Cloud architectures and application brokering: taxonomy and survey, in: Software: Practice and Experience, vol. 44, 2014, iss. 3, pp. 369–390

Gruber, T. R. (1993), A translation approach to portable ontology specifications, in: Knowledge acquisition, vol. 5, 1993, iss. 2, pp. 199–220

Guo, C. J., Sun, W., Huang, Y., Wang, Z. H. and Gao, B. (2007), A framework for native multi-tenancy application development and management, in: the 9th International IEEE Conference on E-Commerce Technology and the 4th International IEEE Conference on Enterprise Computing, E-Commerce, and E-Services (CEC/EEE 2007), pp. 551–558

Gupta, H., Vahid Dastjerdi, A., Ghosh, S. K. and Buyya, R. (2017), iFogSim: A toolkit for modeling and simulation of resource management techniques in the Internet of Things, Edge and Fog computing environments, in: Software: Practice and Experience, vol. 47, 2017, iss. 9, pp. 1275–1296

Gustavo, A., Fabio, C., Harumi, K. and Vijay, M. (2004), Web services: concepts, architectures and applications, Berlin 2004

Haas, H. and Brown, A. (2004), Web services glossary, Technical Report World Wide Web Consortium (W3C), Cambridge MA 2004

Hadas, D., Loy, I., Nagin, K., Rochwerger, B., Glikson, A. and Schour, L. (2014), Migration of virtual resources over remotely connected networks, Technical Report US Patent Application, Washington 2014

Haendler, T. and Neumann, G. (2019), A Framework for the Assessment and Training of Software Refactoring Competences, in: Proceedings of 11th International Conference on Knowledge Management and Information Systems (KMIS 2019), pp. 307–316

Hajjat, M., Sun, X., Sung, Y.-W. E., Maltz, D., Rao, S., Sripanidkulchai, K. and Tawarmalani, M. (2010), Cloudward bound: Planning for beneficial migration of enterprise applications to the cloud, in: ACM SIGCOMM Computer Communication Review, vol. 40, 2010, iss. 4, pp. 243–254

Hale, D. P., Haworth, D. A. and Sharpe, S. (1990), Empirical software maintenance studies during the 1980s, in: Proceedings of the Conference on Software Maintenance (CSM 1990), pp. 118–123

Hamdaqa, M., Sabri, M. M., Singh, A. and Tahvildari, L. (2015), Hadoop: MapReduce for ad-hoc cloud computing, in: Proceedings of the 25th Annual International Conference on Computer Science and Software Engineering (CSSE 2015), pp. 26–34

Hanna, S. (2009), Cloud Computing: Finding the silver lining, Technical Report White Paper Information Services Group Inc. (ISG), Stamford CT 2016

Harsh, P., Dudouet, F., Cascella, R. G., Jegou, Y. and Morin, C. (2012), Using open standards for interoperability issues, solutions, and challenges facing cloud computing, in: Proceedings of the Workshop on Systems Virtualiztion Management (SVM) of the 8th International Conference on Network and Service Management (CNSM 2012), pp. 435–440

Hassan, M. M. and Huh, E.-N. (2010), A Novel Market-Oriented Dynamic Collaborative Cloud Service Platform, in: Handbook of Cloud Computing, vol. 17, 2010, pp. 407–434

Hassan, M. M., Song, B., Yoon, C., Lee, H. W. and Huh, E.-N. (2009), A Novel Market-Oriented Dynamic Collaborative Cloud Service Infrastructure, in: World Conference on Services-II (SERVICES 2009), pp. 9–16

Hauck, M., Huber, M., Klems, M., Kounev, S., Müller-Quade, J., Pretschner, A., Reussner, R. and Tai, S. (2010), Challenges and opportunities of cloud computing, Technical Report Karlsruhe Reports in Informatics 19, Karlsruhe 2010, pp. 1–32

Haug, K. C., Kretschmer, T. and Strobel, T. (2016), Cloud adaptiveness within industry sectors: Measurement and observations, in: Telecommunications Policy, vol. 40, 2016, iss. 4, pp. 291–306

Haziza, M., Voidrot, J.-F., Queille, J.-P., Pofelski, L. and Blazy, S. (1992), Software maintenance: an analysis of industrial needs and constraints, in: Proceedings of the Conference on Software Maintenance (CSM 1992), pp. 18–26

Heinrich, L. J. and Lehner, F. (2005), Informationsmanagement: Planung, Überwachung und Steuerung der Informationsinfrastruktur, 8th Edition, Munich 2005

Herbst, N., Bauer, A., Kounev, S., Oikonomou, G., Eyk, E. V., Kousiouris, G., Evangelinou, A., Krebs, R., Brecht, T. and Abad, C. L. (2018), Quantifying Cloud Performance and Dependability: Taxonomy, Metric Design, and Emerging Challenges, in: ACM Transactions on Modeling and Performance Evaluation of Computing Systems (ToMPECS), vol. 3, 2018, iss. 4, pp. 1–19

Herbst, N. R. (2011), Quantifying the Impact of Platform Configuration Space for Elasticity Benchmarking, Master thesis at Karlsruhe Institut für Technologie (KIT), Karlsruhe, 2011, pp. 1–93

Herbst, N. R., Kounev, S. and Reussner, R. (2013), Elasticity in Cloud Computing: What It Is, and What It Is Not, in: Proceedings of the 10th International Conference on Autonomic Computing (ICAC 2013), pp. 23–27

Herbst, N. R., Kounev, S., Weber, A. and Groenda, H. (2015), BUNGEE: An Elasticity Benchmark for Self-Adaptive IaaS Cloud Environments, in: Proceedings of the 10th International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS 2015), pp. 46–56

Herlihy, M. P. and Wing, J. M. (1990), Linearizability: A correctness condition for concurrent objects, in: ACM Transactions on Programming Languages and Systems (TOPLAS), vol. 12, 1990, iss. 3, pp. 463–492

Hermann, M., Pentek, T. and Otto, B. (2016), Design principles for industrie 4.0 scenarios, in: Proceedings of the 49th Hawaii International Conference on System Sciences (HICSS 2016), pp. 3928–3937

Herzwurm, G. and Pietsch, W. (2009), Management von IT-Produkten: Geschäftsmodelle, Leitlinien und Werkzeugkasten für softwareintensive Systeme und Dienstleistungen, Heidelberg 2009

Hevner, A. and Chatterjee, S. (2010), Design science research in information systems, in: Design research in information systems, pp. 8–22

Hevner, A. R., March, S. T., Park, J. and Ram, S. (2004), Design science in information systems research, in: MIS quarterly, vol. 28, 2004, iss. 1, pp. 75–105

Hilley, D. (2009), Cloud computing: A taxonomy of platform and infrastructure-level offerings, Technical Report Georgia Institute of Technology, Atlanta 2009

Hirzalla, M. (2010), Realizing business agility requirements through SOA and cloud computing, in: Proceedings of the 18th International IEEE Requirements Engineering Conference (RE 2010), pp. 379–380

Horn, P. (2001), Autonomic computing: IBM's Perspective on the State of Information Technology, Technical Report IBM, Ottawa 2001

Hoyer, V., Stanoesvka-Slabeva, K., Janner, T. and Schroth, C. (2008), Enterprise mashups: Design principles towards the long tail of user needs, in: Proceedings of the International IEEE Conference on Services Computing (SCC 2008), pp. 601–602

Hu, J. and Klein, A. (2009), A benchmark of transparent data encryption for migration of web applications in the Cloud, in: Proceedings of the 8th International IEEE Conference on Dependable, Autonomic and Secure Computing (DASC 2009), pp. 735–740

Hu, Y., Deng, B., Peng, F., Hong, B., Zhang, Y. and Wang, D. (2016), A survey on evaluating elasticity of cloud computing platform, in: Proceedings of the World Automation Congress (WAC 2016), pp. 1–4

Huber, N., Brosig, F., Spinner, S., Kounev, S. and Bähr, M. (2016), Model-based self-aware performance and resource management using the descartes modeling language, in: IEEE Transactions on Software Engineering (TSE), vol. 43, 2016, iss. 5, pp. 432–452

Huebscher, M. C. and McCann, J. A. (2008), A survey of autonomic computing: degrees, models, and applications, in: ACM Computing Surveys (CSUR), vol. 40, 2008, iss. 3, pp. 1–7

Hwang, K., Bai, X., Shi, Y., Li, M., Chen, W.-G. and Wu, Y. (2015), Cloud performance modeling with benchmark evaluation of elastic scaling strategies, in: IEEE Transactions on parallel and distributed systems, vol. 27, 2015, iss. 1, pp. 130–143

Ibarra, J. A., Orue-Echevarria, L., Seoane, Z. C., Gorroñogoitia, J. and Karaboga, B. (2014), ARTIST Technical Feasibility Tool: Supporting the Early Technical Feasibility Assessment of Application Cloudifications, in: Proceedings of the

9th International Conference on Software Engineering Advances (ICSEA 2014), pp. 390–395

IBM (2006), An architectural blueprint for autonomic computing, Technical Report White Paper IBM, New York 2006

IBM (2010), Information governance as a holistic approach to managing and leveraging information, Technical Report IBM, New York 2010

ICANN (2021), The Internet Corporation for Assigned Names and Numbers, on the website of ICANN, http://www.icann.org/, last updated 09.10.2020

IDC (2010), Cloud Computing: an IDC Update, Technical Report International Data Corporation, Newtonville 2010

IEEE (1998), IEEE Standard for Software Maintenance IEEE Std 1219-1998, Technical Report IEEE Standard, New York 1998

IEEE-SA (2017), Standard for Intercloud Interoperability and Federation (SIIF) P2302, Technical Report IEEE Standard, New York 2017

IEEE TCCLD (2020), IEEE Cloud Computing the IEEE Technical Committee on Cloud Computing (TCCLD), on the website of the Computer Society (CS), https://tc.computer.org/tccld/, last updated 2020

IETF (2015), System for Cross-domain Identity Management 2.0, on the website of Simple Cloud, http://www.simplecloud.info/, last updated 09.2015

INVENT Consortium (2015), INVENT, Reinventing the Distribution and Delivery of Personal Services through Cloud Apps and Marketplaces, research project funded by GSRT, BMBF and the European Community

Islam, S., Lee, K., Fekete, A. and Liu, A. (2012), How a consumer can measure elasticity for cloud platforms, in: Proceedings of the 3rd joint WOSP/SIPEW International Conference on Performance Engineering (ICPE), pp. 85–96

ISoc (2023), The Internet Society, on the website of the Internet Society, https://www.internetsociety.org/, last updated 2023

ISO/IEC (2006), ISO/IEC 14764:2006 Software Engineering —Software Life Cycle Processes— Maintenance, Technical Report Standard, Geneva 2006, pp. 1–44

ISO/IEC (2008), ISO/IEC 12207 - Standard for Information Technology - Software life cycle processes, Technical Report Standard, Geneva 2008

Jacobs, A. (2009), The pathologies of big data, in: Communications of the ACM, vol. 52, 2009, iss. 8, pp. 36–44

Jamshidi, P., Ahmad, A. and Pahl, C. (2013), Cloud migration research: a systematic review, in: IEEE Transactions on Cloud Computing, vol. 1, 2013, iss. 2, pp. 142–157

Janssen, M. and Joha, A. (2011), Challenges for adopting cloud-based software as a service (SaaS) in the public sector, in: Proceedings of the European Conference on Information Systems (ECIS 2011), pp. 80–93

JClouds Apache (2023), JClouds, on the website of JClouds, http://jclouds.apache.org/, last updated 26.03.2023

Jennings, B. and Stadler, R. (2015), Resource management in clouds: Survey and research challenges, in: Journal of Network and Systems Management, vol. 23, 2015, iss. 3, pp. 567–619

