Institute of Architecture of Application Systems

University of Stuttgart
Universitätsstraße 38
D–70569 Stuttgart

Diplomarbeit Nr. 3743

# Introduction of Cloud Provider-specific Knowledge and Further Evaluation of the Cloud Decision Support Framework

Thi Minh Thu Do

| | |
|---|---|
| **Course of Study:** | Softwaretechnik |
| **Examiner:** | Prof. Dr. Frank Leymann |
| **Supervisor:** | Dr. Vasilios Andrikopoulos,<br>Dipl.-Inf. Johannes Wettinger |
| **Commenced:** | March 1, 2016 |
| **Completed:** | December 7, 2016 |
| **CR-Classification:** | D.2.7, H.3.4, H.3.5, H.4.2 |

# Contents

List of Abbreviations 6

1 Introduction 9

2 Related work 11
2.1 Fundamentals . . . 11
2.2 Cloud Migration Decision Support Systems . . . 15
2.3 Cloud Decision Support Framework (CloudDSF) . . . 19
2.4 Cloud Decision Support Framework Plus (CloudDSF+) . . . 24

3 Extension of the CloudDSF+ Knowledge Base 35
3.1 The Decision Select Cloud Vendor . . . 35
3.2 Cloud Vendors in the Knowledge Base . . . 36
3.3 New Relation Types . . . 51
3.4 Select Cloud Vendor Outcomes and Relations to Other Outcomes . . . 51
3.5 Relations toward Outcomes of the decision Select Cloud Vendor . . . 67
3.6 Summary of the Extension . . . 76

4 Implementation and evaluation of Cloud Decision Support Framework Plus 2 (CloudDSF+2) Prototype 81
4.1 Implementation of CloudDSF+2 . . . 81
4.2 Evaluation of CloudDSF+2 . . . 84

5 Conclusion and future work 91

Bibliography 95

# List of Figures

2.1 An overview of Legacy-to-Cloud Migration Horseshoe Framework[AB14] 19
2.2 Conceptual view of the decision support framework for cloud migration[ASL13] . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 20
2.3 Prototypical Implementation - Force Layout[Daw14] . . . . . . . . . . . 25
2.4 Outcome relations layout with collapsed decisions and collapsed Select Service Provider/Offering decision point[Bal15] . . . . . . . . . . . . . . 31
2.5 Part of the KBNavigator showing the relations to all specified outcomes and a conflict at the On-Premise Hosting outcome[Bal15] . . . . . . . . 32

3.1 Hierarchical layout of the CloudDSF+ Prototype visualizing part of the knowledge base . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 36
3.2 Public Cloud Usage 2015 [Rig14] . . . . . . . . . . . . . . . . . . . . . . 37
3.3 Enterprise Public Cloud Usage 2015 vs. 2014 [Rig14] . . . . . . . . . . . 38

4.1 New outcomes in the hierarchical layout . . . . . . . . . . . . . . . . . . 82
4.2 Arrows presenting relations between decisions in new style . . . . . . . 82
4.3 Arrows presenting relations between outcomes in new style . . . . . . . 82
4.4 Selection of platform services for migrating Website . . . . . . . . . . . . 85

# List of Tables


2.1 Approaches in decision support for application migration to the cloud, based on [Daw14] and [Bal15] . . . . . . . . . . . . . . . . . . . . . . . 15
2.2 The Cloud Decision Support Framework (CloudDSF) Knowledge Base[Aa14] 21
2.3 The refined CloudDSF Knowledge Base[Bal15] . . . . . . . . . . . . . . . 26
2.4 Definition of relationship types between outcomes[Bal15] . . . . . . . . 30

3.1 Definition of new relationship types between outcomes . . . . . . . . . . 52
3.2 Relations between outcomes of the decision Select Cloud Vendor toward outcomes of the decision Define Scalability Level . . . . . . . . . . . . . 54
3.3 Relations between outcomes of the decision Select Cloud Vendor toward outcomes of the decision Select Scaling Type . . . . . . . . . . . . . . . . 56
3.4 Relations between outcomes of the decision Select Cloud Vendor toward outcomes of the decision Select Elasticity Automation Degree . . . . . . 58
3.5 Relations between outcomes of the decision Select Cloud Vendor toward outcomes of decision Select Scaling Trigger . . . . . . . . . . . . . . . . 59
3.6 Relations between outcomes of the decision Select Cloud Vendor toward outcomes of the decision Select Multi-Tenancy Level . . . . . . . . . . . 61
3.7 Relations between outcomes of the decision Select Cloud Vendor toward outcomes of the decision Select Cloud Deployment Model . . . . . . . . 63
3.8 Relations between outcomes of the decision Select Cloud Vendor toward outcomes of the decision Select Cloud Service Model . . . . . . . . . . . 64
3.9 Relations between outcomes of the decision Select Cloud Vendor toward outcomes of the decision Define Cloud Hosting . . . . . . . . . . . . . . 66
3.10 Relations between outcomes of the decision Select Cloud Vendor toward outcomes of the decision Select Pricing Model . . . . . . . . . . . . . . . 68
3.11 Relations between outcomes of the decision Select Cloud Vendor toward outcomes of the decision Define Resource Location . . . . . . . . . . . . 69
3.12 Relations between other outcomes in the CloudDSF+2 knowledge base toward services of Amazon Web Services . . . . . . . . . . . . . . . . . . 70
3.13 Relations between other outcomes in the CloudDSF+2 knowledge base toward services of Microsoft Azure . . . . . . . . . . . . . . . . . . . . . 72
3.14 Relations between other outcomes in the CloudDSF+2 knowledge base toward services of Google Cloud Platform . . . . . . . . . . . . . . . . . . 74

3.15 Quantification of outcome relations(Amazon Web Services) . . . . . . . . 77
3.16 Quantification of outcome relations (Microsoft Azure) . . . . . . . . . . . 78
3.17 Quantification of outcome relations (Goolge Cloud Platform) . . . . . . . 79

# List of Abbreviations

| Abbrv. | | Meaning |
|---|---|---|
| CloudDSF | = | Cloud Decision Support Framework |
| CloudDSF+ | = | Cloud Decision Support Framework Plus |
| CloudDSF+2 | = | Cloud Decision Support Framework Plus 2 |
| IaaS | = | Infrastructure as a Service |
| PaaS | = | Platform as a Service |
| SaaS | = | Software as a Service |
| SLA | = | Service Level Agreement |

# 1 Introduction

Cloud computing is now seen as an advanced information technology that strongly supports business enterprises by offering them more flexibility at a lower cost. Due to the beneficial characteristics like on-demand provisioning and billing against the actual consumption of computing resources, enterprises try to leverage the advantages of cloud computing, not only for their new applications, but also for existing legacy ones by migrating them into the cloud.

The biggest challenge of the application migration is to protect sensitive business data against different types of attacks. As cloud computing is getting more and more mature with viable security measures, the number of enterprises utilizing cloud computing services continues to grow significantly during the last few years [Rig14]. Hybrid cloud, a mixed cloud computing environment that hosts one part of the application on-premise and the rest off-premise, and platform services, which support application development and offer middleware capabilities, are gaining more and more popularity [Rig14]. As a result, it is estimated that not only information technology infrastructures will be provisioned in the cloud but also highly customized and important applications like enterprise resource planning systems will be moved to the cloud [Rig14], [Gar13].

In the past, the application migration oriented mainly toward moving the whole application into the cloud through virtualization technology[ABLS13]. However, legacy applications need a more sophisticated migration mechanism in order to find out the most suitable cloud environment and fully leverage the benefits of cloud computing [ABLS13], [VA12]. Consequently, the application migration decision set is extended to cover different technical and non-technical aspects such as which components should be migrated, with which migration measures and if it is financially beneficial to migrate. Although several different approaches exist to support decision makers to move their applications into the cloud, further research in this area is still required [JAP13].

In [Aa14], the CloudDSF has been developed, which is a conceptual view of the decisions and tasks that should be considered when migrating an application into the cloud. CloudDSF aims at supporting decision makers in evaluating if the migration is really needed and in making necessary decisions before the actual migration is carried out. CloudDSF defines ten tasks and four main decision points which subsume multiple decisions which in turn subsume multiple outcomes. A prototypical implementation of CloudDSF, the CloudDSF Prototype has been developed. The relationships in the CloudDSF Prototype are only defined on the level of decisions but not on the level of concrete outcomes [Daw14].

The CloudDSF was further developed into CloudDSF+ in [Bal15] by:

- refining the CloudDSF knowledge base and reviewing the relations between decisions,
- elaborating and defining the relations between outcomes,
- developing an appropriate visualization mechanism for the extended decision support framework.

The extended framework CloudDSF+ offers decision makers the ability to select specific outcomes relating to scalability, multi-tenancy and other deployment options for the migration strategy. However, with CloudDSF+ it is still not possible to select actual cloud computing services that satisfy the selected dimensions of the migration strategy, because such services are not yet included in the CloudDSF+ knowledge base. In order to make CloudDSF+ more practically useful, this thesis is motivated to extend the CloudDSF+ into CloudDSF+2.

The structure of this thesis is as follows. Chapter 2 updates the state of the art in decision support systems for application migration to the cloud. Chapter 3 extends the CloudDSF+ knowledge base by introducing popular cloud computing services offered by major cloud vendors and then defining relations between those newly added outcomes with other existing ones in the knowledge base and vice versa. Chapter 4 describes the improvement of CloudDSF+ prototype and introduces further evaluation of the framework when new services are added. Finally, chapter 5 summarizes the thesis and gives a brief overview of future work.

# 2 Related work

This chapter defines the terminology, concepts and definitions related to this thesis in its first section, aiming at a unified understanding of the thesis. Subsequently, related works in decision support for application migration to the cloud will be introduced. Finally, the decision support framework CloudDSF as well as its extended and refined framework CloudDSF+ will be described in detail.

## 2.1 Fundamentals

**Cloud Computing**

This thesis is the follow-up work of the [Aa14], [Daw14] and [Bal15]. As a result, the same definition of cloud computing from National Institute of Standard and Technology (NIST) is used, as stated in the following paragraph.

> *Cloud Computing is 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. This cloud model is composed of five essential characteristics, three service models and four deployment models. [MG11]*

The five essential characteristics of cloud computing (i.e. on-demand self-service, broad network access, resource pooling, rapid elasticity and measured service), its three service models (i.e. Software as a Service (SaaS), Platform as a Service (PaaS) and Infrastructure as a Service (IaaS)) and its four deployment models (i.e. private cloud, community cloud, public cloud and hybrid cloud) were explained in more details in many sources, among them are [Ba12a], [MG11] and [BBG11].

As defined in [Ba12a], cloud providers offer cloud services and cloud customers logically consume the offered services. Cloud customers are not necessarily end-users but can be providers who in turn provide cloud services to other customers. As a result, there exists many cloud providers and at the same time cloud consumers and vice versa in the cloud market.

The utilization of cloud services is determined by a contract between a cloud provider and a cloud customer, including Service Level Agreement (SLA), a technical performance document in which the cloud provider states a collection of promises made to the customer (i.e. availability, remedies for failure to perform, data preservation, legal care of consumer information), a collection of limitations (i.e. scheduled outages, force majeure events, service agreement changes, security and service API changes) and a set of obligations that the consumer must accept (i.e. acceptable use policies, licensed software and timely payment)[Ba12a]. An SLA is therefore an important document signed and followed by both cloud providers and cloud customers, except when the terms are explicitly otherwise stated.

### Application Migration to the Cloud

According to the software maintenance standard of the International Organization for Standardization (ISO) and the International Electrotechnical Commission (IEC), adaptive maintenance is defined as "the modification of software product, performed after delivery, to keep a software product usable in a changed or changing environment" [Ins06]. Migrating software applications to the cloud is then considered a special form of adaptive maintenance, because it also aims at maintaining the functionality of legacy software in a different or changing operating environment. Through out this work, the following definition of application migration is in use: *Application migration to the cloud can therefore be described as moving an application from a local data center into a cloud environment without changing its functionality or decreasing its performance[VA12].*

Business organizations are motivated to move their application systems to the cloud because they gain through application migration the benefit of direct cost savings, of productivity improvements and of innovation[KPM12].

The traditional on-premise infrastructure which accommodates a fixed computing capacity causes the risk of overloading or under-utilization of resources. With the virtualized infrastructure of cloud computing, the huge hardware acquisition and maintenance costs within organizations are reduced to services utilization fees paid to cloud providers on the basis of on-demand and per-use model. In this way the economies of scale realized by large data centers are exploited, bringing about monetary saving and at the same time freeing organizations from the burden of low-level tasks like infrastructure management, which enables them to focus on more important business related activities[IBM10], [KPM12] and [Sa11].

Productivity improvements of organizations are interpreted as increased output per unit of cost[KPM12]. More precisely, cloud computing enables organization's capital to be spent more efficiently. Before the era of cloud computing, changes to business could not be made without detailed capacity planning, changes to existing technology or purchases of new technology, which all involve large capital investment with long payback period and functional instability due to under-utilization or lack of resources during different time periods[Sa11]. Using cloud computing, such business changes require less IT related plans and actions. The fraction of output/investment in the case of cloud computing is therefore obviously greater than without it.

Innovation of business organizations means the ability to deliver new and evolving products[KPM12]. IBM Academy of Technology mentioned in [IBM10] the analytics and sense-and-respond capabilities of organizations who consume cloud computing services. A sense-and-respond organization is capable of recognizing change of the market early and then analyzes and acts in response to the change. Cloud computing supports such organizations in the way that it enables quick business re-engineering, increased availability, elastic scalability and simpler management[KPM12].

On the other hand, the challenges and risks accompanying cloud computing discourage the business organizations to adopt cloud services. Enterprises put their biggest concern on security related issues, because cloud computing creates prospective points of attack that do not exist in traditional on-premise IT environment. Cloud Security Alliance lists top nine threats regarding security in cloud computing: data breaches, data loss, account or service traffic hijacking, insecure interfaces and APIs, denial of service, malicious insiders, abuse of cloud services, insufficient due diligence and shared technology vulnerabilities[Clo13].

With respect to business organizations, their sensitive business data should be kept unknown to the wider world, especially their business competitors. As a result, cloud providers should guarantee that data of customers are stored at a secure location, in a secure manner and protected by security measures. The data exchange across enterprise and clouds should also be protected. Those concerns are understandable against the aforementioned threats, a lack of transparency regarding how and where providers store customer data and which security measures they implement to prevent unauthorized access[KPM12], [IBM10], [KPM11].