Jesson, J., Matheson, L. and Lacey, F. M. (2011), Doing your literature review: Traditional and systematic techniques, 1st Edition, Los Angeles, London et al. 2011

Jeusfeld, M. A., Jarke, M. and Mylopoulos, J. (2009), Metamodeling for method engineering, 1st Edition, Boston 2009

Jogalekar, P. and Woodside, M. (2000), Evaluating the scalability of distributed systems, in: IEEE Transactions on Parallel and Distributed Systems, vol. 11, 2000, iss. 6, pp. 589–603

Jonas, E., Pu, Q., Venkataraman, S., Stoica, I. and Recht, B. (2017), Occupy the cloud: distributed computing for the 99%, in: Proceedings of the 2017 Symposium on Cloud Computing, pp. 445–451

Jonas, E., Schleier-Smith, J., Sreekanti, V., Tsai, C.-C., Khandelwal, A., Pu, Q., Shankar, V., Carreira, J., Krauth, K., Yadwadkar, N. et al. (2019), Cloud programming simplified: A berkeley view on serverless computing, Technical Report at the Department of Electrical Engineering and Computer Sciences University of California, Berkeley 2019

Jones, D. and Gregor, S. (2007), The anatomy of a design theory, in: Journal of the Association for Information Systems, vol. 8, 2007, iss. 5, pp. 312–336

Jothimani, A. P. (2022), Enabling Secure Cloud Governance using Policy as Code, Master Thesis at the Department of Computer Science and Engineering of the Chalmers University of Technology within the University of Gothenburg, Gothenburg 2022

Juan-Verdejo, A. (2012), Assisted migration of enterprise applications to the Cloud– A hybrid Cloud approach, in: Tagungsband des 4. Workshops "Business Intelligence" der GI-Fachgruppe Business Intelligence in Zusammenarbeit mit der Fachgruppe Wirtschaftsinformatik der Fachhochschule Mainz, pp. 14–27

Juan-Verdejo, A. (2014), Model-Driven Engineering meets the Platform-as-a-Service Model, in: ACM Student Research Competition (SRC) of the MODELS 2014 conference, ACM/IEEE 17th International Conference on Model Driven Engineering Languages and Systems (MODELS 2014), pp. 1–6

Juan-Verdejo, A. and Baars, H. (2013), Decision support for partially moving applications to the cloud: the example of business intelligence, in: Proceedings of the 5th conference on Hot topics in cloud computing (HotTopiCS 2013), pp. 35–42

Juan-Verdejo, A. and Surajbali, B. (2016), XaaS Multi-Cloud Marketplace Architecture Enacting the Industry 4.0 Concepts, in: Doctoral Conference on Computing, Electrical and Industrial Systems: Technological Innovation for Cyber-Physical Systems (DoCEIS 2016), pp. 11–23

Juan-Verdejo, A., Surajbali, B., Baars, H. and Kemper, H.-G. (2014a), Moving business intelligence to cloud environments, in: Proceedings of the Cross Cloud Workshop of the IEEE Conference on Computer Communications (INFOCOM 2014), pp. 43–48

Juan-Verdejo, A., Zschaler, S., Surajbali, B., Baars, H. and Kemper, H.-G. (2014b), InCLOUDer: A Formalised Decision Support Modelling Approach to Migrate

Applications to Cloud Environments, in: In Proceedings of the 40th Euromicro Conference on Software Engineering and Advanced Applications (SEAA 2014), pp. 467–474

Juels, A. and Kaliski Jr., B. S. (2007), PORs: Proofs of retrievability for large files, in: Proceedings of the 14th ACM conference on Computer and Communications Security (CCS 2007), pp. 584–597

Kagermann, H., Helbig, J., Hellinger, A. and Wahlster, W. (2013), Recommendations for Implementing the Strategic Initiative INDUSTRIE 4.0: Securing the Future of German Manufacturing Industry; Final Report of the Industrie 4.0 Working Group, Technical Report Forschungsunion National Academy of Science and Engineering, Munich 2013

Kaisler, S. and Money, W. H. (2011), Service migration in a Cloud architecture, in: 44th Hawaii International Conference on System Sciences (HICSS 2011), pp. 1–10

Kandias, M., Virvilis, N. and Gritzalis, D. (2011), The insider threat in cloud computing, in: International Workshop on Critical Information Infrastructures Security, pp. 93–103

Karac, I. and Turhan, B. (2018), What Do We (Really) Know about Test-Driven Development?, in: IEEE Software, vol. 35, 2018, iss. 4, pp. 81–85

Karataş, A. S., Oğuztüzün, H. and Doğru, A. (2010), Mapping extended feature models to constraint logic programming over finite domains, in: International Conference on Software Product Lines, pp. 286–299

Karunakaran, S., Krishnaswamy, V. and Rangaraja, S. P. (2015), Business view of cloud: Decisions, models and opportunities–a classification and review of research, in: Management Research Review, vol. 38, 2015, iss. 6, pp. 582–604

Katz, R. N. (Ed., 2008), The Tower and the Cloud: Higher Education in the Age of Cloud Computing, no place given 2008

Kazman, R., Woods, S. G. and Carriere, J. S. (1998), Requirements for integrating software architecture and reengineering models: CORUM II, in: Proceedings of the 5th Working Conference on Reverse Engineering (Cat. No.98TB100261), pp. 154–163

Kecskemeti, G., Maurer, M., Brandic, I., Kertesz, A., Nemeth, Z. and Dustdar, S. (2012), Facilitating self-adaptable Inter-Cloud management, in: Proceedings of the 20th Euromicro International IEEE Conference on Parallel, Distributed and Network-Based Processing (PDP 2012), pp. 575–582

Kemerer, C. F. and Slaughter, S. (1999), An Empirical Approach to Studying Software Evolution, in: IEEE Transitions of Software Engineering, vol. 25, 1999, iss. 4, pp. 493–509

Kephart, J. O. and Chess, D. M. (2003), The vision of autonomic computing, in: IEEE Computer, vol. 36, 2003, iss. 1, pp. 41–50

Kern, T., Lacity, M. C. and Willcocks, L. (2002), Netsourcing: renting business applications and services over a network, New York, London et al. 2002

Khadka, R., Saeidi, A., Idu, A., Hage, J. and Jansen, S. (2013), Legacy to SOA evolution: a systematic literature review, in: Migrating Legacy Applications: Challenges in Service Oriented Architecture and Cloud Computing Environments, vol. 3, 2013, pp. 40–70

Khajeh-Hosseini, A., Greenwood, D. and Sommerville, I. (2010a), Cloud Migration: A Case Study of Migrating an Enterprise IT System to IaaS, in: Proceedings of the 3rd International IEEE Conference on Cloud Computing (CLOUD 2010), pp. 450–457

Khajeh-Hosseini, A., Greenwood, D. and Sommerville, I. (2010b), Cloud migration: A case study of migrating an enterprise it system to IaaS, in: Proceedings of the 3rd International IEEE Conference on Cloud Computing (CLOUD 2010), pp. 450–457

Khajeh-Hosseini, A., Sommerville, I., Bogaerts, J. and Teregowda, P. (2011), Decision support tools for Cloud migration in the enterprise, in: Proceedings of the International IEEE Conference on Cloud Computing (CLOUD 2011), pp. 541–548

Khajeh-Hosseini, A., Sommerville, I. and Sriram, I. (2010c), Research challenges for enterprise Cloud Computing, in: Proceedings of the 1st ACM Symposium on Cloud Computing (SOCC 2010), pp. 1–11

Khan, S. M. and Hamlen, K. W. (2012), AnonymousCloud: A data ownership privacy provider framework in cloud computing, in: 11th International IEEE Conference on the Trust, Security and Privacy in Computing and Communications (TrustCom 2012), pp. 170–176

Khayer, A., Talukder, M. S., Bao, Y. and Hossain, M. N. (2020), Cloud computing adoption and its impact on SMEs' performance for cloud supported operations: A dual-stage analytical approach, in: Technology in Society, vol. 60, 2020, pp. 1–28

Khusidman, V. and Ulrich, W. (2007), Architecture-driven modernization: Transforming the enterprise, in: Seminar Software Analysis and Transformation, pp. 1–7

Kim, J.-S., Nguyen, C. and Hwang, S. (2016), Moha: Many-task computing meets the big data platform, in: Proceedings of the 12th International IEEE Conference on e-Science (e-Science 2016), pp. 193–202

Kistowski, J. V., Herbst, N., Kounev, S., Groenda, H., Stier, C. and Lehrig, S. (2017), Modeling and extracting load intensity profiles, in: ACM Transactions on Autonomous and Adaptive Systems (TAAS), vol. 11, 2017, iss. 4, pp. 1–23

KIT, FZI and Paderborn (2021), Palladio Simulator Toolkit, Simulator Documentation, on the website of Palladio, http://www.palladio-simulator.com/, last updated 12.2021

Klarl, H. (2011), Zugriffskontrolle in Geschäftsprozessen, New York 2011

Klems, M., Nimis, J. and Tai, S. (2008), Do Clouds compute? A framework for estimating the value of Cloud Computing, in: Workshop on E-Business: Designing E-Business Systems, Markets, Services, and Networks, pp. 110–123

Klems, M., Tai, S., Shwartz, L. and Grabarnik, G. (2010), Automating the delivery of IT Service Continuity Management through cloud service orchestration, in: IEEE Network Operations and Management Symposium (NOMS 2010), pp. 65–72

Klusch, M., Kapahnke, P., Schulte, S., Lecue, F. and Bernstein, A. (2016), Semantic web service search: a brief survey, in: KI-Künstliche Intelligenz, vol. 30, 2016, iss. 2, pp. 139–147

Kniberg, H. (2011), Lean from the trenches: Managing large-scale projects with Kanban, Boston, San Francisco et al. 2011

KnowHow Consortium (2014), KnowHow a novel Knowledge Management Platform as a Service (KM-PaaS) environment for social business applications, FP7 ICT research project result CAS Software, Karlsruhe 2014

KnowHow Consortium (2015), KnowHow Final Report – Financial analysis, Technical Report Deliverable EU FP7 project, Thessaloniki 2015

Koehler, P., Anandasivam, A., Dan, M. and Weinhardt, C. (2010a), Cloud Services from a Consumer Perspective, in: Americas Conference on Information Systems (AMCIS 2010), pp. 1–10

Koehler, P., Anandasivam, A., Dan, M. and Weinhardt, C. (2010b), Customer Heterogeneity and Tariff Biases in Cloud Computing. In: 31 st International Conference on Information Systems (ICIS 2010), pp. 1–19

Koltun, G., Basirati, M. R., Hammeed, M. S., Böhm, M., Krcmar, H. and Vogel-Heuser, B. (2019), Reverse Engineering on changed Functional Specification Documents for Model-Based Requirements Engineering, in: Proceedings of the International IEEE Conference on Industrial Cyber Physical Systems (ICPS 2019), pp. 687–692

Kooper, M. N., Maes, R. and Lindgreen, E. R. (2011), On the governance of information: Introducing a new concept of governance to support the management of information, in: International Journal of Information Management, vol. 31, 2011, iss. 3, pp. 195–200

Korsgaard, H., Klokmose, C. N. and Bødker, S. (2016), Computational alternatives in participatory design: putting the T back in socio-technical research, in: Proceedings of the 14th Participatory Design Conference: Full papers, vol. 1, 2016, pp. 71–79

Kounev, S., Rizou, S., Zschaler, S., Alexakis, S., Bures, T., Jézéquel, J.-M., Kourtesis, D. and Pantelopoulos, S. (2013), RELATE: a research training network on engineering and provisioning of service-based cloud applications, in: Proceedings of the International Workshop on Hot topics in cloud services (HotTopiCS 2013), pp. 51–54

Kozaczynski, W. and Ning, J. Q. (1994), Automated program understanding by concept recognition, in: Automated software engineering, vol. 1, 1994, iss. 1, pp. 61–78