Further obstacles affecting adoption and growth of cloud computing are business continuity, compliance, data transfer bottlenecks, integration to internal system, lack of expertise and the difficulties of the management of multiple cloud services[Aa10], [Rig14]. However, the surveys conducted by RightScale Inc. shows that the benefits of cloud computing is increasing considerably with the better expertise of enterprises in cloud computing. At the same time, the annual cloud computing adoption rate increases because challenges are decreasing sharply[Rig14].

An increasing number of enterprises are utilizing or planning for hybrid cloud strategy, which allows to seamlessly transfer workload to the cloud when local computing resources are insufficient. In this way enterprises are able to take the benefit of cloud computing and at the same time minimize number and severity of security threats[Rig14], [Gar13]. However, hybrid cloud scenarios are often complex and more difficult to implement in contrast to private or public clouds[Ba12a], [BBG11].

The process of migrating a legacy application to the cloud often requires reengineering efforts[ABLS13], [Sa05]. Due to the fact that the architecture of the cloud and in some cases of the existing applications is complex, it is neccessary to conduct an assessment prior to the migration to see if the migration is reasonable and benificial or not[JAP13], [KPM11]. In the assessment, other non-technical factors like organization, legislation, compliance, finance and technical factors like existing infrastructure, IT skills and application architecture can be as well under consideration[JAP13].

### Application Migration and Decision Support

It is necessary to support the decision for application migration to the cloud, because the task is complex, refers to many implementation possibilities and can lead to business changes. Decision support can generally be realized by a decision support system (DSS). A detailed explanation of decision support, DSSs and their architecture can be found in [BH08] and [Pow02]. The latter document defines five different types of DSSs, among them knowledge-driven DSS, which recommends users with suitable actions based on the knowledge about a specific problem domain. This approach is applied to the decision support systems CloudDSF[Daw14] and CloudDSF+[Bal15], which help to select a cloud provider for an application migration.

Migrating an application to the cloud is classified as a multiple-criteria decision making (MCDM) problem[Daw14], [FGE05], which can be solved by several approaches available in [FGE05]. Relating techniques include analytic hierarchy process (AHP) and analytic network process (ANP). The former is used more widely in research due to its lower complexity. However, AHP requires that the criteria are shaped into a hierarchical order, which is often not possible since several criteria might be related and the alternatives may effect criteria across levels. From the same set of decisions, it is possible that different hierarchies are constructed, resulting in different decision outcomes. As a consequence, various techniques are applied for recent application migration decision support approaches.

As mentioned above, CloudDSF (see Section 2.3), with the visualization of necessary decisions and their relationships, offers the ground for the knowledge-driven decision support system for application migration to the cloud. CloudDSF+ (see Section 2.4)

**Table 2.1:** Approaches in decision support for application migration to the cloud, based on [Daw14] and [Bal15]

| Name | Year | Reference(s) | Method |
|---|---|---|---|
| Cloudward Bound | 2010 | [HSS+10] | Integer Linear Programming |
| The Cloud Adoption Toolkit | 2012 | [Ka12] | Checklist |
| CloudStep | 2012 | [Ba12b] | Question-based |
| CloudGenius | 2012 | [MR12], [Ma14] | AHP |
| Towards Process Support for Migrating Applications to Cloud Computing | 2012 | [CA12] | Step-Based |
| ARTIST | 2013 | [Ma13], [Aa13] | Model-Driven |
| CloudMIG | 2013 | [Fre14] | Architecture/Model-Driven |
| Moving Business Intelligence to Cloud Environments (InCLOUDer) | 2014 | [Ja14] | AHP |
| Legacy-to-Cloud Migration Horseshoe | 2014 | [AB14] | Architecture-Driven |

refines the decisions available in CloudDSF and extends the system to the point that a single optimized solution of the migration problem can be determined.

## 2.2 Cloud Migration Decision Support Systems

In a survey conducted by Jamshidi, Ahmad, and Pahl in 2013 [JAP13], twenty three studies regarding until then existing cloud migration decision support systems were systematically reviewed. The approaches that most relevant to CloudDSF were briefly described in [Daw14]. In [Bal15], four more approaches were summarized at the status of 2013. All of these systems are listed in Table 2.1 and the latter four are further described in the following.

#### Advanced software-based service provisioning and migration of legacy software (Artist)

Artist is a model-driven approach which offers a legacy system migration framework together with supporting tools. Artist differs from other projects in the way that it does not exclusively support the software migration, but also suggests activities before and after the migration is really conducted. The complete process consists of four main phases:

- **Premigration**: In this phase a technical and economic feasibility study is conducted in order to see if the migration is beneficial. Furthermore the consequences of possible migration strategies are analyzed.
- **Migration**: In this phase the migration is actually implemented. First of all, through reverse engineering a platform-specific model is created, consolidated and then transformed into a platform-independent model. This process makes use of the patterns across multiple migration/modernization scenarios. Secondly, the performance and usage characteristics of application elements are examined and profiled to define the necessary target environment. At the next steps, the platform-independent model is transformed, with regard to the requirements of the target cloud environment, into a cloud-specific model which is in its turn transformed via forward engineering into the executable migrated software.
- **Post-migration**: Major activities of this phase include deployment of the modernized application components onto the target environment and objectives validation check.
- **Migration Artifacts Reuse and Evolution**: This phase includes application maintenance activities after migration to the cloud such as software updates or cloud provider changes. Furthermore, artifacts produced along the migration process that can be reused across projects are made available via a marketplace.

According to [Ma13], Artist has following advantages over other approaches:

- safe investment in software migration/modernization due to feasibility study
- focus on cloud-compliant architectural issues at application and infrastructure levels
- delivery of business model issues that strongly linked to technical decisions
- considers the impact of business model shift on organization processes
- enables reusability and automation
- prepares the software for its evolution

### CloudMIG

The CloudMIG[Fre14] approach specifies three challenges of the application migration process:

- **Cloud environment constraints - CEC**: correspond to limitations caused by cloud environment, e.g. direct access to data system or opening of network sockets by guest applications.
- **CEC violations**: happen when the guest applications e.g. write to the data system or open the network sockets. Those CEC violations are usually not systematically checked during the migration process.
- **Cloud deployment options - CDOs**: There is variety of application deployment possibilities, but comparison regarding their performance and costs is not fully provided.

CloudMIG then aims at supporting SaaS providers to overcome those challenges during application migration to IaaS- and PaaS-based cloud environment. It focuses on:

- recognizing CEC violations using automatic conformance checking approach with the help of constraint validation
- creating and optimizing CDOs with the help of simulation-based genetic algorithm CDOXplorer

CloudMIG has the advantage of significantly simplifying the recognization of CEC violations as well as the creation of suitable CDOs. Once CEC violations are recognized, SaaS providers can save the time-consuming and expensive code reviews and no longer have the risk of malfunctions. CDOs can be automatically generated, all the manual implementation, evaluation and comparison can be saved.

### InCLOUDer

The ultimate purpose of InCLOUDer cloud migration decision support system is to rank the different alternatives to migrate an application to the cloud. In order to realize the work, InCLOUDer first of all provides a formal description of the parameters that affect the migration of applications to the cloud, including the application to be migrated, the selected cloud service offerings (CSO) and the requirements of the organization in consideration with its criteria and constraints.

At the next step, unsuitable migration alternatives according to the specified constraints are discarded. The information regarding criteria is then collected from organization and arranged into a hierarchy of criteria using the Analytical Hierarchy Process (AHP). The rankings for every alternative are calculated based on those criteria. It is possible in InCLOUDer to weight criteria expressed in different dimensions.

Generating the alternatives, rejecting the non-viable ones and weighting them for every criterion are automatically conducted. However, the InCLOUDer approach is considered semi-automatic, because it officially defines a five-step process (see below) to find the optimal cloud migration strategy[Ja14] and only two of them are automated.

- **Step 1**: modeling the problem as a hierarchy with a goal, criteria and cloud migration alternatives
- **Step 2**: prioritizing the criteria
- **Step 3**: evaluating the different alternatives for every criterion
- **Step 4**: checking the consistency of the judgments
- **Step 5**: coming to a decision

InCLOUDer also provides a prototype of its decision support system, which offers three EMF-based[1] editors for the organization to model the input required by InCLOUDer, namely the application, the criteria and the CSO. Given these three models, the prototype generates suitable alternatives and automatically weights them to find out the highest ranked migration strategy. Furthermore the prototype implements two feedback mechanisms to improve the migration support and let the organization to drive the decision-making regardless of the migration strategy suggested.

### Legacy-to-Cloud Migration Horseshoe

This approach[AB14] extends the classical re-engineering horseshoe model[KWC98] and OMG's Architecture Driven Modernization (ADM) framework[2] and presents a framework of four processes to migrate legacy applications to the cloud, each process is further refined with sub-processes and activities as seen in Figure 2.1.

The proposed framework takes legacy source code as input and uses software re-engineering concepts to recover the architecture from code. Then it exploits the software evolution concepts to support architecture-driven migration of legacy systems to cloud-based architecture and generates cloud-enable code. This is a round-trip engineering which can be repeated on the target code (cloud-enable) for refinements.

[1]Eclipse Modeling Framework: www.eclipse.org/modeling/emf/
[2]Object Management Group:http://www.omg.org/adm

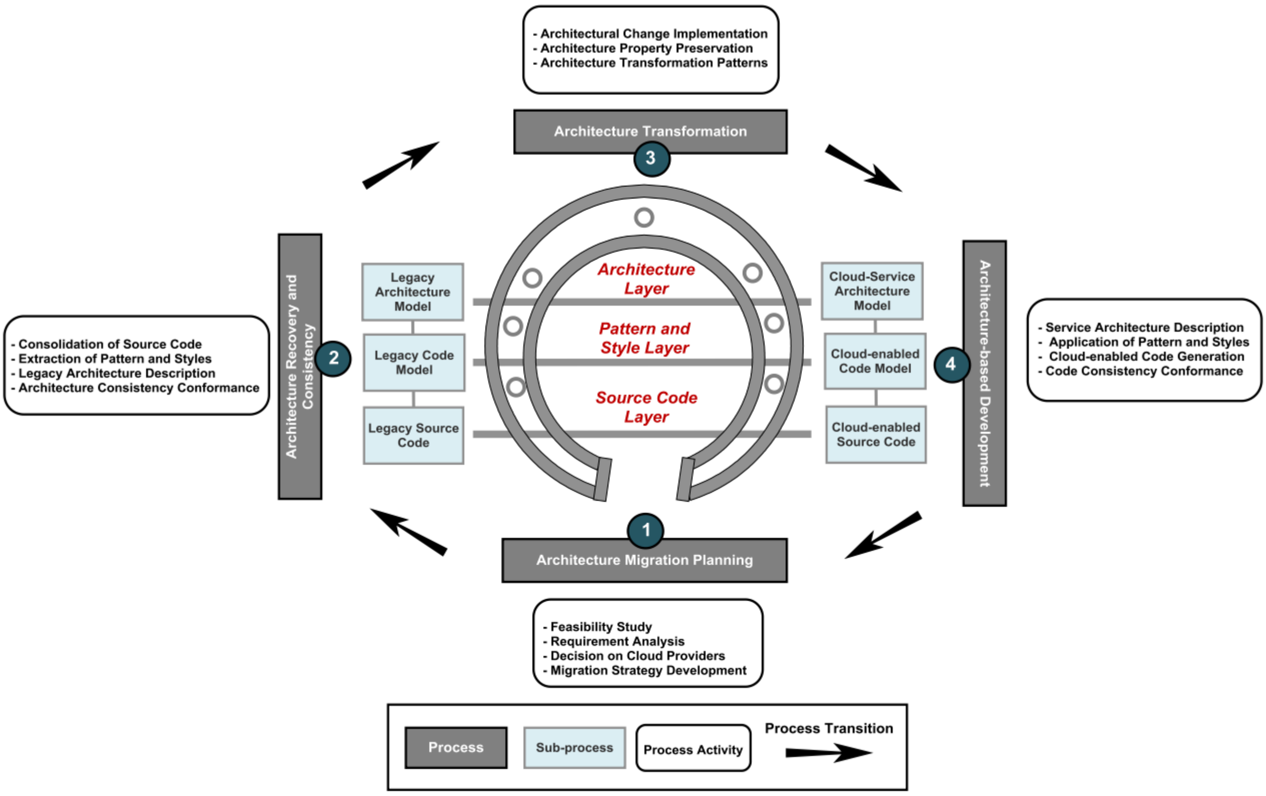


**Figure 2.1:** An overview of Legacy-to-Cloud Migration Horseshoe Framework[AB14]

As stated in[AB14], this project is less comprehensive in comparison with Artist because it focuses merely on architectural-driven migration. That's why it takes less effort to enable a migration using legacy-to-cloud migration Horseshoe.

## 2.3 Cloud Decision Support Framework (CloudDSF)

### Conceptual framework

Before [ASL13] was presented in 2013, decision support systems for application migration to the cloud focused mostly on one specific type of migration decision, e.g. the selection of an IaaS provider that best supports application performance at the lowest cost. Andrikopoulus, Strauch and Leymann introduced in their paper a new concept of cloud decision support system, which sees the migration task as a multi-dimensional problem with multiple decision points and related analysis tasks as can be seen in Figure 2.2. Their conceptual CloudDSF was built on the argument that new cloud offerings enable various migration scenarios. As a result, decision support needs to include

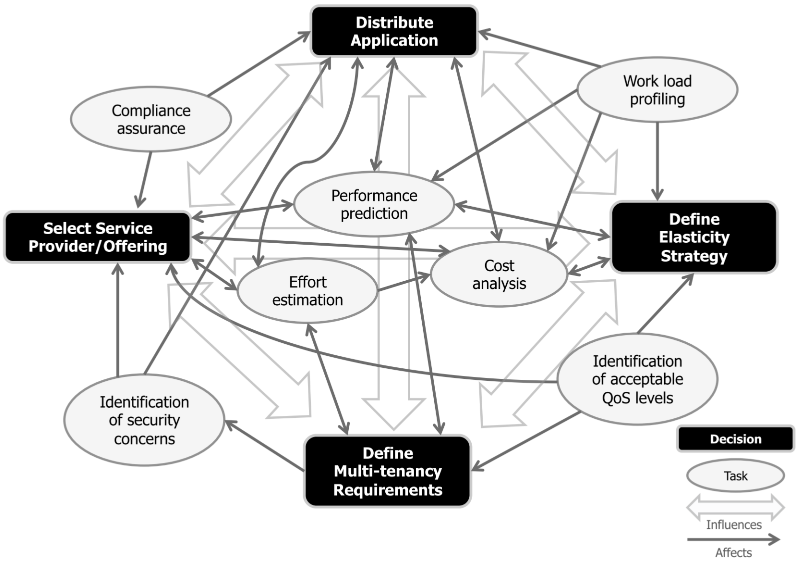


**Figure 2.2:** Conceptual view of the decision support framework for cloud migration[ASL13]

options for segmenting and distributing applications in the cloud, scaling strategies and implementation of multi-tenancy.

### Elaborated version of CloudDSF

CloudDSF[Aa14], [Daw14] was developed in order to add more decision points to the CloudDSF. The knowledge base on which CloudDSF based itself contains four decision points and their belonging sub-decisions which in turn subsume multiple outcomes. Furthermore a visualization of this knowledge base is provided to actually support the decision making process. Table 2.2 provides an overview of the knowledge base.

### Decision points

Based on the definitions stated in [ASL13], [Aa14] and [Daw14], the four decision points of the CloudDSF will be briefly described as following.

- **Application Distribution**: This decision point determines the distribution of applications across service providers as well as between the cloud and the local data center. Distribution can be decided based on logical layers, physical tiers or

**Table 2.2:** The CloudDSF Knowledge Base[Aa14]

| Decision Point | Decision | Outcomes |
|---|---|---|
| Application Distribution | Select Application Layer | - Presentation/Business/Data Layer<br>- Multiple Layer |
| | Select Application Tier | - Client/Application/Data Tier<br>- Multiple Tiers |
| | Select Application Components | - Single Components<br>- Multiple Components |
| | Select Migration Type | - Type I, II, III or IV |
| Elasticity Strategy | Define Scalability Level | - Instance/Container/VM/Virtual Resource/Hardware Level<br>- Mutiple Level |
| | Select Scaling Type | - Vertical/Horizontal Scaling<br>- Hybrid Scaling |
| | Select Elasticity Automation Degree | - Manual Scaling<br>- Semi-automatic Scaling<br>- Automatic Scaling |
| | Select Scaling Trigger | - Event-driven<br>- Proactive |
| Multi-Tenancy Requirements | Select Kind of Multi-Tenancy | - Multiple Instances Multi-Tenancy<br>- Native Multi-Tenancy |
| | Select Multi-Tenancy Architecture | - Any of the Possible Combination |
| Provider/Offering Selection | Select Cloud Deployment Model | - Private/Community/Public/Hybrid Cloud |
| | Select Cloud Service Model | - S/P/IaaS |
| | Define Cloud Hosting | - On Premise/Off Premise<br>- Hybrid Hosting |
| | Denfine Roles of Responsibility | - Ownership/Operation/Management Role<br>- Any Combination of Roles |
| | Select Pricing Model | - Free/Pay-per-Use/-Unit/Subscription<br>- Combined Model |
| | Select Cloud Vendor | - Evaluated Vendor |
| | Define Resource Location | - Evaluated Physical Resource Location |

components that span multiple layers and tiers. It is also possible to use one of the four migration types defined in [ABLS13]

- **Define Elasticity Strategy**: This decision point determines the elasticity strategy that the application should implement to satisfy SLA's requirements and expectations of users. The first decision regarding elasticity strategy refers to the scalability level required for the application (e. g. physical hardware level, virtualization level built above hardware or application level on top of the stack). Further decisions are what kind of scaling type as well as trigger should be used and the desired extend of automation.
- **Define multi-tenancy requirements**: This decision point determines if the application is required to support multi-tenancy, if it is designed for this purpose and how it should be engineered or re-engineered to support multi-tenancy. The fact that two types of multi-tenancy can be realized at different system levels (hardware, virtualization and application level) points out different possibilities to share resources between tenants. These possibilities were considered and included as decisions.
- **Select Service Provider/Offering**: This decision point determines the provider and offerings that satisfy application needs in terms of cost, performance, compliance and security. It covers fundamental decisions like deployment model, service model and cloud hosting, which have severe impact on other decisions. Furthermore, organizational decisions such as definition of roles of responsibility, pricing model, selection of a specific cloud vendor and resource location are also parts of this decision point.

### Analysis Tasks

In CloudDSF exists a set of ten analysis tasks, seven of which were defined in [ASL13] and the three additional ones were described in [Daw14].

- **Workload Profiling**: The estimated workload profile serves as an prerequisite for carrying performance calculation and cost analysis. It is also input for decisions relating how to distribute the application and which elasticity strategy fits this profile.
- **Cost Analysis**: This task has bilateral affecting relation with application distribution, elasticity strategy and service provider selection decisions. It should also take in consideration the estimated effort for adapting the application to operate in the cloud.

- **Effort Estimation**: This task provides the estimated amount of work required to adapt the application so that it operates well in the cloud. The work on its turn depends on the selected type of migration and the architectural layer that is affected. As a result, this task requires input from the application distribution, service provider selection and multi-tenancy requirements decisions and may lead to changes to these decisions.
- **Performance Prediction**: Similarly to the effort estimation, this task takes input from the application distribution, service provider selection and multi-tenancy requirements decisions and gives out the non-functional behavior of the application after it is migrated to the cloud. This behavior report can be used to change previous decisions, making a feedback loop between them.
- **Identification of Acceptable QoS levels**: The acceptable levels for QoS characteristics are inferred from the planned and existing SLAs. This task then supports the selection of service provider and definition of an appropriate elasticity strategy, making sure that these selection and definition fit well with QoS levels for the given workload profile. Furthermore, the task also constrains the options for application multi-tenancy.
- **Compliance Assurance**: This task affects directly the selection of service providers, especially in terms of location of data service, and the distribution of applications, which sometimes requires that personal data should be retained on-premises.
- **Identification of Security Concerns**: This task defines which data and communications are critical to be protected. Hence it affects the selection of service providers that fulfill the security constraints. Further more it has impact on the development of applications with multi-tenancy characteristic, because constrains with respect to data isolation should be considered during design phase.
- **Workforce Capabilities Identification**: Consuming cloud services requires, especially in case of private and hybrid cloud, some cloud expertise in terms of new roles, tasks and skills. This task evaluates the capability of cloud consumer and at the same time identifies their capability deficiencies, which service as input for calculating training costs and knowledge acquisitions. The influenced decisions are therefore cloud hosting, roles of responsibility and the elasticity automation degree.
- **Application Analysis**: An analysis of existing application regarding architecture, programming language, current hardware and other characteristics supports the decision for application distribution. Workload profiling, performance prediction and the identification of acceptable QoS levels are related to this task.

- **Vendor Evaluation**: The other tasks of CloudDSF focus mainly on technical aspects of cloud vendors. This task on the other hand presents non-technical benchmarks for evaluating a cloud vendor, including reputation (e.g. reference projects, benchmarks, certificate, reports) and capability (e.g. resources, knowledge, technical and business skills). The combination of these two aspects supports a decision for an appropriate cloud vendor based on both business-related and technical considerations.

#### Relations between decisions

Two types of bilateral relations between decisions of four above-mentioned decision points were defined, namely influencing and determining relation. The concrete relations of each decision with other decisions were also described in[Daw14].

#### CloudDSF Prototype

The CloudDSF Prototype was developed as a web application with standard web technologies (HTML, CSS, Scalable Vector Graphics SVG, JavaScript) and available at http://www.clouddsf.com. Javascript Object Notation (JSON)[3] was used for encoding the CloudDSF Knowledge Base, Data-Driven-Document (D3)[4] and jQuery[5] libraries for the visualization.

The prototype implements five types of visualizations which are divided into two groups: network layouts (including network layout and cluster layout) and hierarchy layouts (including tree layout, treemap layout and partition layout) Figure 2.3 visualizes the relationships between tasks and decisions of the CloudDSF in force layout.

The CloudDSF prototype aims at providing decision makers with a platform-independent and publicly accessible visualization of CloudDSF knowledge base.

## 2.4 CloudDSF+

The CloudDSF+[Bal15] is the refinement and extension of CloudDSF[Daw14]. The related work includes:

[3]http://json.org
[4]http://d3js.org
[5]http://jqurey.com


**Figure 2.3:** Prototypical Implementation - Force Layout[Daw14]

- Refinement of the CloudDSF knowledge base
- Extension of the CloudDSF with relations between outcomes
- Implementation of CloudDSF+ prototype

### Refinement of the CloudDSF knowledge base

After reviewing and evaluating the decision points together with their decisions, the following changes were made to the previous knowledge base:

- The decision point Application Distribution was renamed to Define Application Distribution.
- Outcomes of several decisions (e.g. Section Application Layer, Select Application Tier, Define Scalability Level) were refined by listing all possible combinations between possible outcome values.
- New outcomes were added into outcome lists of decisions Select Automation Degree, Select Scaling Trigger and Select Cloud Service Model.
- Several outcomes of decision Define Cloud Hosting were renamed.
- Significant changes regarding outcomes were made to decisions Select Application Components, Define Scalability Level and Define Roles of Responsibility.
- Some outcomes of the decision Select Pricing Model were removed.
- The single outcome of decision Define Resource Location was replaced by two new outcomes.
- The complete decision Select Kind of Multi-tenancy of decision point Define Multi-tenancy Requirements was removed.

Further changes were made to relations between decisions as well. The determining relation no longer exists after the review process. The influencing relation can exist between any two decisions and consistently refers no transitivity characteristic (A influences B and B influences C do not infer that A influences C). The uni-lateral binding relation was added, which exists only from decision Select Cloud Vendor toward other decisions. On the other hand, the affecting relation is also uni-lateral and points from other decisions toward Select Cloud Vendor. The requiring relation can exist between any two decisions. It denotes that as soon as a decision in requiring relation with another decision is selected, the latter should as well be selected.

The updated and changed knowledge base is shown in Table 2.3 for each decision point:

**Table 2.3:** The refined CloudDSF Knowledge Base[Bal15]

| **Decision Point - Define Application Distribution** |
| --- |

| | |
|---|---|
| Select Application Layer | Presentation Layer Layer<br>Business Layer<br>Resource Layer<br>Presentation + Business Layer<br>Presentation + Resource Layer<br>Business + Resource Layer<br>Presentation + Business + Resource Layer |
| Select Application Tier | Client Tier<br>Application Tier<br>Data Tier<br>Client + Application Tier<br>Client + Data Tier<br>Application + Data Tier<br>Client + Application + Data Tier |
| Select Application Components | Application Component<br>Application Components<br>Middleware Component<br>Middleware Components<br>Application + Middleware Component<br>Application Component + Middleware Components<br>Middleware Component + Application Components<br>Application + Middleware Components |
| Select Migration Type | Migration Type I<br>Migration Type II<br>Migration Type III<br>Migration Type IV |
| **Decision Point - Define Elasticity Strategy** | |
| Define Scalability Level | No Scaling<br>VM Level Scaling<br>Middleware Level Scaling<br>Application Level Scaling<br>VM + Middleware Level Scaling<br>VM + Application Level Scaling<br>Middleware + Application Level Scaling<br>VM + Middleware + Application Level Scaling |

| | |
|---|---|
| Select Scaling Type | Vertical Scaling |
| | Horizontal Scaling |
| | Hybrid Scaling |
| Select Elasticity Automation Degree | Manual Scaling |
| | Semi-Automatic Scaling |
| | Semi-Automatic Third-Party Scaling |
| | Automatic Scaling |
| | Automatic Third-Party Scaling |
| Select Scaling Trigger | No Trigger |
| | Event-Driven Trigger |
| | Proactive Trigger |
| **Decision Point - Define Multi-Tenancy Requirements** | |
| Select Multi-Tenancy Level | Shared Hardware Multi-Tenancy |
| | Shared OS Multi-Tenancy |
| | Shared Middleware Multi-Tenancy |
| | Shared Application Multi-Tenancy |
| **Decision Point - Select Service Provider / Offering** | |
| Select Cloud Deployment Model | Public Cloud |
| | Private Cloud |
| | Community Cloud |
| | Hybrid Cloud |
| Select Cloud Service Model | IaaS |
| | PaaS |
| | SaaS |
| | IaaS + PaaS |
| | IaaS + SaaS |
| | PaaS + SaaS |
| | IaaS + PaaS + SaaS |
| Define Cloud Hosting | On-Premise Hosting |
| | Off-Premise Hosting |
| | Hybrid Hosting |
| Define Roles of Responsibility | Inhouse |
| | Management |
| | Outsourced |
| | Inhouse + Management |
| | Inhouse + Outsourced |
| | Management + Outsourced |
| | Inhouse + Management + Outsourced |

| | |
|---|---|
| Select Cloud Vendor | Evaluated Cloud Vendor |
| Select Pricing Model | Free<br>Pay-Per-Use<br>Pay-Per-Unit<br>Charge-Per-Use (Subscription) |
| Define Resource Location | Data In Same Jurisdiction<br>Data In Different Jurisdiction |

### Extension of the CloudDSF with relations between outcomes

In order to further support the cloud decision making process, relationship types between outcomes were defined as seen in Table 2.4. However, only combinations of outcomes whose respective decisions are related would be under consideration. Several assumptions were made before the relations from each outcome of a decision towards all other relevant outcomes are defined. Firstly, it's possible to select only one exclusive outcome within each decision. Secondly, several decisions consist of basic outcomes and outcomes which are combination of basic outcomes of the decision. Generally the combinatorial outcome inherits the relations of the basic outcomes. In cases that contributing basic outcomes own relations that contradict each other toward another outcome, the prevailing relation will be determined depending on specific situation.

It is necessary to mention that the decision Select Cloud Vendor has no outcomes in [Daw14], because specific vendors have not been identified. As a result, the binding and affecting relations remains general at decision level and they need to be specialized at outcome level according to the specific vendors considered.

Although CloudDSF+ has limited ability to depict all influences in a large migration project, it has significantly increased expressiveness and more granular view of the relations with respect to CloudDSF.