Kozaczynski, W., Ning, J. Q. and Engberts, A. (1992), Program concept recognition and transformation, in: IEEE Transactions on Software Engineering, vol. 18, 1992, iss. 12, pp. 1065–1075

Kozuch, M. A., Ryan, M. P., Gass, R., Schlosser, S. W., O'Hallaron, D., Cipar, J., Krevat, E., López, J., Stroucken, M. and Ganger, G. R. (2009), Tashi: location-aware cluster management, in: Proceedings of the 1st workshop on Automated control for datacenters and clouds, pp. 43–48

Krasteva, I., Stavru, S. and Ilieva, S. (2013), Agile Model-Driven Modernization to the Service Cloud, in: The 8th International Conference on Internet and Web Applications and Services (ICIW 2013), pp. 1–9

Krcmar, H. (2015), Informationsmanagement, New York

Krebs, R., Loesch, M. and Kounev, S. (2014a), Platform-as-a-Service Architecture for Performance Isolated Multi-Tenant Applications, in: Proceedings of the 7th International IEEE Conference on Cloud Computing (CLOUD 2014), pp. 914–921

Krebs, R., Momm, C. and Kounev, S. (2012), Architectural Concerns in Multi-tenant SaaS Applications, in: Proceedings of the International Conference on Cloud Computing and Services Science (CLOSER 2012), vol. 12, 2012, pp. 426–431

Krebs, R., Momm, C. and Kounev, S. (2014b), Metrics and techniques for quantifying performance isolation in cloud environments, in: Science of Computer Programming, vol. 90, 2014, pp. 116–134

Krupitzer, C., Roth, F. M., Van Syckel, S., Schiele, G. and Becker, C. (2015), A survey on engineering approaches for self-adaptive systems, in: Pervasive and Mobile Computing, vol. 17, 2015, pp. 184–206

Krutz, R. L., Vines, R. D. and Brunette, G. (2010), Cloud security: A comprehensive guide to secure cloud computing, Indianapolis 2010

Kubicek, H. (1977), Heuristische Bezugsrahmen und heuristisch angelegte Forschungsdesign als Elemente einer Konstruktionsstrategie empirischer Forschung, in: Köhler, Richard (Ed., 1977), pp. 3–36

Kühne, T. (2006), Matters of (meta-) modeling, in: Software & Systems Modeling, vol. 5, 2006, iss. 4, pp. 369–385

Kuhrmann, M., Diebold, P., Münch, J., Tell, P., Garousi, V., Felderer, M., Trektere, K., McCaffery, F., Linssen, O. and Hanser, E. (2017), Hybrid software and system development in practice: waterfall, scrum, and beyond, in: Proceedings of the International Conference on Software and System Process (ICSSP 2017), pp. 30–39

Kuperberg, M., Herbst, N., Kistowski, J. von and Reussner, R. (2011), Defining and quantifying elasticity of resources in cloud computing and scalable platforms, Technical Report KIT Fakultät für Informatik, Karlsruhe 2011

Lamport, L. (1979), How to make a multiprocessor computer that correctly executes multiprocess programs, in: IEEE Transactions on Computers, vol. 100, 1979, iss. 9, pp. 690–691

Lara, J. de, Guerra, E. and Cuadrado, J. S. (2015), Model-driven engineering with domain-specific meta-modelling languages, in: Software & Systems Modeling, vol. 14, 2015, iss. 1, pp. 429–459

Lasi, H., Fettke, P., Kemper, H.-G., Feld, T. and Hoffmann, M. (2014), Industry 4.0, in: Business & Information Systems Engineering, vol. 6, 2014, iss. 4, pp. 239–242

Laudon, K. C., Laudon, J. P. and Schoder, D. (2009), Wirtschaftsinformatik: Eine Einführung, 3rd Edition, München, Boston et al. 2009

Lazouski, A., Mancini, G., Martinelli, F. and Mori, P. (2012a), Usage control in cloud systems, in: International Conference for Internet Technology and Secured Transactions (ICITST 2012), pp. 202–207

Lazouski, A., Martinelli, F. and Mori, P. (2012b), A prototype for enforcing usage control policies based on XACML, in: International Conference on Trust, Privacy and Security in Digital Business (Trustbus 2012), pp. 79–92

Lee, E., Lee, B., Shin, W. and Wu, C. (2003), A reengineering process for migrating from an object-oriented legacy system to a component-based system, in: Proceedings of the 27th Annual International IEEE Computer Software and Applications Conference (COMPSAC 2003), pp. 336–341

Lehman, M. M. (1969), The programming process, Technical Report IBM, New York 1969

Lehner, F. (1991), Software life cycle management based on a phase distinction method, in: Microprocessing and Microprogramming, vol. 32, 1991, iss. 1, pp. 603–608

Lehrig, S. (2014), Applying Architectural Templates for Design-Time Scalability and Elasticity Analyses of SaaS Applications, in: Proceedings of the 2nd International Workshop on Hot Topics in Cloud service Scalability (HotTopiCS 2014), pp. 1–2

Lehrig, S., Eikerling, H. and Becker, S. (2015), Scalability, Elasticity, and Efficiency in Cloud Computing: A Systematic Literature Review of Definitions and Metrics, in: Proceedings of the 11th International ACM SIGSOFT Conference on Quality of Software Architectures (QoSA 2015), pp. 83–92

Lei, H., Ganjeizadeh, F., Jayachandran, P. K. and Ozcan, P. (2017), A statistical analysis of the effects of Scrum and Kanban on software development projects, in: Robotics and Computer-Integrated Manufacturing, vol. 43, 2017, pp. 59–67

Leitner, P., Wittern, E., Spillner, J. and Hummer, W. (2019), A mixed-method empirical study of Function-as-a-Service software development in industrial practice, in: Journal of Systems and Software, vol. 149, 2019, pp. 340–359

Lemke, C. and Brenner, W. (2015), Geschäftsmodelle und Märkte im digitalen Zeitalter, in: Einführung in die Wirtschaftsinformatik, vol. 1, 2015, pp. 193–256

Lenk, A., Klems, M., Nimis, J., Tai, S. and Sandholm, T. (2009), What's inside the Cloud? An architectural map of the Cloud landscape, in: Proceedings of the Workshop on Software Engineering Challenges of Cloud Computing of the International Conference on Software Engineering (ICSE 2009), pp. 23–31

Leymann, F. and Fritsch, D. (2009), Cloud computing: The next revolution in IT, in: Proceedings of the 52^{th} Photogrammetric Week, pp. 3–12

Li, J., Wéldrich, O. and Ziegler, W. (2008), Towards SLA-based software licenses and license management in grid computing, in: Priol, Thierry and Vanneschi, Marco (Ed., 2008), pp. 139–152

Li, J., Squicciarini, A. C., Lin, D., Sundareswaran, S. and Jia, C. (2017), title, in: IEEE Transactions on Dependable and Secure Computing, vol. 14, 2017, iss. 2, pp. 185–198

Li, Y., Kamousi, P., Han, F., Yang, S., Yan, X. and Suri, S. (2013), Memory efficient minimum substring partitioning, in: Proceedings of the Very Large Data Bases (VLDB) Endowment, vol. 6, 2013, iss. 3, pp. 169–180

Lientz, B. P., Swanson, B. E. and Tompkins, G. E. (1978), Characteristics of application software maintenance, in: Communications of the ACM, vol. 21, 1978, iss. 6, pp. 466–471

Linnenluecke, M. K. (2017), Resilience in business and management research: A review of influential publications and a research agenda, in: International Journal of Management Reviews, vol. 19, 2017, iss. 1, pp. 4–30

Lioudakis, G. V., Koukovini, M. N., Papagiannakopoulou, E. I., Dellas, N., Kalaboukas, K., Carvalho, R. M. de, Hassani, M., Bracciale, L., Bianchi, G., Juan-Verdejo, A., Alexakis, S., Gaudino, F., Cascone, D. and Barracano, P. (2019), Facilitating GDPR compliance: the H2020 BPR4GDPR approach, in: Conference on e-Business, e-Services and e-Society (I3E 2019), pp. 72–78

Liu, F., Tong, J., Mao, J., Bohn, R., Messina, J., Badger, L. and Leaf, D. (2011), NIST cloud computing reference architecture, in: NIST special publication, vol. 500, 2011, pp. 1–292

Liu, Y. and Xu, X. (2017), Industry 4.0 and cloud manufacturing: A comparative analysis, in: Journal of Manufacturing Science and Engineering, vol. 139, 2017, iss. 3, pp. 1–8

Lowe, S. (2011), Mastering VMware vSphere 5, New York, Berlin et al. 2011

Lübke, D. and Lessen, T. van (2016), Modeling test cases in BPMN for behavior-driven development, in: IEEE software, vol. 33, 2016, iss. 5, pp. 15–21

Lucassen, G., Dalpiaz, F., Werf, J. M. E. M. van der and Brinkkemper, S. (2016), Improving agile requirements: the quality user story framework and tool, in: Requirements Engineering, vol. 21, 2016, iss. 3, pp. 383–403

Machi, L. A. and McEvoy, B. T. (2016), The literature review: Six steps to success, 3rd Edition, Thousand Oaks, London et al. 2016

Macias, M. and Guitart, J. (2012), Client classification policies for SLA enforcement in shared cloud datacenters, in: Proceedings of the 12th International IEEE/ACM Symposium on Cluster, Cloud and Grid Computing (CCGRID 2012), pp. 156–163

Madiot, F. (2010), MoDisco, a model-driven platform to support real legacy modernization use cases, in: Information Systems Transformation: Architecture-Driven Modernization Case Studies, pp. 1–365

Mahy, Y., Ouzzif, M. and Bouragba, K. (2016), Toward a shared view of IT governance, in: International Journal of Innovation, Management and Technology, vol. 7, 2016, iss. 4, pp. 1–125

Mall, R. (2018), Fundamentals of software engineering, 5th Edition, Delhi 2018

Mani, N., Helfert, M. and Pahl, C. (2017), A Domain-specific Rule Generation Using Model-Driven Architecture in Controlled Variability Model, in: Procedia computer science, vol. 112, 2017, pp. 2354–2362

Manyika, J., Chui, M., Brown, B., Bughin, J., Dobbs, R., Roxburgh, C. and Byers, A. H. (2011), Big data: The next frontier for innovation, competition, and productivity, Technical Report McKinsey Global Institute, Washington 2011

Mao, Y., You, C., Zhang, J., Huang, K. and Letaief, K. B. (2017), A survey on mobile edge computing: The communication perspective, in: IEEE Communications Surveys & Tutorials, vol. 19, 2017, iss. 4, pp. 2322–2358

Marinescu, D. C. (2017), Cloud computing: theory and practice, Cambridge 2017

Marjanovic, O. (2007), The next stage of operational business intelligence: Creating new challenges for business process management, in: 40th Annual Hawaii International Conference on System Sciences (HICSS 2007), pp. 215–224

Marston, S., Li, Z., Bandyopadhyay, S., Zhang, J. and Ghalsasi, A. (2011), Cloud Computing — The business perspective, in: Decision Support Systems, vol. 51, 2011, iss. 1, pp. 176–189

Marston, S., Li, Z., Bandyopadhyay, S., Zhang, J. and Ghalsasi, A. (2016), Cloud Computing Survey, in: IDG Executive Summary, vol. 51, 2016, iss. 1, pp. 176–189

Matetic, S., Ahmed, M., Kostiainen, K., Dhar, A., Sommer, D., Gervais, A., Juels, A. and Capkun, S. (2017), ROTE: Rollback Protection for Trusted Execution, in: 26th USENIX Security Symposium (USENIX Security 2017), pp. 1289–1306