### Implementation of CloudDSF+ prototype

Visualization of the extended knowledge base in CloudDSF+ is either static or dynamic. The static view was already partly implemented by CloudDSF[Daw14], which has the ability to show the hierarchical nature of the knowledge base and the relations between decisions and tasks. In CloudDSF+, the static visualization was expanded to include

**Table 2.4:** Definition of relationship types between outcomes[Bal15]

| Relationship Type | Abbrev. | Definition |
|---|---|---|
| Allowing | a | An allowing relation between outcome A and outcome B denotes that in case A is selected, B can be selected as well. Consequently, A neither entails nor prohibits B. |
| Excluding | ex | An excluding relation between outcome A and outcome B denotes that if A is selected, B can no longer be selected anymore. Hence, A prohibits B. |
| Including | in | An including relation between outcome A and outcome B denotes that if A is selected, B becomes obligatory and has to be selected as well. Hence, A entails B. |
| Affecting | aff | An affecting relation can only exist from outcomes of any decision towards the Evaluated Cloud Vendor outcome of the SelectCloud Vendor decision. It denotes that the participating outcome imposes certain requirements upon the cloud vendor, hence affecting the selection of the vendor per se. |
| (Externally) Binding | eb | An (externally) binding relation can only exist from the outcome Evaluated Cloud Vendor of the Select Cloud Vendor decision towards any other decision's outcomes. It denotes that the variable participating outcome is subject to the cloud vendor regarding the support of the desired feature and the actual implementation. Thus, it is externally bounded and out of the influence of the migrator. |

three new relations between decisions (binding, affecting and requiring relation) as well as relationship types between outcomes. Figure 2.4 shows the relations between outcomes without displaying decisions and decision points.

The dynamic interactive visualization is added by CloudDSF+. It enables the navigation throughout the knowledge base, i.e. depicting the impact of chosen outcomes towards the other decisions' outcomes and informing the user about possible conflict. Figure 2.5 shows part of the knowlege base navigator with relations to all specified outcomes and a conflict at the On-Premise Hosting outcome.

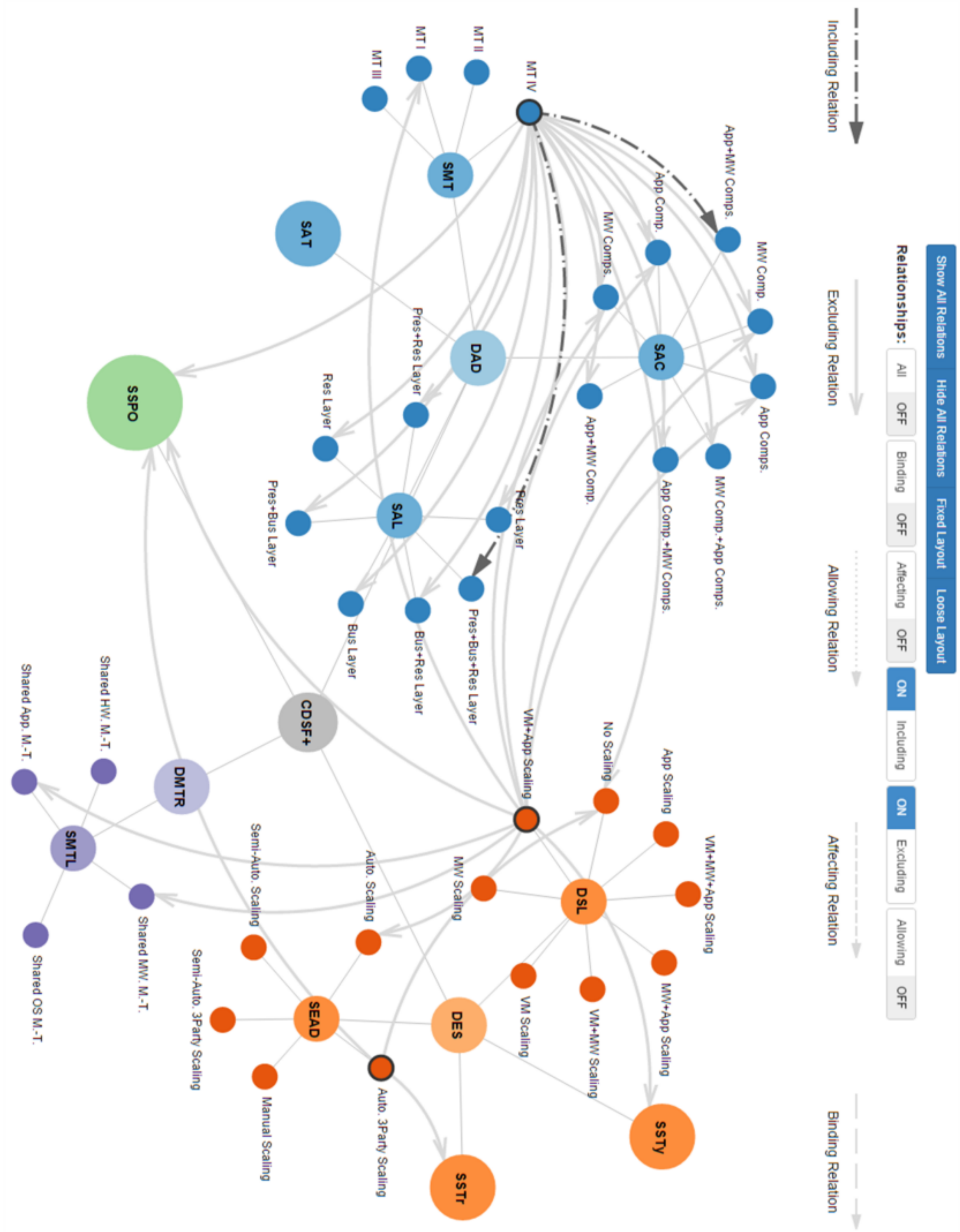


**Figure 2.4:** Outcome relations layout with collapsed decisions and collapsed Select Service Provider/Offering decision point[Bal15]

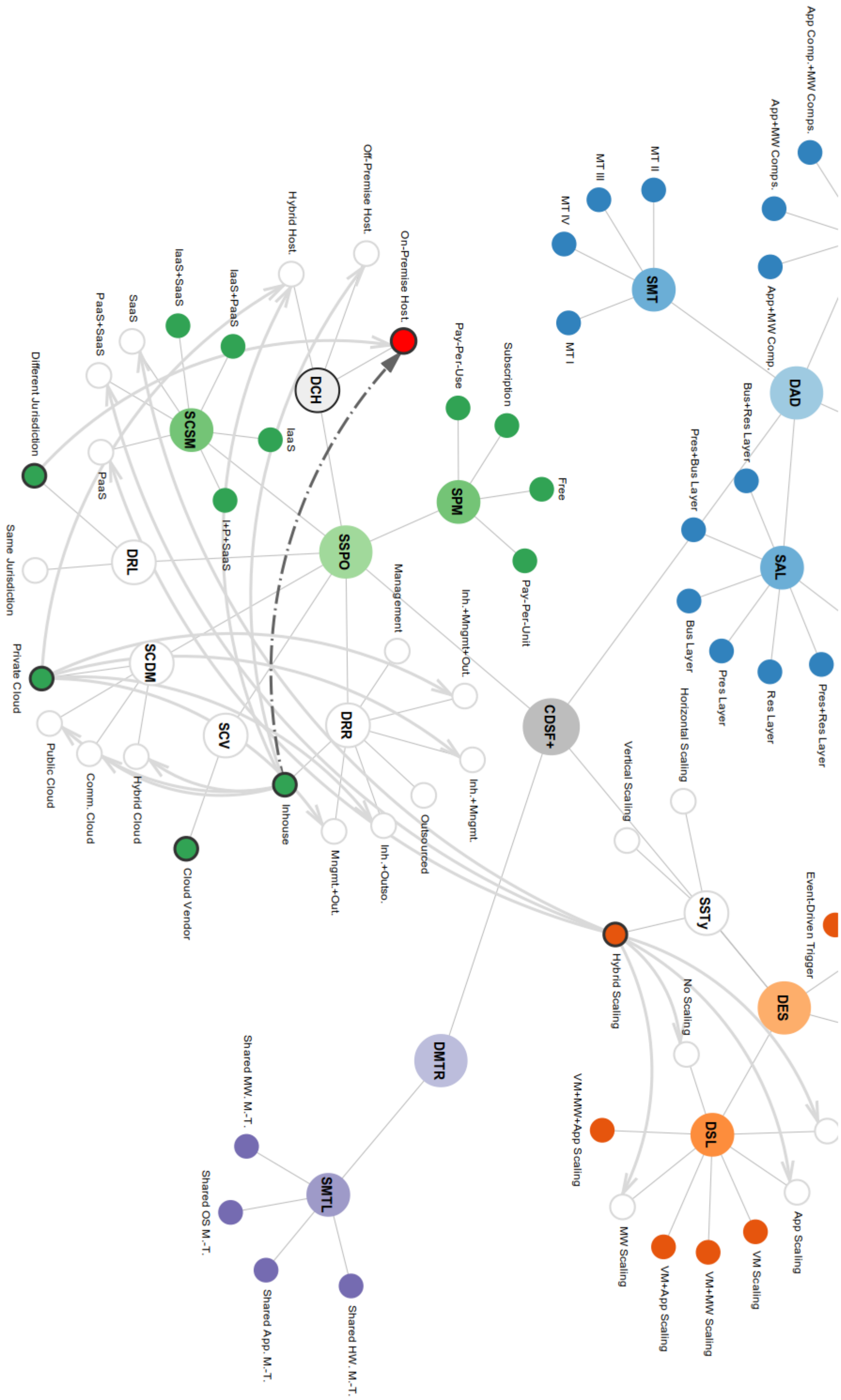


**Figure 2.5:** Part of the KBNavigator showing the relations to all specified outcomes and a conflict at the On-Premise Hosting outcome[Bal15]

In order to support the composition of input files for existing CloudDSF as well as CloudDSF+ prototype, the extended knowledge base is now rendered as an excel file with different sheets:

- Decision Level: presents all relations between decisions. The direction of relation is determined through left-to-top reading order of the sheet.
- Required level: presents only requiring relations between decisions
- Outcome level: contains all relations between decision outcomes.
- Task level: includes the CloudDSF task and their relations towards decisions. This sheet is not updated, therefor depicts the status of CloudDSF but not CloudDSF+

A parser with the capability to automatically convert an excel table into a JSON file was then additionally developed in Java programming language and made publicly available[6] under the Apache 2.0 license[7].

The source code of CloudDSF+ prototype is publicly available[8] at GitHub under the Apache 2.0 license.

[6]Source code of the CloudDSF+ parser: https://github.com/bametz/clouddsfPlusParser
[7]http://www.apache.org/licenses/LICENSE-2.0
[8]Source code of the CloudDSF+ prototype: https://github.com/bametz/clouddsfPlus

# 3 Extension of the CloudDSF+ Knowledge Base

In this chapter the specific outcomes of the decision Select Cloud Vendor, which are cloud platforms and their belonging services, are selected and briefly introduced. Then the CloudDSF+ knowledge base is extended with the relations between these services and the existing outcomes. During the course of the extension, new relationship types are also defined.

## 3.1 The Decision Select Cloud Vendor

The decision Select Cloud Vendor belongs to the decision point Select Service Provider/Offering. After CloudDSF was reviewed and updated to CloudDSF+, this decision still owned no specific outcomes. A general outcome named Evaluated Cloud Vendor was therefore added as the place-holder for the specific vendors that would be inspected in the future (see the most right-hand column of Figure 3.1).

As mentioned in section 2.4, the knowledge base of the CloudDSF+ is presented as a JSON file, which is automatically created by parsing an excel file, which serves as a comfortable tool to collect and organize information and includes among others the spreadsheets with information at the decision level and outcome level. At the decision level, a unilateral binding relation was defined as the relation from decision Select Cloud Vendor toward other decisions. On the other hand, the affecting relation is also unilateral and points from other decisions toward Select Cloud Vendor. These two relations serve as place-holders for future specialized relations at the outcome level, as long as specific vendors are identified.

As a result, the main task of this thesis is to complete the knowledge base of the CloudDSF+ by adding specific cloud vendor offerings as outcomes of the decision Select Cloud Vendor, to determine the relations among new outcomes with existing outcomes and to visualize those newly added relations.

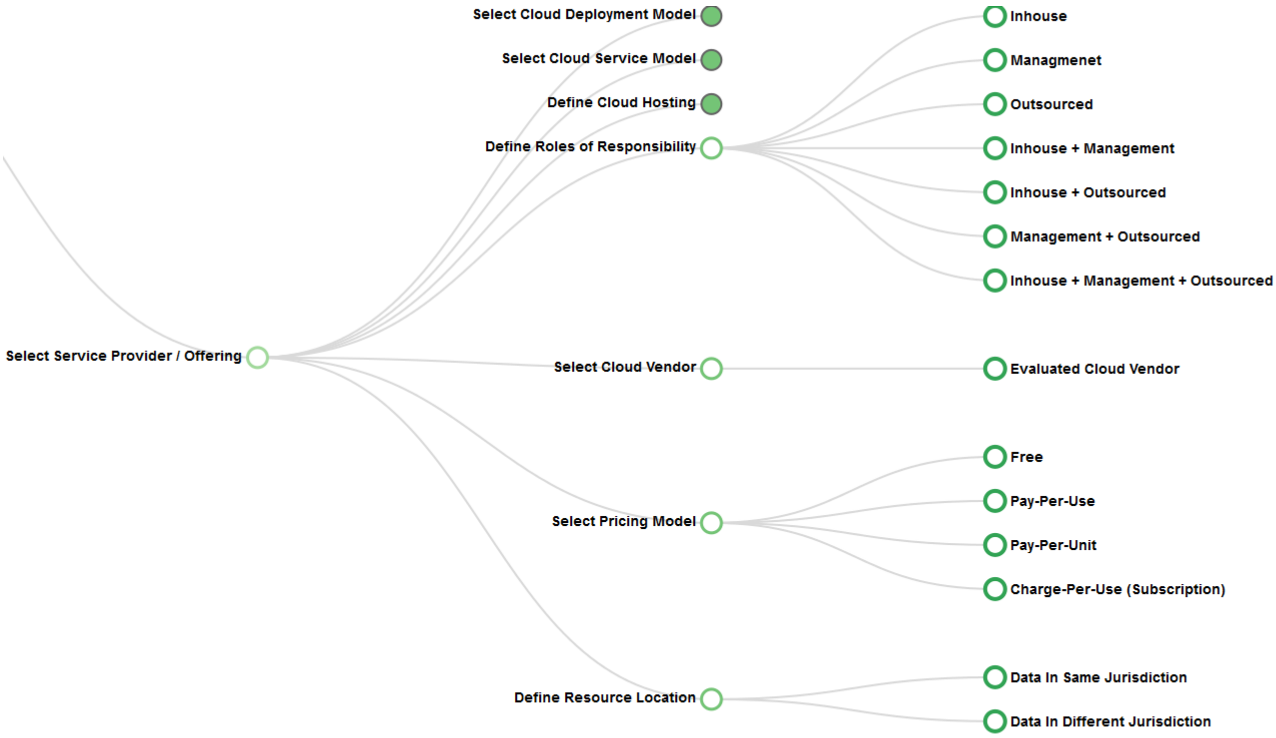


**Figure 3.1:** Hierarchical layout of the CloudDSF+ Prototype visualizing part of the knowledge base

## 3.2 Cloud Vendors in the Knowledge Base

Three major cloud vendors namely Amazon.com Inc., Microsoft Corporation[1] and Google Inc.[2] provide cloud platforms which are respectively named as Amazon Web Services, Microsoft Azure and Google Cloud Platform. This section introduces briefly each platform together with their typical belonging services. The relations between those selected services with other outcomes of the existing decisions in the knowledge base are discussed in sections 3.4 and 3.5.