Mayer, R. J., Crump, J. W., Fernandes, R., Keen, A. and Painter, M. K. (1995), Information integration for concurrent engineering (IICE) compendium of methods report, Technical Report DTIC Document, Fort Belvoir 1995

McAffer, J., Lemieux, J.-M. and Aniszczyk, C. (2010), Eclipse Rich Client platform, Boston, San Francisco et al. 2010

McCarthy, J. C. and Bidgoli, H. (2006), Digital Libraries: Security and Preservation Considerations, in: Handbook of Information Security, Key Concepts, Infrastructure, Standards, and Protocols, vol. 1, 2006, pp. 49–76

McGough, A. S., Gerrard, C., Haldane, P., Sharples, D., Swan, D., Robinson, P., Hamlander, S. and Wheater, S. (2010), Intelligent power management over large clusters, in: Proceedings of the International IEEE/ACM Conference on Green Computing and Communications (GREENCOM 2010) and International Conference on Cyber, Physical and Social Computing (CPSCom 2010), pp. 88–95

McKinnon, T. (2008), The Force.com Multitenant Architecture: Understanding the Design of Salesforce.com's Internet Application Development Platform, Technical Report Salesforce, San Francisco 2008

Mellegård, N., Ferwerda, A., Lind, K., Heldal, R. and Chaudron, M. R. V. (2015), Impact of introducing domain-specific modelling in software maintenance: An industrial case study, in: IEEE Transactions on Software Engineering, vol. 42, 2015, iss. 3, pp. 245–260

Melnikov, A. and Zeilenga, K. (2006), Simple authentication and security layer (SASL) RFC 4422, Technical Report OpenLDAP Foundation, Vienna 2006

Mengerink, J., Schiffelers, R. R. H., Serebrenik, A. and Brand, M. van den (2016), DSL/Model Co-Evolution in Industrial EMF-Based MDSE Ecosystems, in: Proceedings of the 19th International ACM/IEEE Conference on Model Driven Engineering Languages and Systems (MODELS 2016), pp. 2–7

Mens, T., Wermelinger, M., Ducasse, S., Demeyer, S., Hirschfeld, R. and Jazayeri, M. (2005), Challenges in software evolution, in: Proceedings of the 8th International Workshop on Principles of Software Evolution, pp. 13–22

Menychtas, A., Konstanteli, K., Alonso, J., Orue-Echevarria, L., Gorronogoitia, J., Kousiouris, G., Santzaridou, C., Bruneliere, H., Pellens, B. and Stuer, P. (2014), Software modernization and cloudification using the ARTIST migration methodology and framework, in: Scalable Computing: Practice and Experience, vol. 15, 2014, iss. 2, pp. 131–152

Menychtas, A., Santzaridou, C., Kousiouris, G., Varvarigou, T., Orue-Echevarria, L., Alonso, J. M., Gorronogoitia, J., Bruneliere, H., Strauss, O. and Senkova, T. (2013), ARTIST Methodology and Framework: A novel approach for the migration of legacy software on the Cloud, in: Proceedings of the 15th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC 2013), pp. 424–431

Mertens, P., Back, A., König, W., Krallmann, H., Rieger, B., Scheer, A.-W., Seibt, D., Stahlknecht, P., Strunz, H., Thome, R. and Wedekind, H. (2001), Lexikon der Wirtschaftsinformatik, 3rd ed., Berlin and Heidelberg 2001

Messerschmitt, D. G. and Szyperski, C. (2000), Industrial and economic properties of software: technology, processes, and value, Technical Report Microsoft, Seatle 2000

Metsch, T., Edmonds, A. and Parák, B. (2010), Open cloud computing interface-infrastructure no. GFD-R in The Open Grid Forum Document Series, Technical Report Standards Track Open Cloud Computing Interface (OCCI) Working Group, Muncie IN 2010

Mietzner, R., Unger, T., Titze, R. and Leymann, F. (2009), Combining different multi-tenancy patterns in service-oriented applications, in: Proceedings of the International IEEE Enterprise Distributed Object Computing Conference (EDOC 2009), pp. 131–140

Miyachi, C. (2018), What is "Cloud"? It is time to update the NIST definition?, in: IEEE Cloud computing, vol. 5, 2018, iss. 3, pp. 6–11

MODAClouds Consortium (2012), MODAClouds MOdel-Driven Approach for design and execution of applications on multiple Clouds, FP7 ICT research project Politecnico di Milano, Milano 2013

Mogul, J. C., Isaacs, R. and Welch, B. (2017), Thinking about availability in large service infrastructures, in: Proceedings of the 16th Workshop on Hot Topics in Operating Systems (HotOS 2017), pp. 12–17

Mohagheghi, P., Berre, A. J., Sadovykh, A., Barbier, F. and Benguria, G. (2010), Reuse and migration of legacy systems to interoperable cloud services-the REMICS project, in: Proceedings of the International Conference on Modeling, Design, and Analysis for the Service Cloud (MDA4ServiceCloud 2010), pp. 1–8

Mohagheghi, P. and Sæther, T. (2011), Software Engineering Challenges for Migration to the Service Cloud Paradigm: Ongoing work in the REMICS Project, in: Proceedings of the IEEE World Congress on Services (SERVICES 2011), pp. 507–514

Mohammadi, S. V., Bauer, M. and Juan-Verdejo, A. (2012), Dynamic Cloud Reconfiguration to Meet QoS Requirements, in: Proceedings of the 6th EuroSys Doctoral Workshop (EuroDW 2012), pp. 1–2

Mohammadi, S. V., Kounev, S., Juan-Verdejo, A. and Surajbali, B. (2013), Soft Reservations: Uncertainty-Aware Resource Reservations in IaaS Environments, in: Proceedings of the 3rd International Symposium on Business Modeling and Software Design (BMSD 2013), pp. 223–229

Molano, J. I. R., Lovelle, J. M. C., Montenegro, C. E., Granados, J. J. R. and Crespo, R. G. (2018), Metamodel for integration of internet of things, social networks, the cloud and industry 4.0, in: Journal of ambient intelligence and humanized computing, vol. 9, 2018, pp. 709–723

MongoDB, Inc. (2022), Mongo DB, on the website of mongodb, http://www.mongodb.org/, last updated 2022

Mooney, J. D. (1997), Bringing portability to the software process, in: Department of Statistics and Computer Science, pp. 1–9

Morgan, L. and Conboy, K. (2018), Cloud Computing: An Exploration of Factors Impacting Adoption, in: Software Technology: 10 Years of Innovation in IEEE Computer, vol. 1, 2018, pp. 295–327

Morin, C. and Cascella, R. (2014), Contrail final publishable summary report, Technical Report White Paper Contrail consortium, Rennes 2014

mOSAIC Consortium (2013), mOSAIC Open-Source API and Platform for Multiple Clouds, EU FP7 project result

Moscato, F., Aversa, R., Di Martin, B., Rak, M., Venticinque, S. and Petcu, D. (2017), An Ontology for the Cloud in mOSAIC, in: Cloud Computing, vol. 1, 2017, Boca Raton, London and New York, pp. 467–485

Motahari-Nezhad, H. R., Stephenson, B. and Singhal, S. (2009), Outsourcing business to cloud computing services: Opportunities and challenges, in: IEEE Internet Computing, vol. 10, 2009, iss. 4, pp. 1–17

Mueller, H. A., Jahnke, J. H., Smith, D. B., Storey, M.-A., Tilley, S. R. and Wong, K. (2000), Reverse engineering: A roadmap, in: Proceedings of the Conference on the Future of Software Engineering (FOSE 2000), pp. 47–60

Müller, J. M., Buliga, O. and Voigt, K.-I. (2018), Fortune favors the prepared: How SMEs approach business model innovations in Industry 4.0, in: Technological Forecasting and Social Change, vol. 132, 2018, pp. 2–17

Na, S.-H. and Huh, E.-N. (2015), A broker-based cooperative security-SLA evaluation methodology for personal cloud computing, in: Security and Communication networks, vol. 8, 2015, iss. 7, pp. 1318–1331

Nabi, M., Toeroe, M. and Khendek, F. (2016), Availability in the cloud: State of the art, in: Journal of Network and Computer Applications, vol. 60, 2016, pp. 54–67

Negash, S. (2004), Business intelligence, in: Communications of the Association for Information Systems, vol. 13, 2004, iss. 1, pp. 177–195

Nelson, V. P. (1990), Fault-tolerant computing: Fundamental concepts, in: Computer, vol. 23, 1990, iss. 7, pp. 19–25

Nepal, S. and Zic, J. (2008), A conflict neighbouring negotiation algorithm for resource services in dynamic collaborations, in: Proceedings of the International IEEE Conference on Services Computing (SCC 2008), vol. 2, 2008, pp. 283–290

Nepal, S., Zic, J. and Chan, J. (2007), A distributed approach for negotiating resource contributions in dynamic collaboration, in: Proceedings of the 8th International IEEE Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT 2007), pp. 82–86

Noël, N. E. D., Conn, B., Read, J. I., Carrera, R., Dolphin, A. and Rix, H. W. (2015), The MAGellanic Inter-Cloud (MAGIC) project–II. Slicing up the Bridge, in: Monthly Notices of the Royal Astronomical Society, vol. 452, 2015, iss. 4, pp. 4222–4235

Noor, T. H., Sheng, Q. Z., Maamar, Z. and Zeadally, S. (2016), Managing trust in the cloud: State of the art and research challenges, in: Computer, vol. 49, 2016, iss. 2, pp. 34–45

Noran, O. (2004), Towards a meta-methodology for collaborative networked organisations, in: Working Conference on Virtual Enterprises, pp. 71–78

Noran, O. and Bernus, P. (2017), Business Cloudification-An Enterprise Architecture Perspective, in: Proceedings of the 19th International Conference on Enterprise Information Systems (ICEIS 2017), pp. 353–360

Noroozi, E. and Seifzadeh, H. (2018), Proposing novel measures to alleviate the risks of migration to open source software, in: Proceedings of the 10th International Conference on Computer Modeling and Simulation, pp. 134–139

Novais, L., Maqueira, J. M. and Ortiz-Bas, Á. (2019), A systematic literature review of cloud computing use in supply chain integration, in: Computers & Industrial Engineering, vol. 129, 2019, pp. 296–314

Nunamaker, J. F. J. and Briggs, R. O. (2011), Toward a broader vision for Information Systems, in: ACM Transactions on Management Information Systems (TMIS), vol. 2, 2011, iss. 4, pp. 1–20

OASIS (2014), Cloud Application Management for Platforms Version 1.1 - Draft 03, Technical Report Specification OASIS, Burlington 2014

Oberle, K. and Fisher, M. (2010), ETSI CLOUD–initial standardization requirements for cloud services, in: Economics of Grids, Clouds, Systems, and Services, vol. 1, 2010, pp. 105–115

OCDA (2012), Master Usage Model: Compute Infratructure as a Service, Technical Report Open Data Center Alliance (OCDA), Miami 2012

Odun-Ayo, I., Agono, F. and Misra, S. (2018), Cloud Migration: Issues and Developments, in: Proceedings of the International MultiConference of Engineers and Computer Scientists (IMECS 2018), vol. 1, 2018, pp. 1–6

Oliveira Albuquerque, R. de, Villalba, L. J. G., Orozco, A. L. S., Sousa Júnior, R. T. de and Kim, T.-H. (2016), Leveraging information security and computational trust for cybersecurity, in: Journal of Supercomputing, vol. 72, 2016, iss. 10, pp. 3729–3763

O'Loughlin, M. (2019), ITIL® 4 and the Cloud, Technical Paper White Paper The Stationary Office, London 2019

OMG (2015), Object Management Group (OMG) Unified Modeling Language (UML) Superstructure Version 2.3, Technical Report Standard, Milford MA 2015

OMG (2016), Architecture-Driven Modernization: Knowledge Discovery Meta-Model (KDM), v 1.4, Technical Report Standard, Milford MA 2015

Ongowarsito, H., Kosala, R. and Prabowo, H. (2017), IT governance contingency factors in cloud computing services, in: Proceedings of the International IEEE Conference on Innovative and Creative Information Technology (ICITech 2017), pp. 1–5

OPTIMIS Consortium (2013), OPTIMIS Optimized Infrastructure Services, Technical Report ATOS, Barcelona 2013, EU FP7 project

OPTIMIS-Consortium (2022), OPTIMIS Toolkit Final Version (v3.0) integrated in Open Nebula, Toolkit, Open Nebula, https://opennebula.io/, last updated 2022

Ortiz, J., Lee, B., Balazinska, M., Gehrke, J. and Hellerstein, J. L. (2018), SLAOrchestrator: reducing the cost of performance SLAs for cloud data analytics, in: Proceedings of the USENIX Annual Technical Conference (USENIX ATC 2018), pp. 547–560

Osanaiye, O., Choo, K.-K. R. and Dlodlo, M. (2016), Distributed denial of service (DDoS) resilience in cloud: Review and conceptual cloud DDoS mitigation framework, in: Journal of Network and Computer Applications, vol. 67, 2016, pp. 147–165

Österle, H., Becker, J., Frank, U., Hess, T., Karagiannis, D., Krcmar, H., Loos, P., Mertens, P., Oberweis, A. and Sinz, E. J. (2011), Memorandum on design-oriented information systems research, in: European Journal of Information Systems, vol. 20, 2011, iss. 1, pp. 7–10

Ostrom, A. L., Parasuraman, A., Bowen, D. E., Patrício, L., Voss, C. A. and Lemon, K. (2015), Service research priorities in a rapidly changing context, in: Journal of Service Research, vol. 18, 2015, iss. 2, pp. 127–159

PaaSport Consortium (2014a), PaaSport Deliverable D1.1 – Requirements Analysis, Technical Report Deliverable EU FP7 project, Karlsruhe 2016

PaaSport Consortium (2014b), PaaSport Deliverable D1.2 – PaaSport Reference Architecture, Technical Report Deliverable EU FP7 project, Karlsruhe 2016

PaaSport Consortium (2015), PaaSport Deliverable D5.2 – PaaSport Marketplace Infrastructure — First Release, Technical Report Deliverable EU FP7 project, Karlsruhe 2016

PaaSport Consortium (2016a), PaaSport A semantically-enhanced marketplace of interoperable Platform-as-a-Service offerings for the deployment and migration of business applications of Small and Medium Enterprises, FP7 SME research project Bundesverband IT-Mittelstand EV, Aachen 2016

PaaSport Consortium (2016b), PaaSport Deliverable D6.2 – PaaSport Marketplace Demonstration Infrastructure, Technical Report Deliverable EU FP7 project, Karlsruhe 2016

Paelke, V. (2014), Augmented reality in the smart factory: Supporting workers in an industry 4.0. environment, in: Proceedings of the IEEE Emerging Technology and Factory Automation Conference (ETFA 2014), pp. 1–4

Palos-Sanchez, P. R., Robina-Ramirez, R. and Velicia-Martin, F. (2019), What role does corporate governance play in the intention to use cloud computing technology?, in: Symmetry, vol. 11, 2019, iss. 10, pp. 1–19

Pantelopoulos, S., Bratanis, K., Paraskakis, I., Dautov, R., Alexakis, S. and Juan-Verdejo, A. (2015), Reinventing the Distribution and Delivery of Personal Services through Cloud Apps and Marketplaces, in: Greek-German Days on Research, Innovation & Young Scientists, p. 1

Papazoglou, M. P. and Van Den Heuvel, W.-J. (2007), Service-oriented architectures: approaches, technologies and research issues, in: International Journal of Very Large Data Bases (VLDB), vol. 16, 2007, iss. 3, pp. 389–415

Paquette, S., Jaeger, P. T. and Wilson, S. C. (2010), Identifying the security risks associated with governmental use of cloud computing, in: Government information quarterly, vol. 27, 2010, iss. 3, pp. 245–253

Parkhill, D. F. (1966), The challenge of the computer utility, 2^{nd} Edition, New York, Berlin et al. 1966

Parno, B. (2008), Bootstrapping Trust in a "Trusted" Platform, in: Proceedings of the 3rd conference on Hot topics in security (HOTSEC 2008), pp. 1–6

Patterson, D., Brown, A., Broadwell, P., Candea, G., Chen, M., Cutler, J., Enriquez, P., Fox, A., Kiciman, E. and Merzbacher, M. (2002), Recovery-oriented computing (ROC): Motivation, definition, techniques, and case studies UCB/CSD-02-1175, Technical Report UC Berkeley Computer Science, Berkeley 2002

Pawluk, P., Simmons, B., Smit, M., Litoiu, M. and Mankovski, S. (2012), Introducing STRATOS: A cloud broker service, in: Proceedings of the 5th International IEEE Conference on Cloud Computing (CLOUD 2012), pp. 891–898

Peffers, K., Tuunanen, T., Rothenberger, M. A. and Chatterjee, S. (2007), A design science research methodology for information systems research, in: Journal of management information systems, vol. 24, 2007, iss. 3, pp. 45–77

Peltier, T. R. (2013), Information security fundamentals, 2nd Edition, Boca Raton, London and New York 2013

Pérez, M. S., Herrero, P., Gannon, D. and Katz, D. S. (2010), Grid computing, high performance and distributed application, in: Concurrency Computation: Practical Experience, vol. 22, 2010, pp. 1335–1337

Pérez-Castillo, R., De Guzman, I. G.-R. and Piattini, M. (2011), Knowledge Discovery Metamodel-ISO/IEC 19506: A standard to modernize legacy systems, in: Computer Standards & Interfaces, vol. 33, 2011, iss. 6, pp. 519–532

Petcu, D., Crăciun, C., Neagul, M., Panica, S., Di Martino, B., Venticinque, S., Rak, M. and Aversa, R. (2011), Architecturing a sky computing platform, in: Proceedings of the ServiceWave 2010 Workshops of the 3rd European Conference on a Service-Based Internet (ServiceWave 2010), pp. 1–13

Petcu, D., Macariu, G., Panica, S. and Crăciun, C. (2013a), Portable cloud applications: from theory to practice, in: Future Generation Computer Systems, vol. 29, 2013, iss. 6, pp. 1417–1430

Petcu, D., Panica, S., Crăciun, C., Neagul, M. and Şandru, C. (2013b), Cloud resource orchestration within an open-source component-based platform as a service, in: Concurrency and Computation: Practice and Experience, pp. 2443–2469

Petruch, K., Stantchev, V. and Tamm, G. (2011), A survey on IT-governance aspects of cloud computing, in: International Journal of Web and Grid Services, vol. 7, 2011, iss. 3, pp. 268–303

Pfeiffer, S., Held, M. and Lee, H. (2018), Digitalisierung „machen“–Ansichten im Engineering zur partizipativen Gestaltung von Industrie 4.0, in: Arbeit 4.0–Digitalisierung, IT und Arbeit, vol. 1, 2018, pp. 113–129

Pham, M. L. (2016), Roboconf: an Autonomic Platform Supporting Multi-level Fine-grained Elasticity of Complex Applications on the Cloud, PhD Dissertation at the Université Grenoble Alpes, Grenoble 2016

Phaphoom, N., Wang, X., Samuel, S., Helmer, S. and Abrahamsson, P. (2015), A survey study on major technical barriers affecting the decision to adopt cloud services, in: Journal of Systems and Software, vol. 103, 2015, pp. 167–181

Pigoski, T. M. and Nelson, L. E. (1994), Software maintenance metrics: a case study, in: Proceedings of the International IEEE Conference on Software Maintenance, 1994, pp. 392–401

Piiroinen, M. (2020), Containerization and Cloud Migration of Legacy Web Services, in

Pires Barbosa, F. and Schwertner Charão, A. (2012), Impact of pay-as-you-go Cloud platforms on software pricing and development: a review and case study, in: International Conference on Computational Science and Its Applications (ICCSA 2012), pp. 404–417

Platt, R. and Thompson, N. (2019), The Past, Present, and Future of UML, in: Advanced Methodologies and Technologies in Network Architecture, Mobile Computing, and Data Analytics, vol. 1, 2019, pp. 1452–1460

Pohl, K., Böckle, G. and Der Linden, F. J. van (2005), Software product line engineering: foundations, principles and techniques, Berlin, Heidelberg and New York, 2005

Polyviou, A., Pouloudi, N. and Rizou, S. (2014), Which Factors Affect Software-as-a-Service Selection the Most? A Study from the Customer's and the Vendor's Perspective, in: Proceedings of the 7th Hawaii International IEEE Conference on System Sciences (HICSS 2014), pp. 5059–5068

Poole, J. D. (2001), Model-driven architecture: Vision, standards and emerging technologies, in: Workshop on Metamodeling and Adaptive Object Models at the European Conference on Object-Oriented Programming (ECOOP 2001), vol. 50, 2001, pp. 1–15

Potts, C. (1993), Software-engineering research revisited, in: IEEE software, vol. 10, 1993, iss. 5, pp. 19–28

Prieto-González, L., Tamm, G. and Stantchev, V. (2015), Governance of cloud computing: semantic aspects and cloud service brokers, in: Proceedings of the International Journal of Web and Grid Services (IJWGS 2015), vol. 11, 2015, iss. 4, pp. 377–389

Priol, T. and Vanneschi, M. (Ed., 2008), From Grids to Service and Pervasive Computing, Boston MA 2008

Pritchett, D. (May 2008), BASE: An Acid Alternative, in: Queue journal, vol. 6, 2008, iss. 3, pp. 48–55

Qu, C., Calheiros, R. N. and Buyya, R. (2018), Auto-scaling web applications in clouds: A taxonomy and survey, in: ACM Computing Surveys (CSUR), vol. 51, 2018, iss. 4, pp. 1–73

Quinton, C., Haderer, N., Rouvoy, R. and Duchien, L. (2013), Towards multi-cloud configurations using feature models and ontologies, in: Proceedings of the 2013 International ACM workshop on Multi-cloud applications and federated clouds, pp. 21–26

Quinton, C., Romero, D. and Duchien, L. (2016), SALOON: a platform for selecting and configuring cloud environments, in: Software: Practice and Experience, vol. 46, 2016, iss. 1, pp. 55–78

Qureshi, K. A., Mohammed, W. M., Ferrer, B. R., Lastra, J. L. M. and Agostinho, C. (2017), Legacy systems interactions with the supply chain through the C2NET cloud-based platform, in: 15th International IEEE Conference on Industrial Informatics (INDIN 2017), pp. 725–731

Rabai, L. B. A., Jouini, M., Aissa, A. B. and Mili, A. (2013), A cybersecurity model in cloud computing environments, in: Journal of King Saud University-Computer and Information Sciences, vol. 25, 2013, iss. 1, pp. 63–75

Rahimi, F., Møller, C. and Hvam, L. (2016), Business process management and IT management: The missing integration, in: International Journal of Information Management, vol. 36, 2016, iss. 1, pp. 142–154

Rai, R., Sahoo, G. and Mehfuz, S. (2015), Exploring the factors influencing the cloud computing adoption: a systematic study on cloud migration, in: Springerplus, vol. 4, 2015, iss. 197, pp. 1–12

Raicu, I., Zhao, Y., Dumitrescu, C., Foster, I. and Wilde, M. (2007), Falkon: a Fast and Light-weight tasK executiON framework, in: Proceedings of the ACM/IEEE conference on Supercomputing (SC 2007), pp. 1–43

Ranjan, R., Benatallah, B., Dustdar, S. and Papazoglou, M. P. (2015), Cloud resource orchestration programming: overview, issues, and directions, in: IEEE Internet Computing, vol. 19, 2015, iss. 5, pp. 46–56