There a plenty of cloud services on the cloud market, but Amazon Web Services is always on top of the list because it was early on the marketplace and also takes the biggest share in public cloud. According to statistics disclosed by RightScale Inc. in [Rig14], in the year 2014, 54 procent of enterprises participating in the survey were running their applications on AWS. In 2015 this number increased to 57 procent and continued to be

[1] https://www.microsoft.com/de-de/
[2] https://cloud.google.com/

more than four times the cloud adoption rate of the next competitor, as shown in Figure 3.2.

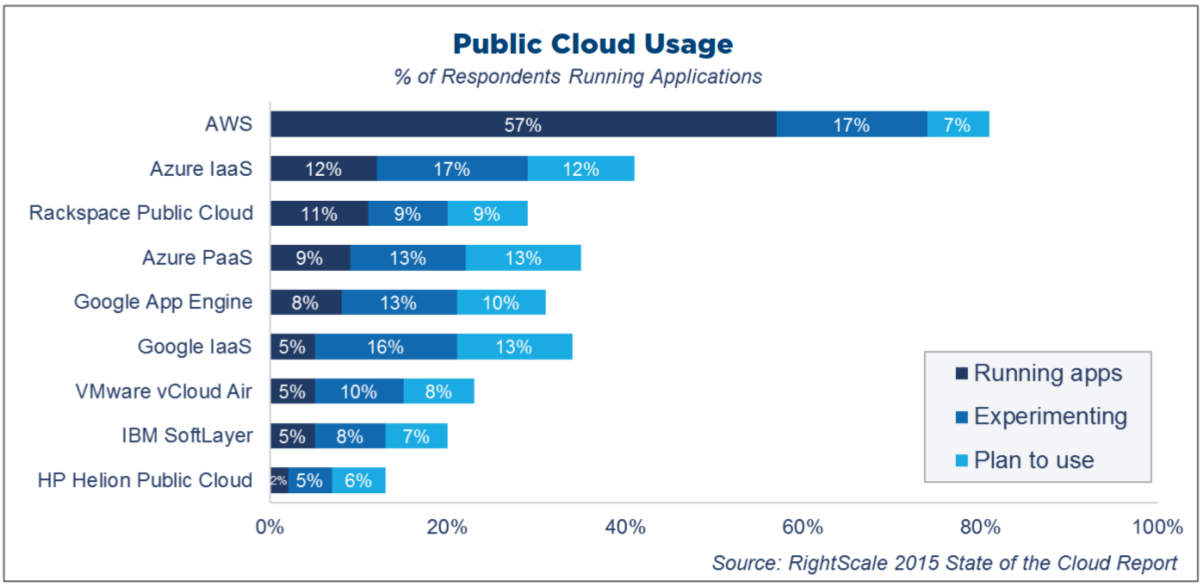


**Figure 3.2:** Public Cloud Usage 2015 [Rig14]

Microsoft Azure and Google Cloud Platform are promisingly raising their market share on the cloud market. Azure IaaS moved into the second position in public cloud segment by doubling from 6 percent in 2014 to 12 percent in 2015. At the same time, 29 procent of survey respondents were experimenting with the IaaS offerings from Azure and Google or planning to use these clouds. In 2015, Azure IaaS also gained significant increase in providing services for enterprises with more than one thousand employees (see Figure 3.3).

It is obvious that the three above-mentioned cloud providers are pioneers in cloud computing and their offerings are the most accepted products in the market. As a result, these three cloud platforms are taken into consideration for the update of the knowledge base of the CloudDSF+ and some of their typical services are defined as outcomes of the decision Select Cloud Vendor. More specifically:

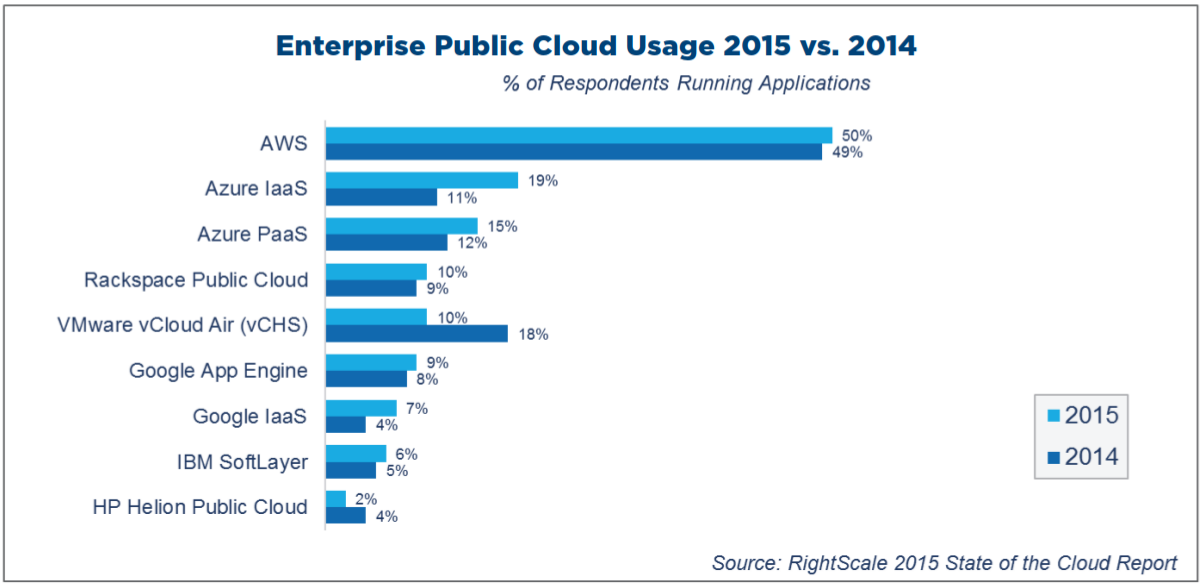


**Figure 3.3:** Enterprise Public Cloud Usage 2015 vs. 2014 [Rig14]

### 3.2.1 Amazon Web Services (AWS)

AWS is a platform which offers a bundle of cloud computing products and services which support the development and deployment of applications on the cloud. Once demanded, these services are available in a few seconds and will be charged against actual usage.

More than seventy AWS services[3] are divided into different groups:

- **Compute**: Amazon EC2, Amazon EC2 Container Registry, Amazon EC2 Container Service, AWS Elastic Beanstalk, AWS Lambda, Auto Scaling, Elastic Load Balancing, Amazon VPC
- **Storage and Content Delivery**: Amazon S3, Amazon CloudFront, Amazon EBS, Amazon Elastic File System, Amazon Glacier, AWS Import/Export Snowball, AWS Storage Gateway
- **Database**: Amazon RDS, AWS Database Migration Service, Amazon DynamoDB, Amazon ElastiCache, Amazon Redshift

[3] https://aws.amazon.com/

- **Network**: Amazon VPC, AWS Direct Connect, Elastic Load Balancing, Amazon Route 53
- **Developer Tools**: AWS CodeCommit, AWS CodeDeploy, AWS CodePipeline, AWS-Befehlszeilen-Tool
- **Management Tools**: Amazon CloudWatch, AWS CloudFormation, AWS CloudTrail, AWS-Befehlszeilen-Tool, AWS Config, AWS Management Control, AWS OpsWork, AWS Service Catalog, AWS Application Discovery Service, AWS Trusted Advisor
- **Security and Identity**: AWS Identity and Access Management (IAM), AWS Certificate Manager, AWS CloudHSM, AWS Directory Service, Amazon Inspector, AWS Key Management Service, AWS WAF
- **Analytics**: Amazon EMR, AWS Data Pipeline, Amazon Elasticsearch Service, Amazon Kinesis, Amazon Machine Learning, Amazon QuickSight, Amazon Redshift
- **Mobile Services**: AWS Mobile Hub, Amazon API Gateway, Amazon Cognito, AWS Device Farm, Amazon Mobile, Amazon Analytics, AWS SDK for mobile devices, Amazon Simple Notification Service (Amazon SNS)
- **Application Services**: Amazon API Gateway, Amazon AppStream, Amazon CloudSearch, Amazon Elastic Transcoder, Amazon FDS, Amazon SES, Amazon SNS, Amazon SQS, Amazon SWF
- **Enterprise Applications**: Amazon WorkSpaces, Amazon WorkDocs, Amazon WorkMail
- **Development of games**: Amazon Lumberyard

Among those listed services, five popular services are chosen for the CloudDSF+2 knowledge base, namely Amazon Elastic Compute Cloud (Amazon EC2), Amazon EC2 Container Service (Amazon ECS), AWS Elastic Beanstalk (Amazon ESB), Amazon Virtual Private Cloud (Amazon VPC) and Amazon Simple Storage Service (Amazon S3).

#### Amazon Elastic Compute Cloud (Amazon EC2)

Amazon EC2[4] is a web service which is used to create, launch and terminate scalable virtual machines, to configure security and networking and to manage storage in the AWS cloud. Amazon EC2 defines and provides the following features:

- Instances: virtual computing environments;

[4]http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/concepts.html

- Amazon Machine Images (AMIs): preconfigured templates for instances;
- Secure login information for instances using key pairs;
- Instance types: configurations for CPU, memory, storage and networking capacity for instances;
- Instance store volumes: storage volumes for temporary data that is deleted when an instance is stopped or terminated;
- Amazon EBS volumes: persistent storage volumes for data using Amazon Elastic Block Store (Amazon EBS);
- Availability Zones: physical location for resources like instances and Amazon EBS volumes. Users are able to control over the geographical location of instances that allows for latency optimization and high levels of redundancy;
- Security groups: protocols, ports and source IP ranges to reach instances;
- Elastic IP addresses: static IP addresses for dynamic cloud computing;
- Tags: metadata that can be created and assigned to Amazon EC2 resources;
- Virtual private clouds: can be created and optionally connected to an existing cloud;

With such functionalities, Amazon EC2 forms a central part of the AWS platform.

#### Amazon EC2 Container Service (Amazon ECS)

Amazon EC2 container service[5] is a management service that is used to run, stop and manage Docker[6] containers on a cluster, a logical grouping of Amazon EC2 instances.

A Docker container is a standardized unit of software development, containing all application running requirements such as code, runtime, system tools, system libraries, etc. Containers are created from a template called an image, which is typically built from a Dockerfile that specifies all the components included in the container. These images are then stored in a registry from which they can be downloaded and run on the container instances.

Only applications that are architected to run in container can be deployed on Amazon ECS. To prepare for an application running, a task definition is created, which describes

[5] http://docs.aws.amazon.com/AmazonECS/latest/developerguide/Welcome.html
[6] https://www.docker.com

one or more containers that form the application. A specific task is instantiated from the task definition. Once a task definition is created, it is possible to specify the number of tasks that will run on a cluster. In each container resides a container agent who sends information about the instance's current running tasks and resource utilization to Amazon ECS, and starts and stops tasks whenever it receives a request from Amazon ECS.

Beside the ability to launch and stop container-based applications with simple API calls and to get the state of a cluster, Amazon ECS allows users to access many Amazon EC2 familiar features. With Amazon ECS, it is neither necessary to operate any other cluster or configuration management systems nor to worry about scaling the management infrastructure.

### AWS Elastic Beanstalk

AWS Elastic Beanstalk[7] is used to quickly deploy and manage applications in the AWS cloud without caring about the infrastructure that runs those applications.

First of all, the developed application together with its profile is uploaded in the form of an application bundle to Elastic Beanstalk. Elastic Beanstalk uses services that are available in the AWS Free Usage Tier to automatically launch an environment and then create and configure the AWS resources needed to run the application. After the created environment is launched, it is possible for users to manage the environment and perform most deployment tasks, such as changing the size of Amazon EC2 instances or monitoring the application or deploying new application versions. The information about the deployed application, including metrics, events and environment status, is available through the AWS Management Console, APIs or Command Line Interfaces. Elastic Beanstalk also allows to choose among variants of persistent storage and database service options.

### Amazon Virtual Private Cloud (Amazon VPC)

Amazon Virtual Private Cloud[8] enables the launch of AWS resources into a virtual network defined by the user. This network adds the benefit of using the scalable infrastructure of AWS to a traditional network operating in the data center.

[7] http://docs.aws.amazon.com/elasticbeanstalk/latest/dg/Welcome.html
[8] http://awsdocs.s3.amazonaws.com/VPC/latest/vpc-ug.pdf

Amazon VPC is supported by different AWS services, such as Amazon EC2, Amazon ElastiCache, Amazon Elastic MapReduce (Amazon EMR), Amazon Relational Database Service (Amazon RDS), Amazon Redshift, Auto Scaling, AWS Data Pipeline, Elastic Beanstalk and Elastic Load Balancing.

There are several access points to Amazon VPC, either through Amazon VPC console, a web-based user interface, or through one of the available command line interfaces (CLI): AWS CLI, Amazon EC2 CLI tools, AWS tools for Windows PowerShell. Amazon VPC also provides a query API which uses HTTP requests and query parameter *Action*.