Rashid, M. A. and Lo, B. W. N. (2017), A task-oriented software maintenance model, in: Malaysian Journal of Computer Science, vol. 9, 2017, iss. 2, pp. 36–42

Rashid, N. and Khan, S. U. (2013), Software Re-engineering Risks Management Model, in: e-Proceeding of Software Engineering Postgraduates Workshop (SEPoW 2013), pp. 76–82

RELATE ITN (2015), RELATE Trans-European Research Training Network on Engineering and Provisioning of Service-Based Cloud Applications, FP7 ITN research project Karlsruhe Institut für Technologie (KIT), Karlsruhe 2015

REMICS Consortium (2013), REMICS Reuse and Migration of legacy applications to Interoperable Cloud Services, FP7 ICT research project SINTEF, Oslo 2013

Ren, L., Zhang, L., Wang, L., Tao, F. and Chai, X. (2017), Cloud manufacturing: key characteristics and applications, in: International Journal of Computer Integrated Manufacturing, vol. 30, 2017, iss. 6, pp. 501–515

Reussner, R., Becker, S., Happe, J., Heinrich, R., Koziolek, A., Koziolek, H., Kramer, M. and Krogmann, K. (2016), Modeling and simulating software architectures: The Palladio approach, Cambridge and London 2016

Rimal, B. P. and Maier, M. (2016), Workflow scheduling in multi-tenant cloud computing environments, in: IEEE Transactions on parallel and distributed systems, vol. 28, 2016, iss. 1, pp. 290–304

Rings, T., Caryer, G., Gallop, J., Grabowski, J., Kovacikova, T., Schulz, S. and Stokes-Rees, I. (2009), Grid and cloud computing: opportunities for integration with the next generation network, in: Journal of Grid Computing, vol. 7, 2009, iss. 3, pp. 375–393

Rittinger, J., Jaecksch, B., Merkel, D., Kazmaier, G. S., Volker, L., Uhrig, S. and Choi, Y. (2018), Enterprise performance management planning operations at an enterprise database, Technical Report US Patent Application, Washington 2018

Rittinghouse, J. W. and Ransome, J. F. (2009), Cloud computing: implementation, management, and security, 1st Edition, Boca Raton, London and New York 2009

Robertazzi, T. (2012), Grid and Cloud Computing, 1st Edition, New York, pp. 65–68

Robinson, J. and Park, J. S. (2010), Towards Trusted Cloud Computing, in: Proceedings of the iConference, iSchool (iConference 2010), pp. 188–193

Rochwerger, B., Breitgand, D., Levy, E., Galis, A., Nagin, K., Llorente, I. M., Montero, R., Wolfsthal, Y., Elmroth, E., Caceres, J., Ben-Yehuda, M., Emmerich, W. and Galan, F. (2009), The Reservoir model and architecture for open federated cloud computing, in: IBM Journal of Research and Development, vol. 53, 2009, iss. 4, pp. 1–11

Rochwerger, B., Breitgand, D., Levy, E., Maraschini, A., Massonet, P., Muñoz, H., Toffetti, G., Epstein, A., Hadas, D. and Loy, I. (2011), Reservoir-when one cloud is not enough, in: Computer iss. 3, pp. 44–51

Rodero-Merino, L., Vaquero, L. M., Gil, V., Galán, F., Fontán, J., Montero, R. S. and Llorente, I. M. (2010), From infrastructure delivery to service management in clouds, in: Future Generation Computer Systems, vol. 26, 2010, iss. 8, pp. 1226–1240

Rosado, D., Gomez, R., Mellado, D. and Fernandez-Medina, E. (2012), Security analysis in the migration to cloud environments, in: Future Internet, vol. 4, 2012, iss. 2, pp. 469–487

Rosenberg, L. H. and Hyatt, L. E. (1996), Software re-engineering, in: Software Assurance Technology Center, vol. 1, 1996, pp. 2–3

Rossi, M. (2015), Metamodeling, Ontology, and Methodology Engineering, in: Management Information Systems, Wiley Encyclopedia of Management, vol. 7, 2015, pp. 1–4

Saaty, T. L. (1977), A scaling method for priorities in hierarchical structures, in: Journal of mathematical psychology, vol. 15, 1977, iss. 3, pp. 234–281

Saaty, T. L. (2005), Theory and applications of the analytic network process: decision making with benefits, opportunities, costs, and risks, 2nd Edition, Pittsburgh 2005

Saaty, T. L. and Gonzalez Vargas, L. L. (2001), Models, methods, concepts, and applications of the AHP, Boston, Dordrecht and London 2001

Sadeeq, M. M., Abdulkareem, N. M., Zeebaree, S. R., Ahmed, D. M., Sami, A. S. and Zebari, R. R. (2021), IoT and Cloud computing issues, challenges and opportunities: A review, in: Qubahan Academic Journal, vol. 1, 2021, iss. 2, pp. 1–7

Sáez, S. G., Andrikopoulos, V., Bitsaki, M., Leymann, F. and Van Hoorn, A. (2018), Utility-based decision making for migrating cloud-based applications, in: ACM Transactions on Internet Technology (TOIT), vol. 18, 2018, iss. 2, pp. 1–22

Salama, M. and Bahsoon, R. (2015), Quality-driven architectural patterns for self-aware cloud-based software, in: Proceedings of the 8th International IEEE Conference on Cloud Computing (CLOUD 2015), pp. 844–851

Salehie, M. and Tahvildari, L. (2009), Self-adaptive software: Landscape and research challenges, in: ACM Transactions on Autonomous and Adaptive Systems (TAAS), vol. 4, 2009, iss. 2, pp. 1–14

Sandru, C., Petcu, D. and Munteanu, V. I. (2012), Building an open-source platform-as-a-service with intelligent management of multiple cloud resources, in: Proceedings of the 2012 IEEE/ACM 5th International Conference on Utility and Cloud Computing, pp. 333–338

Santos, N., Gummadi, K. P. and Rodrigues, R. (2009), Towards trusted cloud computing, in: Proceedings of the 1st USENIX conference on Hot topics in cloud computing (HotCloud 2009), pp. 3–3

Satish, C. J. and Mahendran, A. (2017), The effect of 3D visualization on mainframe application maintenance: A controlled experiment, in: Journal of King Saud University-Computer and Information Sciences, vol. 31, 2017, iss. 3, pp. 1–12

SCALR (2022), SCALR: Open source Cloud Management Platform, on the website of SCALR, http://www.scalr.com/, last updated 2022

Schmuck, F. B. and Haskin, R. L. (2002), GPFS: A Shared-Disk File System for Large Computing Clusters, in: Proceedings of the Conference on File and Storage Technologies (FAST 2002), vol. 2, 2002, iss. 19, pp. 1–19

Schneider, S. and Sunyaev, A. (2016), Determinant factors of cloud-sourcing decisions: reflecting on the IT outsourcing literature in the era of cloud computing, in: Journal of Information Technology, vol. 31, 2016, iss. 1, pp. 1–31

Schoder, D., Bichler, M., Buhl, U., Hess, T., Krcmar, H. and Sinz, E. (2011), Profil der Wirtschaftsinformatik, in: Einstimmiger Beschluss der gemeinsamen Sitzung der Wissenschaftlichen Kommision Wirtschaftsinformatik (WKWI) im Verband der Hochschullehrer für Betriebswirtschafts eV und des Fachbereichs Wirtschaftsinformatik (FB WI) in der Gesellschaft für Informatik e. V.(GI), vol. 18, 2011, pp. 1–6

Sciancalepore, S., Piro, G., Caldarola, D., Boggia, G. and Bianchi, G. (2018), On the Design of a Decentralized and Multiauthority Access Control Scheme in

Federated and Cloud-Assisted Cyber-Physical Systems, in: IEEE Internet of Things Journal, vol. 5, 2018, iss. 6, pp. 5190–5204

Seaman, C. B. (1999), Qualitative methods in empirical studies of software engineering, in: IEEE Transactions on Software Engineering (TSE), vol. 25, 1999, iss. 4, pp. 557–572

Shaikh, R. and Sasikumar, M. (2015), Data Classification for achieving Security in cloud computing, in: Procedia computer science, vol. 45, 2015, pp. 493–498

Sharma, M., Gupta, R. and Acharya, P. (2020), Analysing the adoption of cloud computing service: a systematic literature review, in: Global Knowledge, Memory and Communication, vol. 1, 2020, pp. 1–38

Shoaib, M., Ishaq, A., Ahmad, M. A., Talib, S., Mustafa, G. and Ahmed, A. (2017), Software Migration Frameworks for Software System Solutions: A Systematic Literature Review, in: International Journal of Advanced Computer Science and Applications, vol. 8, 2017, iss. 11, pp. 192–204

Shollo, A. and Kautz, K. (2010), Towards an understanding of business intelligence, in: Australasian Conference on Information Systems (ACIS 2010), pp. 1–10

Shukla, A. and Simmhan, Y. (2018), Toward reliable and rapid elasticity for streaming dataflows on clouds, in: 2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS), IEEE, pp. 1096–1106

Shumaker, J., Ward, K., Petter, S. and Riley, J. (2017), Mitigating the threat of lost knowledge within information technology departments, in: Proceedings of the 50th Hawaii International Conference On System Sciences (HICSS 2017), pp. 5440–5449

Siegel, J. and Perdue, J. (2012), Cloud services measures for global use: the service measurement index (SMI), in: 2012 Annual SRII global conference, IEEE, pp. 411–415

Silva, F. S. D., Lemos, M. O., Medeiros, A., Neto, A. V., Pasquini, R., Moura, D., Rothenberg, C., Mamatas, L., Correa, S. L. and Cardoso, K. V. (2018), Necos project: Towards lightweight slicing of cloud federated infrastructures, in: Proceedings of the 4th IEEE Conference on Network Softwarization and Workshops (NetSoft 2018), pp. 406–414

Simon, H. A. (1955), A behavioral model of rational choice, in: Quarterly journal of economics, vol. 69, 1955, iss. 1, pp. 99–118

Singh, A. and Chatterjee, K. (2017), Cloud security issues and challenges: A survey, in: Journal of Network and Computer Applications, vol. 79, 2017, pp. 88–115

Singh, J., Singh, K. and Singh, J. (2019), Reengineering framework for open source software using decision tree approach, in: International Journal of Electrical and Computer Engineering, vol. 9, 2019, iss. 3, pp. 2041–2048

Singh, S., Jeong, Y.-S. and Park, J. H. (2016), A survey on cloud computing security: Issues, threats, and solutions, in: Journal of Network and Computer Applications, vol. 75, 2016, pp. 200–222

Sinha, U. and Shekhar, M. (2015), Comparison of various cloud simulation tools available in cloud computing, in: International Journal of Advanced Research in Computer and Communication Engineering, vol. 4, 2015, iss. 3, pp. 171–176

Sirianni, C. A., Singer, P. and Sabbagh, P. (2019), Innovation and Transformation of Service Business Models through Cloud Technology to Achieve Co-Creation Value within the Service Ecosystem, in: Journal of Service Science and Management, vol. 12, 2019, iss. 1, pp. 1–91

Sjoberg, D. I., Dyba, T. and Jorgensen, M. (2007), The future of empirical methods in software engineering research, in: Proceedings of the Conference on the Future of Software Engineering (FOSE 2007), pp. 358–378

Smart, J. F. (2014), BDD in Action: Behavior-Driven Development for the Whole Software Lifecycle, 1st Edition, Boston, New York et al. 2014

Sneed, H. M. (1995), Planning the reengineering of legacy systems, in: IEEE software, vol. 12, 1995, iss. 1, pp. 24–34

Sneed, H. M. (1996), Encapsulating legacy software for use in client/server systems, in: Proceedings of the 3rd IEEE Working Conference on Reverse Engineering, 1996, pp. 104–119

Sneed, H. M. (2015), Migrating from legacy to SoA (Invited Talk), in: Proceedings of the 9th International IEEE Symposium on the Maintenance and Evolution of Service-Oriented and Cloud-Based Environments (MESOCA 2015), pp. 1–6

Sneed, H. M. (2016), Planung von Migrationsprojekten, Gesellschaft für Informatik eV, in: Informatik 2016, vol. 1, 2016, pp. 1941–1946

Sneed, H. M., Ackermann, E., Winter, A., Hasitschka, M. and Teichmann, M. T. (2009), Software-Migration: Uebertragung alter Software-Systeme in eine neue Umgebung, Heidelberg 2009

Sneed, H. M., Hasitschka, M. and Teichmann, M.-T. (2005), Software-Produktmanagement: Wartung und Weiterentwicklung bestehender Anwendungssysteme, Heidelberg 2005

Sobhani, R., Seifzadeh, H. and Gandomani, T. J. (2018), A Review of Migration Processes to Open Source Software, in: International Journal of Open Source Software and Processes (IJOSSP), vol. 9, 2018, iss. 1, pp. 20–31

Solomon, T. (2017), Ranking of Cloud Service Providers Using a Dynamic TOPSIS Model for Provisioning of Enterprise IT Infrastructure in the Cloud, PhD Dissertation at the George Washington University, Washington 2017

Someswar, M. G. and Hemalatha, S. (2012), Identification and Implementation of Suitable Solutions to Critical Security Issues in Cloud Computing, in: International Journal of Engineering Research and Development, vol. 4, 2012, iss. 7, pp. 1–10

Sommerville, I. (2010), Software Engineering, 9th Edition, Harlow, London et al. 2010

Sommerville, I. and Kotonya, G. (1998), Requirements engineering: processes and techniques, New York, London et al. 1998

Sousa, P., Tibolet, J. and Guerreiro, S. (2017), Enterprise Cartography: From Theory to Practice, Technical Report Link Consulting, Lisboa 2017

Stackato (2021), Stackato Documentation: Stackato 2.10 documentation, Technical Documentation, on the website of Stackato, https://docs.huihoo.com/stackato/2.10/, last updated 12.07.2013

Stahlknecht, P. and Hasenkamp, U. (2005), Einführung in die Wirtschaftsinformatik, New York 2005

Stamou, A., Dimitriou, N., Kontovasilis, K. and Papavassiliou, S. (2019), Autonomic handover management for heterogeneous networks in a future internet context: A survey, in: IEEE Communications Surveys & Tutorials, vol. 21, 2019, iss. 4, pp. 3274–3297

Stanoevska-Slabeva, K., Wozniak, T. and Ristol, S. (2010), Grid and Cloud Computing, New York 2010

Stantchev, V. and Schröpfer, C. (2009), Negotiating and enforcing QoS and SLAs in grid and cloud computing, in: Proceedings of the International Conference on Grid and Pervasive Computing (GPC 2009), vol. 5529, 2009, pp. 25–35

Štefanič, P, Cigale, M., Fernandez, F. Q., Rogers, D., Knight, L., Jones, A and Taylor, I. (2018), TOSCA-based SWITCH workbench for application composition and infrastructure planning of time-critical applications, in: The 3rd Edition in the Series of Workshop on Interoperable Infrastructures for Interdisciplinary Big Data Sciences (IT4RIs 2018), pp. 1–9

Steinberg, D., Budinsky, F., Paternostro, M. and Merks, E. (2008), EMF: Eclipse Modeling Framework, 2nd Edition, Boston, San Francisco et al. 2008

Stender, J., Berlin, M. and Reinefeld, A. (2012), XtreemFS: a file system for the cloud, in: Data intensive storage services for cloud environments, pp. 267–285

Stergiou, C., Psannis, K. E., Kim, B.-G. and Gupta, B. (2018), Secure integration of IoT and cloud computing, in: Future Generation Computer Systems, vol. 78, 2018, pp. 964–975

Stickel, E. (2001), Informationsmanagement, Berlin 2001

Stol, K.-J., Ralph, P. and Fitzgerald, B. (2016), Grounded theory in software engineering research: a critical review and guidelines, in: Proceedings of the

38th International IEEE/ACM Conference on Software Engineering (ICSE 2016), pp. 120–131

Stonebraker, M. (2010), SQL databases v. NoSQL databases, in: Communications of the ACM, vol. 53, 2010, iss. 4, pp. 10–11

Strauch, S., Andrikopoulos, V. and Sáez, S. G. (2012), Enabling Tenant-Aware Administration and Management for JBI environments, in: Proceedings of the 5th International IEEE Conference on Service-Oriented Computing and Applications (SOCA 2012), pp. 206–213

Stray, V., Sjøberg, D. I. and Dybå, T. (2016), The daily stand-up meeting: A grounded theory study, in: Journal of Systems and Software, vol. 114, 2016, pp. 101–124

Suárez-Figueroa, M. C., Gómez-Pérez, A. and Fernández-López, M. (2012), The NeOn methodology for ontology engineering, New York 2012

Subashini, S. and Kavitha, V. (2011), A survey on security issues in service delivery models of cloud computing, in: Journal of network and computer applications, vol. 34, 2011, iss. 1, pp. 1–11

Suicimezov, N. and Georgescu, M. R. (2014), IT governance in Cloud, in: Procedia Economics and Finance, vol. 15, 2014, pp. 830–835

Surajbali, B. and Juan-Verdejo, A. (2014), A Marketplace Broker for Platform-as-a-Service Portability, in: Proceedings of the 3rd ACM European Conference on Service-Oriented and Cloud Computing (ESOCC 2014), pp. 275–282

Surajbali, B., Juan-Verdejo, A., Bär, H., Alexakis, S., Hübsch, G. and Bauer, M. (2014a), A cloud-based approach for collaboration of serviced-enhanced products, in: Proceedings of the International IEEE Conference on Industrial Engineering and Engineering Management (IEEM 2014), pp. 1295–1299

Surajbali, B., Juan-Verdejo, A., Bär, H., Alexakis, S., Hübsch, G. and Bauer, M. (2014b), A Cloud-Based Collaborative Platform Supporting Serviced-Enhanced Products for Emerging Markets, in: Proceedings of the International IEEE Conference on Cloud Computing in Emerging Markets (CCEM 2014), pp. 1–8

Susanto, H., Leu, F.-Y. and Chen, C. K. (2019), High-Performance Grid Computing for Science: A Review, in: Chemical Technology and Informatics in Chemistry with Applications, vol. 1, 2019, pp. 317–338

Swanson, E. B. (1976), The dimensions of maintenance, in: Proceedings of the 2nd international conference on Software engineering, pp. 492–497

Szvetits, M. and Zdun, U. (2016), Systematic literature review of the objectives, techniques, kinds, and architectures of models at runtime, in: Software & Systems Modeling, vol. 15, 2016, iss. 1, pp. 31–69

Szyperski, C., Bosch, J. and Weck, W. (1999), Component-oriented programming, in: Object-oriented technology workshop reader, European Conference on Object-Oriented Programming (ECOOP 1999), pp. 184–192

Taleb, T., Samdanis, K., Mada, B., Flinck, H., Dutta, S. and Sabella, D. (2017), On multi-access edge computing: A survey of the emerging 5G network edge cloud architecture and orchestration, in: IEEE Communications Surveys & Tutorials, vol. 19, 2017, iss. 3, pp. 1657–1681

Tang, M., Dai, X., Liu, J. and Chen, J. (2017), Towards a trust evaluation middleware for cloud service selection, in: Future Generation Computer Systems, vol. 74, 2017, pp. 302–312

Tesgera, C., Klein, M. and Juan-Verdejo, A. (2014), A cloudlet-based approach to tackle network challenges in mobile cloud applications, in: Proceedings of the International IEEE Conference on Advances in ICT for Emerging Regions (ICTer 2014), pp. 253–253

Thanos, G. A., Courcoubetis, C. and Stamoulis, G. D. (2007), Adopting the grid for business purposes: the main objectives and the associated economic issues, in: Grid Economics and Business Models, pp. 1–15

Thompson, W. J. and Walt, J. S. Van der (2010), Business intelligence in the cloud: original research, in: South African Journal of Information Management, vol. 12, 2010, iss. 1, pp. 1–15

Tilley, S. R. and Smith, D. (1995), Perspectives on legacy system reengineering, PhD Dissertation at Carnegie Mellon University, Pittsburgh 1995

Toffetti, G., Gambi, A., Pezzé, M. and Pautasso, C. (2010), Engineering autonomic controllers for virtualized web applications, in: International Conference on Web Engineering, pp. 66–80

Toosi, A. N., Calheiros, R. N. and Buyya, R. (2014), Interconnected cloud computing environments: Challenges, taxonomy, and survey, in: ACM Computing Surveys (CSUR), vol. 47, 2014, iss. 1, pp. 1–7

Torre, D. (2016), Verifying the consistency of UML models, in: Proceedings of the International IEEE Symposium on Software Reliability Engineering Workshops (ISSREW 2016), pp. 53–54

Tran, V., Lee, K., Fekete, A., Liu, A. and Keung, J. (2011), Size estimation of cloud migration projects with cloud migration point (CMP), in: Proceedings of the International IEEE Symposium on Empirical Software Engineering and Measurement (ESEM 2011), pp. 265–274

Tran, V. X., Tsuji, H. and Masuda, R. (2009), A new QoS ontology and its QoS-based ranking algorithm for Web services, in: Simulation Modelling Practice and Theory, vol. 17, 2009, iss. 8, pp. 1378–1398

Truong-Huu, T. and Tham, C.-K. (2014), A novel model for competition and cooperation among cloud providers, in: IEEE Transactions on Cloud Computing, vol. 2, 2014, iss. 3, pp. 251–265

Tusa, F., Paone, M., Villari, M. and Puliafito, A. (2010), CLEVER: A cloud-enabled virtual environment, in: Proceedings of the IEEE Symposium on Computers and Communications (ISCC 2010), pp. 477–482

Tzeng, G.-H. and Huang, J.-J. (2011), Multiple attribute decision making: methods and applications, Boca Raton, London and New York 2011

Ulrich, W. (2004), A status on OMG architecture-driven modernization task force, in: Proceedings of the EDOC Workshop on Model-Driven Evolution of Legacy Systems (MELS 2004), pp. 1–4

US-NIST (2009), The NIST Definition of Cloud Computing Recommendations of the Computer Security Division, Information Technology Laboratory Special

Publication 800-145, at the National Institute of Standards and Technology, Technical Report NIST Special Publication, Gaithersburg 2009

Van Hoorn, A., Frey, S., Goerigk, W., Hasselbring, W., Knoche, H., Köster, S., Krause, H., Porembski, M., Stahl, T. and Steinkamp, M. (2011), DynaMod project: Dynamic analysis for model-driven software modernization, in: Proceedings of the 1st International Workshop on Model-driven Software Migration (MDSM 2011), pp. 12–13

Vaquero, L. M., Rodero-Merino, L., Caceres, J. and Lindner, M. (2008), A break in the Clouds: towards a Cloud definition, in: ACM SIGCOMM Computer Communication Review, vol. 39, 2008, iss. 1, pp. 50–55

Varghese, B., Akgun, O., Miguel, I., Thai, L. and Barker, A. (2016), Cloud benchmarking for maximising performance of scientific applications, in: IEEE Transactions on Cloud Computing, vol. 7, 2016, iss. 1, pp. 170–182

Varghese, B. and Buyya, R. (2018), Next generation cloud computing: New trends and research directions, in: Future Generation Computer Systems, vol. 79, 2018, pp. 849–861