#### Amazon Simple Storage Service (Amazon S3)

Amazon Simple Storage Service[9] provides developers and IT teams scalable Web-based service of cloud storage. It is possible with Amazon S3 to upload, store and download arbitrary number of files or objects and pay for the actual volume without any minimum or set-up charge.

Amazon S3 offers three storage classes that are suitable to different requirements. Amazon S3 Standard holds generally used data with frequent access. Amazon S3 Standard - Infrequent Access (Standard-IA) is for long-term data with infrequent access. Amazon Glacier serves the long-term archiving.

There is also a configurable data life-cycle manager which automatically sends data to the appropriate storage class without making any changes to the application.

Amazon S3 can be used either separately or together with other AWS services like Amazon EC2, AWS Identity and Access Manager (IAM) or even with data migration services and gateways to take the data input.

### 3.2.2 Microsoft Azure

Microsoft Azure[10] is a cloud platform which at the moment (fourth quarter of 2016) offers fifty eight cloud services regarding different areas of cloud computing:

- **Compute**: Virtual Machines, Scaling Groups for VM, Azure Container Service, Functions, Batch, Service Factory, Cloud Services, RemoteApp

[9] https://aws.amazon.com/de/s3
[10] https://azure.microsoft.com/de-de/

- **Network**: Virtual Network, Load Balancer, Application Gateway, VPN Gateway, Azure DNS, CDN, Traffic Manager
- **Storage**: Azure Storage, Data Lake-storage, StorSimple, Backup
- **Web and mobile applications**: App Service, CDN, Media Services
- **Database**:SQL Database, SQL Data Warehouse, SQL Serve Stretch Database, DocumentDB, Table Storage, Redis Cache, Data Factory
- **Intelligence and Analysis**: HDInsight, Machine Learning, Stream Analytics, Cognitive Services, Data Lake-analysis, Data Lake-storage, Data Factory, Power BI Embedded
- **Internet of Things**: Azure IoT Hub, Event Hub, Stream Analytics, Machine Learning, Notifications Hubs
- **Enterprise Integration**: Logic-App, BizTalkService, Service Bus, API Management, StorSimple, SQL Server Stretch-Database, Data Factory
- **Security and Identity**: Security Center, Key Vault, Azure Active Directory, Azure Active Directory B2C, Azure Active Directory - Domain Service, Multi-Factor Authentication
- **Development tools**: Visual Studio Team Services, Azure Dev/Test Lab, Visual Studio Application Insights, API Management, HockeyApp, Developertools and SKDs
- **Monitoring and Management**: Microsoft Azure-Portal, Azure Resource Manager, Visual Studio Application Insights, Log Analytics, Automation, Backup, Site Recovery, Scheduler

Out of those offered services, five are selected for consideration within the CloudDSF+2 knowledge base, namely Azure Virtual Machine, Azure Container Service, Azure App Service, Azure Virtual Network and Azure Blob Storage.

#### Azure Virtual Machines

The Azure Virtual Machines[11] service enables the creation, launch and termination of virtual machines on Microsoft Azure cloud. It supports Linux[12], Windows Server[13], SQL

[11] https://azure.microsoft.com/de-de/services/virtual-machines/
[12] https://www.linux.com/
[13] https://www.microsoft.com/de-de/server-cloud/products/windows-server-2016/default.aspx

Server[14], Oracle[15], IBM[16], SAP[17] and therefore enables flexible virtualization for software development and test, for application execution or for expanding a computer center.

Azure Virtual Machines offers various options regarding operating system and location of virtual machines. It is possible either to specify an image or to download pre-configured images from the marketplace. Azure Virtual Machines provides scalability and security of virtual machines created whose price is calculated based on time of usage.

#### Azure Container Service

Azure Container Service (ACS) is used to create, configure and manage a cluster of virtual computers which is preconfigured for execution of applications in containers.

Azure Container Service uses Docker[18] container format to make sure that the application containers are portable. It supports additionally Marathon[19] orchestration platform and DC/OS[20] or Docker Swarm[21], in order to scale applications up to thousand or ten thousands of containers.

In order to communicate with those services, Azure Container Service uses the standard API endpoints for Docker, DCOS Command Line Interface (DCOS-CLI) for DC/OS and Docker-CLI for Docker Swarm. It is also possible to use arbitrary applications that can communicate with those offered endpoints.

#### Azure App Service

Azure App Service[22] is a cloud platform focusing on providing developers with practical facilities to develop and deliver applications at the cloud scale. Following are some features and capabilities of App Service:

[14]https://www.microsoft.com/en-us/cloud-platform/sql-server
[15]https://www.oracle.com/de/index.html
[16]https://www.ibm.com/de-de/
[17]http://go.sap.com/germany/index.html
[18]https://www.docker.com/
[19]https://mesosphere.github.io/marathon/
[20]https://dcos.io/
[21]https://www.docker.com/products/docker-swarm
[22]https://azure.microsoft.com/en-us/documentation/articles/app-service-value-prop-what-is/

- Supporting multiple language and frameworks, including ASP.NET[23], Node.js[24], Java[25], PHP[26] and Python[27]. It is possible to run Windows Powershell[28] and other scrips or executables on App Service virtual machines;
- Optimizing DevOps by setting up continuous integration and deployment with Visual Studio Team Services[29], GitHub[30] or BitBucket[31], by promoting updates through test and staging environments, by using Azure Powershell or cross-platform command-line interface to manage applications;
- Offering manual or automatic global scale with high availability;
- connecting to SaaS platforms by choosing from more than fifty connectors for enterprise systems such as SAP[32], Siebel[33] and Oracle[34], for SaaS services such as Salesforce[35] and Office365[36] and for internet services such as Facebook[37] and Twitter[38]. Access to on-premise data is gained through Azure Hybrid Connections and Azure Virtual Networks.
- Supporting compliance with International Organization for Standardization[39] (ISO), Security Operations Center (SOC) and Protocol Control Information (PCI);
- Offering application templates in Azure Marketplace by allowing users to install popular open source softwares;
- integrating with Visual Studio by dedicating tools in Visual Studio streamline the work of creating, deploying and debugging;

[23]http://www.asp.net/
[24]https://nodejs.org/en/
[25]https://www.java.com/de/
[26]http://php.net/
[27]https://www.python.org/
[28]https://msdn.microsoft.com/en-us/powershell
[29]https://www.visualstudio.com/de/team-services/
[30]https://github.com/
[31]https://bitbucket.org/
[32]http://go.sap.com/germany/index.html
[33]http://www.oracle.com/us/products/applications/siebel/overview/index.html
[34]https://www.oracle.com/de/index.html
[35]https://www.salesforce.com/de/
[36]https://products.office.com/de-DE/business/get-latest-office-365-for-your-business-with-2016-apps/
[37]https://www.facebook.com/
[38]https://twitter.com
[39]www.iso.org

### Azure Virtual Network

Azure Virtual Network[40] allows to build a logical isolation of the Azure cloud dedicated to a specific subscription. It is possible to control the IP address blocks, Domain Name System (DNS) settings, security policies and route tables within the network.

Benefits of an virtual network include:

- Each virtual network is totally isolated from other ones. As a result, it is possible to create separate networks for development, testing and production that use the same Classless Inter-Domain Routing (CIDR) address blocks.
- All virtual machines and instances in a virtual network can access the public Internet by default. The access can be controlled by using network security groups.
- Access to other virtual machines within the virtual network is through the use of private IP addresses possible, even if they are in different subnets.
- Azure offers internal name resolution for virtual machines and instances deployed in virtual networks.
- High level of security
- Connection between virtual networks, with on-premise data center are possible by using Virtual Private Network (VPN) or ExpressRoute[41] connection.

### Azure Blob Storage

Azure storage provides four storage services: blob storage, table storage, queue storage and file storage. As a result, blob storage shares the durability, availability, scalability and performance features that a general purpose storage account owns.

Blob storage offers three types of blobs:

- Block blobs: optimized for streaming and storing cloud objects.
- Append Blobs: optimized for append operations. An append blob can be updated only by adding new blocks to the end.
- Page blobs: optimized for representing IaaS disks and supporting random writes.

It is possible to use blob storage to store contents such as:

[40] https://azure.microsoft.com/en-us/services/virtual-network/
[41] https://azure.microsoft.com/de-de/services/expressroute/

- Documents
- Photos, videos, music, blogs
- Backups of files, computers, databases and devices
- Configuration data for cloud applications
- Big data

Blob storage introduces two access tiers:

- Hot access tier for more frequently accessed objects;
- Cool access tier for less frequently accessed objects;

With those access tiers it is possible to minimize the storage charges according to actual demand and usage.

### 3.2.3 Google Cloud Platform

Google Cloud Platform[42] is another cloud platform beside Amazon Web Services and Microsoft Azure. It offers the following service groups:

- **Compute**: Compute Engine, App Engine, Container Engine, Container Registry, Cloud Functions;
- **Storage and Databases**: Cloud Storage, Cloud SQL, Cloud Bigtable, Cloud Datastore, Persistent Disk;
- **Networking**: Cloud Virtual Network, Cloud Load Balancing, Cloud CDN (Content Delivery Network), Cloud Interconnect, Cloud DNS (Domain Name System);
- **Big Data**: BigQuery, Cloud Dataflow, Cloud Dataproc, Cloud Datalab, Cloud Pub/Sub, Genomics;
- **Machine Learning**: Cloud Machine Learning Platform, Vision API, Speech API, Natural Language API, Translate API;
- **Management Tools**: Stackdriver Overview, Monitoring, Logging, Error Reporting, Trace, Debugger, Deployment Manager, Cloud Console, Cloud Shell, Cloud Mobile App, Billing API, Cloud APIs;

[42] https://cloud.google.com/

- **Developer Tools**: Cloud SDK, Deployment Manager, Cloud Resource Repositories, Cloud Endpoints, Cloud Tools for Android Studio, Cloud Tools for IntelliJ, Cloud Tools for PowerShell, Cloud Tools for Visual Studio, Google Plugin for Eclipse, Cloud Test Lab;
- **Identity and Security**: Cloud IAM (Identiy & Access Management), Cloud Resource Manager, Cloud Security Scanner, Cloud Platform Security Overview;

The five services of Google Cloud Platform that are selected to be inspected for the CloudDSF+2 knowledge base include Google Compute Engine, Google Container Engine, Google App Engine, Google Compute Engine Network and Google Cloud Storage.

### Google Compute Engine

Google Compute Engine[43] allows to create and run virtual machines on Google infrastructure. It also offers scale, performance and value that enables large compute clusters to be launched on Google infrastructure.

Compute Engine offers the following features:

- Preconfigured virtual machines for various need;
- Ability to create virtual machines with CPU and memory that is right for specific workload;
- Ability to create Hard Disk Drives (HDD) or Solid-State Drive (SSD) persistent disks and attach to virtual machines. If a VM instance terminates, its persistent disk can retain data and attach to another instance. It is possible to use snapshots of persistent disks.
- Local encrypted SSD, which is physically attached to server hosting the virtual machine instance and results in very high input/output operations per second and very low latency.
- Ability to move virtual machines, even under extreme load, to nearby hosts while underlying host machines undergo maintenance;
- Global load balancing;
- Linux and Windows support;
- Ability to run large compute and batch jobs;

[43] https://cloud.google.com/compute/

- Commitment to information security through completion of ISO 27001 (the international norm for information security management systems - ISMS), Statement on Standards for Attestation Engagements 16 (SSAE 16), Service Organization Controls 1, 2 and 3 (SOC 1, 2 and 3) certifications;
- After minimum ten-minute charge, service is billed for the actual use time;

### Google Container Engine

Google Container Engine[44] is a cluster manager and orchestration system for running Docker containers. Container Engine schedules containers in clusters and automatically manages them based on requirements defined by the users. When the requirements change, it is possible to adjust the cluster resources allocated to the containers or the size of container clusters. An IP address range is reserved for each container cluster, which allows cluster IPs to coexist with private network IPs via Google Cloud VPN.

Container Engine ensures that the created clusters are always available and up-to-date. Container Engine is used together with Google Container Registry to make it easier to store and access private Docker images. A logging service is additionally offered to gain insight how an application is running.

### Google App Engine

Google App Engine[45] is a platform for building web applications and mobile back-ends. With App Engine the users just have to upload the code and the required infrastructure will be automatically provisioned and managed. App Engine also scales applications automatically. Furthermore, App Engine provides following built-in services:

- User authentication with Google Accounts;
- NoSQL data store with scalable storage, a rich data modeling API and an SQL-like query language;
- Google Cloud SQL: a web service for creating, configuring and using relational databases that live in Google cloud;
- Memcache: a distributed, in-memory data cache;

[44]https://cloud.google.com/container-engine/
[45]https://cloud.google.com/appengine/

- Security scanner: works automatically to detect security vulnerabilities;
- Search: performs Google-like searches;
- Traffic splitting: routes incoming requests to different app versions;
- Logging: provides access to application and request logs;
- Task queues;

#### Google Compute Engine Network

Google Compute Engine Network[46] handles communication between virtual machine instances and between instances with other networks. Each instance has an internal IP address and an external IP address which is routable over the Internet.

Compute Engine networks support only IPv4 unicast traffic, but not IPv4 broadcast or IPv4 multicast. Compute Engine networks do not support IPv6.

There are two types of Compute Engine network, namely legacy network and subnet network. A legacy Compute Engine network has a single network prefix range and a single gateway IP address for the whole network. The network is global in scope and spans all cloud regions. A subnet Compute Engine network divides the network into regional subnets, each with its own prefix. Subnet networks are regional in scope.

A Compute Engine Network can contain maximal seven thousand virtual machine instances. It is possible to create up to five networks per project, including the default network. If default network is not wanted, it is possible to delete it and create another network instead.

#### Google Cloud Storage

Google Cloud Storage[47] offers object storage in three options:

- Standard storage: optimal for data with low latency access or data that is frequently accessed. The availability of standard data is highest compared with the other two options, but storage cost is slightly higher.

[46] https://cloud.google.com/compute/docs/networking/
[47] https://cloud.google.com/storage/

- Durable Reduced Availability (DRA) storage: allows to store data at lower cost, but also lower availability than standard storage.
- Nearline storage: low-cost, highly-durable service for data archiving, online backup and disaster recovery.

## 3.3 New Relation Types

As seen in Table 2.4, the existing relationship types between outcomes of different decisions in the CloudDSF+ knowledge base include allowing, excluding, including, affecting and binding. In this phase, the two placeholder relations namely affecting and binding should be replaced by relationship types that represent the actual relation between outcomes of decision Select Cloud Vendor and outcomes of other decisions. For this purpose, three new relationship types are defined (see Table 3.1).

These newly defined relationship types were determined in a four-step process as following:

- A sample of the services offered by cloud vendors is randomly selected.
- Relations between the randomly selected services and other outcomes in the CloudDSF+2 knowledge base are under consideration. In case the existing relationship types do not cover the considered relations, new candidate relations are identified.
- All candidate relations are grouped and formalized into Table 3.1.
- The identified candidate relations are verified through their application to the list of all services selected as outcomes of the decision Select Cloud Vendor, as discussed in Sections 3.4 and 3.5.

## 3.4 Select Cloud Vendor Outcomes and Relations to Other Outcomes

Five services of each of the three cloud platforms Amazon Web Services, Microsoft Azure and Google Cloud Platform were selected as outcomes of the decision Select Cloud Vendor, making a set of fifteen outcomes which are listed and briefly introduced in the Section 3.2. This section defines and discusses the relations between those outcomes toward other outcomes in the CloudDSF+2 knowledge base. Each subsection corresponds to a decision and its outcomes across different decision points. The assumptions described in Section 2.4 about relations between outcomes remain unchanged.

**Table 3.1:** Definition of new relationship types between outcomes

| Relationship Type | Abbrev. | Definition |
|---|---|---|
| Allowing plus | ap | An extended allowing relation between outcome A and outcome B denotes that in case A is selected, another service C should be selected to support the selection of B. Consequently, A plus C allow selection of B. |
| Prerequisite Allowing | ca | A conditional allowing relation between outcome A and outcome B denotes that there must be a condition C to be fulfilled before A is selected. Consequently, C + A allow selection of B. |
| Arbitrary Allowing | aa | An arbitrary allowing relation between outcome A and outcome B denotes that if A is selected, then it is not possible to ensure the selection of B. In some cases the selection of B is possible, but not in other cases. |

### 3.4.1 Define Scalability Level

The relations between outcomes of the decision Select Cloud Vendor toward outcomes of the decision Define Scalability Level are summarized in Table 3.2.

The three services Amazon EC2, Azure Virtual Machines and Google Compute Engine share the main functionality of creating, launching and terminating virtual machines within respective Amazon Web Services, Microsoft Azure and Google Cloud Platform. When each of these three services is used independently, it depicts the no-scaling feature as default.

Scaling at the virtual machine level means that virtual machine instances are automatically added or deleted due to the requested frequency of the hosted applications. The Amazon EC2 service naturally supports no scaling, but it has the functionality of scaling at virtual machine level when working together with other services namely Amazon CloudWatch and Amazon Simple Notification Service. Similarly the Azure Virtual Machines service offers no-scaling feature, but together with Scaling Groups for VMs, which supports both Windows and Linux virtual machines, it supports virtual machine scaling. The Goolge Compute Engine service supports virtual machine scaling using the Google Autoscaler.

The three services Amazon ECS, Azure Container Service and Google Container Engine provide the container management services that make it easy to run, stop and manage Docker containers on a cluster of Amazon EC2, Azure Virtual Machines or Google

Compute Engine instances. Such container services support generally no scaling, but Amazon ECS together with CloudWatch Alarms, Azure Container Service with Scaling Groups for VMs and Google Container Engine with Cluster Autoscaler enable to scale up from one container to thousands of containers without raising the complexity of running the hosted application. As a result, the scalability is supported at the virtual machine level.

The AWS Elastic Beanstalk, Azure App Service and Google App Engine are platforms for building web applications running respectively in Amazon Web Services, Microsoft Azure and Google Cloud Platform. Users just have to upload the code of the developed web application and the respective service of each cloud platform will manage the application's availability, including provision of required infrastructure and automatic scalability. The scalability is naturally not supported, but together with other services it is provided at both virtual machine and application levels.

The virtual infrastructure in the cloud can be built and managed by Amazon Virtual Private Cloud (Amazon VPC), Azure Virtual Network or Google Compute Engine Network. Those services offer the same benefits as other services of the above-mentioned cloud platforms, including no scalability in general situation or scalability at the virtual machine level when they are in use together with other services.

The storage services Amazon S3, Azure Blob Storage and Google Cloud Storage are highly scalable. As explained in [Bal15], this type of scalability is at the middleware level.

### 3.4.2 Select Scaling Type

As seen in Table 3.3, Amazon EC2 supports horizontal scaling, which means the possibility to add or remove Amazon EC2 instances on demand. Vertical scaling is defined as the possibility to assign or remove resources to a single virtual machine [Daw14]. In this sense, Amazon EC2 supports vertical scaling, because it is possible to use Amazon Elastic Block Store service, which offers elastic storage volumes, as data carrier for Amazon EC2 instances [48]. The hybrid scaling is therefore supported.

It is obvious that Azure Virtual Machines supports horizontal scaling. Vertical scaling is also possible with virtual machines of GS series because they use Azure Storage Premium service, which delivers high-performance, low-latency disk support [49]. This service is at the moment available only in USA North 2, USA West, Canada North and Canada Center

[48]https://aws.amazon.com/de/ecs/details/
[49]https://azure.microsoft.com/de-de/services/virtual-machines/

**Table 3.2:** Relations between outcomes of the decision Select Cloud Vendor toward outcomes of the decision Define Scalability Level

| | No Scaling | VM Level Scaling | Middleware Level Scaling | Application Level Scaling | VM + Middleware Level Scaling | VM + Application Level Scaling | Middleware + Application Level Scaling | VM + Middleware + Application Level Scaling |
|---|---|---|---|---|---|---|---|---|
| **AWS services** | **Define Scalability Level** | | | | | | | |
| Amazon EC2 | a | ap | ex | ex | ex | ex | ex | ex |
| Amazon ECS | a | ap | ex | ex | ex | ex | ex | ex |
| AWS Elastic Beanstalk | a | ap | ex | ap | ex | ex | ex | ex |
| Amazon VPC | a | ap | ex | ex | ex | ex | ex | ex |
| Amazon S3 | ex | ex | a | ex | ex | ex | ex | ex |
| **Azure services** | | | | | | | | |
| Azure Virtual Machines | a | ap | ex | ex | ex | ex | ex | ex |
| Azure Container Service | a | ap | ex | ex | ex | ex | ex | ex |
| Azure App Service | a | ap | ex | ap | ex | ex | ex | ex |
| Azure Virtual Network | a | ap | ex | ex | ex | ex | ex | ex |
| Azure Blob Storage | ex | ex | a | ex | ex | ex | ex | ex |
| **Google services** | | | | | | | | |
| Google Compute Engine | a | ap | ex | ex | ex | ex | ex | ex |
| Google Container Engine | a | ap | ex | ex | ex | ex | ex | ex |
| Google App Engine | a | ap | ex | ap | ex | ex | ex | ex |
| Google Compute Engine Network | a | ap | ex | ex | ex | ex | ex | ex |
| Google Cloud Storage | ex | ex | a | ex | ex | ex | ex | ex |

[50]. Azure supports additionally the manual selection of virtual machines of smaller or bigger size.

Beside horizontal scaling support, Google Compute Engine offers the same competitive feature of vertical scaling by attaching up to 64 TB storage to virtual machines as persistent disk [51]. As a result, hybrid scaling is also possible.

Amazon ECS supports applications and services by using its service planner, a document that enables up- or downscaling of container clusters to fulfill the capacity requirements of an application [52]. Azure Container Service shares this feature by using DC/OS oder Docker Swarm for container scaling and orchestration. These two services support only horizontal scaling. Google Container Engine supports additionally vertical scaling by introducing the concept of node pool. A node pool is a collection of machines with the same configuration. A cluster can include multiple node pools of different configurations. For example, a cluster composed of n1-standard-2 machines needs more CPUs. It is then possible to add a node pool to the existing cluster composed of n1-standard-4 machines[53].

AWS Elastic Beanstalk and Google App Engine support the horizontal scaling in the sense that they ensure the availability of uploaded applications by adding or removing virtual machine instances due to the application capacity requirements at different points of time. Azure App Service supports additionally vertical scaling by allowing central upscaling, which brings about more CPUs, more memory and more storage space[54].

Amazon VPC, Azure Virtual Network and Google Compute Engine Network enable users to define virtual networks in their own logically isolated areas within the respective AWS, Azure or Google cloud. It is possible to launch virtual machine instances into the created virtual networks. These three services share the horizontal scaling feature, because they each defines a virtual layer over the horizontally scalable virtual machine instances.

Amazon S3 is vertically scalable, because it is always possible to change the size of storage if necessary. The horizontal scaling is supported through the replication of uploaded objects across different regions. Azure Blob Storage and Google Cloud Storage share these features and therefore support vertical, horizontal and hybrid scaling.

[50] https://azure.microsoft.com/de-de/regions/services/
[51] https://cloud.google.com/compute/
[52] https://aws.amazon.com/de/ecs/faqs/
[53] https://cloudplatform.googleblog.com/2016/05/introducing-Google-Container-Engine-GKE-node-pools.html
[54] https://azure.microsoft.com/de-de/documentation/articles/web-sites-scale/

**Table 3.3:** Relations between outcomes of the decision Select Cloud Vendor toward outcomes of the decision Select Scaling Type

| | Vertical Scaling | Horizontal Scaling | Hybrid Scaling |
|---|---|---|---|
| **Amazon services** | **Select Scaling Type** | | |
| Amazon EC2 | ap | a | a |
| Amazon ECS | ex | a | ex |
| AWS Elastic Beanstalk | ex | a | ex |
| Amazon VPC | ex | a | ex |
| Amazon S3 | a | a | a |
| **Azure services** | | | |
| Azure Virtual Machines | ap | a | a |
| Azure Container Service | ex | a | ex |
| Azure App Service | a | a | ex |
| Azure Virtual Network | ex | a | ex |
| Azure Blob Storage | a | a | a |
| **Google services** | | | |
| Google Compute Engine | a | a | a |
| Google Container Engine | a | a | ex |
| Google App Engine | ex | a | ex |
| Google Compute Engine Network | ex | a | ex |
| Google Cloud Storage | a | a | a |

### 3.4.3 Select Elasticity Automation Degree

As seen in Table 3.4, Amazon EC2 offers the possibility to manually scale the instance group up or down using the AWS console or AWS CLI[55]. The semi-automatic scaling type is supported in the way that a scaling schedule is created and Auto Scaling service will perform scaling actions at the specified time. Automatic scaling is only possible with Auto Scaling service, Amazon CloudWatch service, Amazon Simple Notification Service (SNS) and a policy which tells Auto Scaling how to respond to alarm messages. Azure Virtual

[55]http://docs.aws.amazon.com/autoscaling/latest/userguide/as-manual-scaling.html

Machines is equally competitive by offering manual scaling through Azure portal or Azure classic portal, scheduled scaling and automatic scaling based on different metrics and alarms[56]. Google Compute Engine supports the manual scaling, semi-automatic and automatic scaling of managed instance groups. Unmanaged instance groups are not provided with those scaling services.

Previously, when an application experienced a load spike, Amazon ECS users had to manually scale the number of tasks in the Amzon ECS service. From May 2016, Amazon ECS supports automatic scaling of container-based applications by dynamically growing or shrinking the number of tasks run by Amazon ECS[57]. Azure Container Service supports the manual scaling of container clusters with Docker Swarm or DC/OS. Google Container Engine can automatically scale up and down the hosted applications based on resource utilization.

AWS Elastic Beanstalk allows to change the size of Amazon EC2 instance fleet directly from the Elastic Beanstalk web interface. It supports additionally the automatic scaling by using Auto Scaling service and Amazon CloudWatch alarms[58]. Scaling an application of Azure App Service up or down can be done manually or automatically[59]. Google App Engine offers built-in automatic scaling of applications based on need[60].

Amazon VPC, Azure Virtual Network and Google Compute Engine Network are the networking layers for respective Amazon EC2, Azure Virtual Machines and Google Compute Engine. As a result, they support automatic scaling of virtual machine instances in the network.

With Amazon S3, Azure Blob Storage and Google Cloud Storage, users can store as much data as desired. The storage size can be automatically adjusted due to user's demand pattern.

### 3.4.4 Select Scaling Trigger

Table 3.5 lists all types of triggers that can be activated to signal an automatic scaling. The services Amazon EC2, Amazon ECS, AWS Elastic Beanstalk, Azure Virtual Machines, Azure App Service, Google Compute Engine and Google Container Engine support both manual and automatic scaling. As a result, these services own the conditional allowing

[56]https://azure.microsoft.com/de-de/documentation/articles/monitoring-overview-autoscale/
[57]https://aws.amazon.com/de/about-aws/whats-new/2016/05/amazon-ec2-container-service-supports-automatic-service-scaling/
[58]http://docs.aws.amazon.com/elasticbeanstalk/latest/dg/Welcome.html
[59]https://azure.microsoft.com/de-de/documentation/articles/insights-how-to-scale/
[60]https://cloud.google.com/appengine/

**Table 3.4:** Relations between outcomes of the decision Select Cloud Vendor toward outcomes of the decision Select Elasticity Automation Degree

| | Manual Scaling | Semi-Automatic Scaling | Semi-Automatic Third-Party Scaling | Automatic Scaling | Automatic Third-Party Scaling |
|---|---|---|---|---|---|
| **Amazon services** | **Select Elasticity Automation Degree** | | | | |
| Amazon EC2 | a | a | ex | ap | ex |
| Amazon ECS | a | ex | ex | a | ex |
| AWS Elastic Beanstalk | a | ex | ex | a | ex |
| Amazon VPC | ex | ex | ex | a | ex |
| Amazon S3 | ex | ex | ex | a | ex |
| **Azure services** | | | | | |
| Azure Virtual Machines | a | a | ex | a | ex |
| Azure Container Service | a | ex | ex | a | ex |
| Azure App Service | a | ex | ex | a | ex |
| Azure Virtual Network | ex | ex | ex | a | ex |
| Azure Blob Storage | ex | ex | ex | a | ex |
| **Google services** | | | | | |
| Google Compute Engine | ca | ca | ex | ca | ex |
| Google Container Engine | a | ex | ex | a | ex |
| Google App Engine | ex | ex | ex | a | ex |
| Google Compute Engine Network | ex | ex | ex | a | ex |
| Google Cloud Storage | ex | ex | ex | a | ex |

relation toward the outcome No Trigger. Condition of the relation is the selection of manual scaling.