Vertes, M. and Dufour, L. (2009), Migration method for software application in a multi-computing architecture, method for carrying out functional continuity implementing said migration method and multi-computing system provided therewith, Technical Report US Patent Application, Washington 2009

Vesyropoulos, N., Georgiadis, C. K. and Katsaros, P. (2018), Ensuring business and service requirements in enterprise mashups, in: Information Systems and e-Business Management, vol. 16, 2018, iss. 1, pp. 205–242

Villari, M., Fazio, M., Dustdar, S., Rana, O. and Ranjan, R. (2016), Osmotic computing: A new paradigm for edge/cloud integration, in: IEEE Cloud Computing, vol. 3, 2016, iss. 6, pp. 76–83

Vogel, L. (2015), Eclipse rich client platform, Hamburg 2015, pp. 1–818

Vogels, W. (2009), Eventually consistent, in: Communications of the ACM, vol. 52, 2009, iss. 1, pp. 40–44

Von Detten, M. and Lehrig, S. (2013), Reengineering of component-based software systems in the presence of design deficiencies–an overview, in: Proceedings of the 15th Workshop on Software-Reengineering (WSR 2013), pp. 1–2

Voorsluys, W., Broberg, J. and Buyya, R. (2011), Introduction to cloud computing, in: Cloud Computing, vol. 1, 2011, pp. 1–41

Vouk, M. A. (2004), Cloud Computing—issues, research and implementations, in: Journal of Computing and Information Technology, vol. 16, 2004, iss. 4, pp. 235–246

Wadhwa, B., Jaitly, A., Hasija, N. and Suri, B. (2015), Cloud service brokers: addressing the new cloud phenomenon, in: Proceedings of the Informatics and Communication Technologies for Societal Development (ICICTS 2015), pp. 29–40

Walker, E., Gardner, J. P., Litvin, V. and Turner, E. L. (2006), Creating personal adaptive clusters for managing scientific jobs in a distributed computing environment, in: Proceedings of the IEEE Challenges of Large Applications in Distributed Environments (CLADE 2006), pp. 95–103

Walraven, S., Truyen, E. and Joosen, W. (2011), A middleware layer for flexible and cost-efficient multi-tenant applications, in: International ACM/IFIP/USENIX Conference on Distributed Systems Platforms and Open Distributed Processing (Middleware 2011), pp. 370–389

Wang, C., Chen, Q., Chen, H. and Xu, H. (2015), An SLA-oriented Multiparty Trust Negotiation Model based on HCPN in Cloud Environment, in: International Journal of u-and e-Service, Science and Technology, vol. 8, 2015, iss. 7, pp. 321–336

Wang, J. K., Ding, J. and Niu, T. (2012), Interoperability and Standardization of Intercloud Cloud Computing, in: Distributed, Parallel, and Cluster Computing, vol. 1, 2012, pp. 1–4

Warren, I. (1999), The Renaissance of Legacy Systems, New York 1999

Weyns, D. (2017), Software engineering of self-adaptive systems: an organised tour and future challenges, in: Chapter in Handbook of Software Engineering, vol. 1, 2017, pp. 1–10

Williams, B. J. and Carver, J. C. (2010), Characterizing software architecture changes: A systematic review, in: Information and Software Technology, vol. 52, 2010, iss. 1, pp. 31–51

Winter, A. (2004), Software-Reengineering—Werkzeuge und Prozesse, in: Workshop der GI-Fachgruppe Software-Wartung, vol. 15, 2004, pp. 1–2

Winter, A. and Ziemann, J. (2007), Model-based Migration to Service-oriented Architectures: A Project Outline, in: Workshops of the Conference on Software Maintenance and Reengineering (CSMR 2007), pp. 107–110

Witell, L., Snyder, H., Gustafsson, A., Fombelle, P. and Kristensson, P. (2016), Defining service innovation: A review and synthesis, in: Journal of Business Research, vol. 69, 2016, iss. 8, pp. 2863–2872

Wittern, E., Kuhlenkamp, J. and Menzel, M. (2012), Cloud service selection based on variability modeling, in: Proceedings of the International Conference on Service-Oriented Computing (ICSOC 2012), pp. 127–141

Wittern, E. and Zirpins, C. (2016), Service feature modeling: modeling and participatory ranking of service design alternatives, in: Software & Systems Modeling, vol. 15, 2016, iss. 2, pp. 553–578

Woitsch, R. and Utz, W. (2015), Business process as a service: Model based business and it cloud alignment as a cloud offering, in: Proceedings of the International IEEE Conference on Enterprise Systems (ES 2015), pp. 121–130

Woods, S. G., Quilici, A. E. and Yang, Q. (1998), Constraint-based design recovery for software reengineering: theory and experiments, 1st Edition, New York 1998

Wu, B., Lawless, D., Bisbal, J., Grimson, J., Wade, V., O'Sullivan, D. and Richardson, R. (1997), Legacy system migration: A legacy data migration engine, in: Proceedings of the 17th International IEEE Database Conference (DATASEM 1997), pp. 129–138

Wu, F., Zhang, L. and Huberman, B. A. (2008), Truth-telling reservations, in: Algorithmica, vol. 52, 2008, iss. 1, pp. 65–79

Wu, L., Sahraoui, H. and Valtchev, P. (2005), Coping with legacy system migration complexity, in: Proceedings of the 10th International IEEE Conference on Engineering of Complex Computer Systems (ICECCS 2005), pp. 600–609

Xia, Z., Wang, X., Sun, X. and Wang, Q. (2015), A secure and dynamic multi-keyword ranked search scheme over encrypted cloud data, in: IEEE Transactions on parallel and distributed systems (TPDS 2015), vol. 27, 2015, iss. 2, pp. 340–352

Xing, Y. and Zhan, Y. (2012), Virtualization and cloud computing, in: Future Wireless Networks and Information Systems, vol. 1, 2012, pp. 305–312

Xu, J. and Palanisamy, B. (2018), Optimized contract-based model for resource allocation in federated geo-distributed clouds, in: IEEE Transactions on Services Computing, vol. 14, 2018, iss. 2, pp. 530–543

Xu, X. and Zhao, X. (2015), A framework for privacy-aware computing on hybrid clouds with mixed-sensitivity data, in: Proceedings of the 2015 17th International IEEE Conference on High Performance Computing and Communications (HPCC 2015), 7th IEEE International Symposium on Cyberspace Safety and Security (CSS 2015), and 12th International IEEE Conference on Embedded Software and Systems (ICESS 2015), pp. 1344–1349

Xu, X. (2012), From cloud computing to cloud manufacturing, in: Robotics and computer-integrated manufacturing, vol. 28, 2012, iss. 1, pp. 75–86

Yamazaki, Y. (2004), Dynamic Collaboration: The model of new business that quickly responds to changes in the market through the integrated IT/Network Solutions provided by NEC, in: NEC journal of advanced technology, vol. 1, 2004, iss. 1, pp. 9–16

Yangui, S. and Tata, S. (2016), An OCCI compliant model for PaaS resources description and provisioning, in: Computer Journal, vol. 59, 2016, iss. 3, pp. 308–324

Yanosky, R. (2008), From users to choosers: The cloud and the changing shape of enterprise authority, in: Katz, Richard N. (Ed., 2008), pp. 126–136

Yau, S. S., Nicholl, R. A., Tsai, J. J.-P. and Liu, S.-S. (1988), An integrated life-cycle model for software maintenance, in: IEEE Transactions on Software Engineering, vol. 14, 1988, iss. 8, pp. 1128–1144

Youseff, L., Butrico, M. and Da Silva, D. (2008), Toward a Unified Ontology of Cloud Computing, in: Grid Computing Environments Workshop (GCE 2008), pp. 1–10

Yu, D. (1991), A view on three R's (3Rs): reuse, re-engineering, and reverse-engineering, in: ACM SIGSOFT Software Engineering Notes, vol. 16, 1991, iss. 3, pp. 1–69

Yumerefendi, A. R. and Chase, J. S. (2007), Strong accountability for network storage, in: ACM Transactions on Storage (TOS 2007), vol. 3, 2007, iss. 3, pp. 1–11

Zachos, K., Lockerbie, J., Hughes, B. and Matthews, P. (2011), Towards a framework for describing cloud service characteristics for use by chief information officers, in: Proceedings of the IEEE Workshop on Requirements Engineering for Systems, Services and Systems-of-Systems (RESS 2011), pp. 16–23

Zafar, F., Khan, A., Suhail, S., Ahmed, I., Hameed, K., Khan, H. M., Jabeen, F. and Anjum, A. (2017), Trustworthy data: A survey, taxonomy and future trends of secure provenance schemes, in: Journal of Network and Computer Applications, vol. 94, 2017, pp. 50–68

Zalila, F., Challita, S. and Merle, P. (2019), Model-driven cloud resource management with OCCIware, in: Future Generation Computer Systems, vol. 99, 2019, pp. 260–277

Zatonatska, T. and Dluhopolskyi, O. (2019), Modelling the efficiency of the cloud computing implementation at enterprises, in: vol. 3, 2019, pp. 45–59

Zeleny, M. and Cochrane, J. L. (1973), Multiple criteria decision making, Columbia 1973

Zhang, F., Tang, X., Li, X., Khan, S. U. and Li, Z. (2019), Quantifying cloud elasticity with container-based autoscaling, in: Future Generation Computer Systems, vol. 98, 2019, pp. 672–681

Zhang, Q., Cheng, L. and Boutaba, R. (2010), Cloud computing: state-of-the-art and research challenges, in: Journal of internet services and applications, vol. 1, 2010, iss. 1, pp. 7–18

Zhang, Q., Li, S., Li, Z., Xing, Y., Yang, Z. and Dai, Y. (2015), CHARM: A cost-efficient multi-cloud data hosting scheme with high availability, in: IEEE Transactions on Cloud computing, vol. 3, 2015, iss. 3, pp. 372–386

Zhong, R. Y., Xu, X., Klotz, E. and Newman, S. T. (2017), Intelligent manufacturing in the context of Industry 4.0: a review, in: Engineering, vol. 3, 2017, iss. 5, pp. 616–630

Zhou, Y., Zhang, D. and Xiong, N. (2017), Post-cloud computing paradigms: a survey and comparison, in: Tsinghua Science and Technology, vol. 22, 2017, iss. 6, pp. 714–732

Zhu, W., Zhuang, Y. and Zhang, L. (2017), A three-dimensional virtual resource scheduling method for energy saving in cloud computing, in: Future Generation Computer Systems, vol. 69, 2017, pp. 66–74

Ziegler, W. (2012), SLAs for energy-efficient data centres: The standards-based approach of the OPTIMIS project, in: International Workshop on Energy Efficient Data Centers, pp. 37–46

Zimmer, M., Baars, H. and Kemper, H.-G. (2012), The impact of agility requirements on business intelligence architectures, in: Proceedings of the 45th IEEE Hawaii International Conference on System Sciences (HICSS 2012), pp. 4189–4198

Zou, Y. and Kontogiannis, K. (2002), Migration to object oriented platforms: A state transformation approach, in: Proceedings of the International Conference on Software Maintenance (CSM 2002), pp. 530–539

Erklärung

Hiermit versichere ich,

- dass die Arbeit selbstständig verfasst wurde,
- dass keine anderen als die angegebenen Quellen benutzt und alle wörtlich oder sinngemäß aus anderen Werken übernommenen Aussagen als solche gekennzeichnet wurden,
- dass keine anderen als die angebenden Hilfsmittel verwendet wurden,
- dass die eingereichte Arbeit weder vollständig noch in wesentlichen Teilen Gegenstand eines anderen Prüfungsverfahrens war,
- dass die Arbeit weder vollständig noch in Teilen bereits veröffentlicht wurde und
- dass das elektronische Exemplar mit den anderen Exemplaren übereinstimmt.

Stuttgart, den 31. Januar 2023, Unterschrift

Adrián Juan Verdejo