The extended allowing relation between Amazon ECS and AWS Elastic Beanstalk toward the outcome Event-Driven Trigger specifies that these two services works together with Amazon CloudWatch and the trigger is activated due to results of Amazon CloudWatch alarms.

**Table 3.5:** Relations between outcomes of the decision Select Cloud Vendor toward outcomes of decision Select Scaling Trigger

| | No Trigger | Event-Driven Trigger | Proactive Trigger |
|---|---|---|---|
| **Amazon services** | **Select Scaling Trigger** | | |
| Amazon EC2 | ca | a | ex |
| Amazon ECS | ca | ap | ex |
| AWS Elastic Beanstalk | ca | ap | ex |
| Amazon VPC | ex | a | ex |
| Amazon S3 | a | ex | ex |
| **Azure services** | | | |
| Azure Virtual Machines | ca | a | ex |
| Azure Container Service | a | ex | ex |
| Azure App Service | ca | a | ex |
| Azure Virtual Network | ex | a | ex |
| Azure Blob Storage | a | ex | ex |
| **Google services** | | | |
| Google Compute Engine | ca | a | ex |
| Google Container Engine | ca | a | ex |
| Google App Engine | ex | a | ex |
| Google Compute Engine Network | ex | a | ex |
| Google Cloud Storage | a | ex | ex |

### 3.4.5 Select Multi-Tenancy Level

As seen in Table 3.6, all three cloud platforms Amazon Web Services (AWS), Microsoft Azure and Google Cloud Platform use hypervisors (Xen, Hyper-V and KVM) to provision their virtual machines. As a result, shared hardware multi-tenancy is applied, which means different tenants use the same physical machines on which their applications are running, but for each tenant a separate virtual machine is running. This process occurs in the hypervisor layer through visualization.

It is obvious that the virtual machine services of these three platforms, namely Amazon EC2, Azure Virtual Machines and Google Compute Engine share the feature of multi-tenancy at the hardware level.

The container services (Amazon ECS, Azure Container Service, Google Container Engine) actually build container layers over virtual machine instances and logically own the shared hardware multi-tenancy. The Amazon ECS service is furthermore combined with Amazon EC2 Container Registry (Amazon ECR), which enables the storage, execution and management of the images executed in Amazon ECS. As a result, the same image can be pulled out of the registry and shared between containers of different applications, which constitutes the shared operating system multi-tenancy. The Google Container Engine service is equally competitive with Google Container Registry.

The PaaS services AWS Elastic Beanstalk, Azure App Service and Google App Engine are responsible for deploying applications on respective AWS, Azure and Google clouds, including the provision of required infrastructure. As a result, they share the hardware multi-tenancy feature of virtual machine services. The Google App Engine service allows additionally the shared middleware multi-tenancy by providing database server Google Cloud SQL.

The three network services Amazon VPC, Azure Virtual Network, Google Compute Engine Network build network layers over virtual machine instances and therefore own the shared hardware multi-tenancy. The storage services Amazon S3, Azure Blob Storage, Google Cloud Storage, which serve different tenants from the same shared storage infrastructure, also support shared hardware multi-tenancy.

### 3.4.6 Select Cloud Deployment Model

The deployment models include public cloud, private cloud, community cloud and hybrid cloud, as seen in Table 3.7. Amazon Web Services provide services for building

**Table 3.6:** Relations between outcomes of the decision Select Cloud Vendor toward outcomes of the decision Select Multi-Tenancy Level

| | Shared Hardware Multi-Tenancy | Shared OS Multi-Tenancy | Shared Middleware Multi-Tenancy | Shared Application Multi-Tenancy |
|---|---|---|---|---|
| **Amazon services** | **Select Multi-Tenancy Level** | | | |
| Amazon EC2 | a | ex | ex | ex |
| Amazon ECS | a | ap | ex | ex |
| AWS Elastic Beanstalk | a | ex | ex | ex |
| Amazon VPC | a | ex | ex | ex |
| Amazon S3 | a | ex | ex | ex |
| **Azure services** | | | | |
| Azure Virtual Machines | a | ex | ex | ex |
| Azure Container Service | a | ex | ex | ex |
| Azure App Service | a | ex | ex | ex |
| Azure Virtual Network | a | ex | ex | ex |
| Azure Blob Storage | a | ex | ex | ex |
| **Google services** | | | | |
| Google Compute Engine | a | ex | ex | ex |
| Google Container Engine | a | ap | ex | ex |
| Google App Engine | a | ex | a | ex |
| Google Compute Engine Network | a | ex | ex | ex |
| Google Cloud Storage | a | ex | ex | ex |

public cloud, community cloud and hybrid cloud. AWS GovCloud (US)[61] is an isolated AWS region which allows the U.S. government agencies and customers to move sensitive workloads to the cloud by addressing their specific regulatory and compliance requirements. The GovCloud region adheres to U.S. International Traffic in Arms Regulations (ITAR) and provides the possibility to build a community cloud. AWS offers furthermore integrated network-, security- and access control as well as functions supporting data integration, life cycle, resource and provision management in order to support the building of a hybrid cloud[62]. Amazon does not offer private cloud services for some reasons stated by its senior staff [63]. Firstly, in a private model, companies must still have servers and data space, which requires a fixed initial investment, but capacity is limited by the space available to the infrastructure owned. Secondly, managing a large-scale infrastructure with high availability requires a trained staff as well as the attention of managers.

Users of Microsoft Azure can decide to run virtual machines locally (private cloud), in the public cloud or in a combined environment (hybrid cloud)[64]. The new hybrid cloud platform Azure Stack offers the possibility to use Azure services with local computer center[65].

Google Cloud Platform supports public cloud but not private cloud, because their cloud hosting data are on their public cloud servers as it provides most efficient way to do things. Google is on the way to develop services for hybrid cloud but an official announcement has not been made yet at the time of writing[66].

### 3.4.7 Select Cloud Service Model

All the existing cloud service models and their combination are listed in Table 3.8. Amazon EC2, Azure Virtual Machines and Google Compute Engine are defined as IaaS services. As they are responsible for provisioning virtual machines that can host platforms or applications or both, they are in the position to combine with PaaS and SaaS services.

[61] http://docs.aws.amazon.com/govcloud-us/latest/UserGuide/welcome.html
[62] https://aws.amazon.com/de/enterprise/hybrid/
[63] https://azure.microsoft.com/de-de/overview/azure-stack/
[64] https://azure.microsoft.com/de-de/services/virtual-machines/
[65] https://azure.microsoft.com/de-de/overview/azure-stack/
[66] http://www.cmswire.com/cms/information-management/google-vmware-partner-for-hybrid-cloud-computing-027906.php

**Table 3.7:** Relations between outcomes of the decision Select Cloud Vendor toward outcomes of the decision Select Cloud Deployment Model

| | Public Cloud | Private Cloud | Community Cloud | Hybrid Cloud |
|---|---|---|---|---|
| **Amazon services** | **Select Cloud Deployment Model** | | | |
| Amazon EC2 | a | ex | a | a |
| Amazon ECS | a | ex | a | a |
| AWS Elastic Beanstalk | a | ex | a | a |
| Amazon VPC | a | a | a | a |
| Amazon S3 | a | ex | a | a |
| **Azure services** | | | | |
| Azure Virtual Machines | a | a | ex | a |
| Azure Container Service | a | a | ex | a |
| Azure App Service | a | a | ex | a |
| Azure Virtual Network | a | a | ex | a |
| Azure Blob Storage | a | a | ex | a |
| **Google services** | | | | |
| Google Compute Engine | a | ex | ex | ex |
| Google Container Engine | a | ex | ex | ex |
| Google App Engine | a | ex | ex | ex |
| Google Compute Engine Network | a | ex | ex | ex |
| Google Cloud Storage | a | ex | ex | ex |

Amazon ECS, Azure Container Service and Google Container Engine is the container layer built inside virtual machines and can also host platforms and applications. As a result, they can be used together with other PaaS and SaaS services, too.

AWS Elastic Beanstalk, Azure App Service and Google App Engine are defined as PaaS services. They are responsible for provisioning resources to run applications and store data as well as scaling applications. They are therefore able to work together with IaaS and SaaS services.

Amazon VPC, Azure Virtual Network and Google Compute Engine Network are defined as IaaS services.

Amazon S3, Azure Blob Storage and Google Cloud Storage are responsible for storing cloud data and naturally the IaaS services.

**Table 3.8:** Relations between outcomes of the decision Select Cloud Vendor toward outcomes of the decision Select Cloud Service Model

| | IaaS | PaaS | SaaS | IaaS + PaaS | IaaS + SaaS | PaaS + SaaS | IaaS + PaaS + SaaS |
|---|---|---|---|---|---|---|---|
| **Amazon services** | **Select Cloud Service Model** | | | | | | |
| Amazon EC2 | a | ex | ex | a | a | ex | a |
| Amazon ECS | a | ex | ex | a | a | ex | a |
| AWS Elastic Beanstalk | ex | a | ex | a | ex | a | ex |
| Amazon VPC | a | ex | ex | ex | ex | ex | ex |
| Amazon S3 | a | ex | ex | ex | ex | ex | ex |
| **Azure services** | | | | | | | |
| Azure Virtual Machines | a | ex | ex | a | a | ex | a |
| Azure Container Service | a | ex | ex | a | a | ex | a |
| Azure App Service | ex | a | ex | a | ex | a | ex |
| Azure Virtual Network | a | ex | ex | ex | ex | ex | ex |
| Azure Blob Storage | a | ex | ex | ex | ex | ex | ex |
| **Google services** | | | | | | | |
| Google Compute Engine | a | ex | ex | a | a | ex | a |
| Google Container Engine | a | ex | ex | a | a | ex | a |
| Google App Engine | ex | a | ex | a | ex | a | ex |
| Google Compute Engine Network | a | ex | ex | ex | ex | ex | ex |
| Google Cloud Storage | a | ex | ex | ex | ex | ex | ex |

### 3.4.8 Define Cloud Hosting

As seen in Table 3.9 AWS does not allow on-premise hosting of its services. In case a solution of hybrid cloud is selected, AWS provides enterprises with tools to integrate AWS and enterprise resources to leverage its services. For example, Avere Hybrid Cloud NAS is used to enable application bursting on Amazon EC2 and hybrid storage infrastructure that includes Amazon S3. With Avere, AWS resources become a seamless part of enterprise architecture by storing data anywhere with NAS features like high availability and scalability[67].

Azure supports private cloud and therefore allows on-premise hosting of cloud resources. It also provides tools for network and data integration in hybrid cloud. An example is Hybrid Connections, a feature of Azure BizTallk Services which provide an easy and convenient way to connect the Web Apps features in Azure App Service to on-premise resources.

Google supports solely public cloud and therefore allows only on-premise hosting of cloud resources.

### 3.4.9 Select Pricing Model

The pricing models applied to services are summarized in Table 3.10.

Amazon EC2 is free in the first twelve months since the date of register, if the monthly use is not much than 750 hours for Linux-, RHEL- or SLES t2.micro-instance or Windows t2.micro-instance. The pay-per-use model is applied for on-demand, spot-instances or reserved instances utilization (No Upfront). The utilization of reservered instances in form of All Upfront or Partial Upfront is billed due to charge-per-use model[68].

Amazon ECS and AWS Elastic Beanstalk are free for use. Users just have to pay for AWS resources (e.g. EC2 instances or EBS volumes) used for storage or application running. There is neither minimum fees nor advance payment applied for ECS service.

The price for using Amazon VPC is paid per VPN connection hour, per each NAT gateway or per GB data processed.

Amazon S3 is free for the first 12 months if the monthly use does not exit 5 GB standard storage, 20000 GET requests and 2000 PUT requests. Beyond this time and volume it is monthly charged per storage.

Azure Virtual Machines service requires no advance payment, no termination fee and usage is billed per minute.

Azure Container Service and Azure Virtual Network are also free of charge. Users just have to pay for virtual machines as well as storage and network resources used.

[67]https://aws.amazon.com/de/backup-recovery/partner-solutions/
[68]https://aws.amazon.com/de/ec2/pricing/

**Table 3.9:** Relations between outcomes of the decision Select Cloud Vendor toward outcomes of the decision Define Cloud Hosting

| | On-Premise Hosting | Off-Premise Hosting | Hybrid Hosting |
|---|---|---|---|
| **Amazon services** | **Define Cloud Hosting** | | |
| Amazon EC2 | ex | a | in |
| Amazon ECS | ex | a | in |
| AWS Elastic Beanstalk | ex | a | in |
| Amazon VPC | ex | a | in |
| Amazon S3 | ex | a | in |
| **Azure services** | | | |
| Azure Virtual Machines | a | a | in |
| Azure Container Service | a | a | in |
| Azure App Service | a | a | in |
| Azure Virtual Network | a | a | in |
| Azure Blob Storage | a | a | in |
| **Google services** | | | |
| Google Compute Engine | ex | a | ex |
| Google Container Engine | ex | a | ex |
| Google App Engine | ex | a | ex |
| Google Compute Engine Network | ex | a | ex |
| Google Cloud Storage | ex | a | ex |

Azure Blob Storage and Google Cloud Storage services are charged due to monthly volume stored.

Azure App Service offers free service plan which is suitable for 10 applications with 200 daily logical actions and 1GB hard disk space. Other service plans (shared, basic, standard and premium) are charged upon size and per hour base.

Google Compute Engine offers predefined machine types and custom machine types. All machine types are charged a minimum of ten minutes, after that instances are charged in one minute increments.

Google Container Engine charges a flat fee per hour per cluster for the cluster management, depending on the number of nodes in that cluster.

Google App Engine applications run as instances within the standard environment or the flexible environment. Instances within the standard environment have access to a daily limit of resource usage that is provided at no charge defined by a set of quotas. Beyond that level, applications will incur charges.

Google Cloud Storage charge users per GB per month.

### 3.4.10 Define Resource Location

AWS, Microsoft Azure and Google Cloud Platform each own data centers in different regions of the world. They replicate and cache data in locations closed to the end-user's location to lower the latency to access the data. Upon creating an account to use each of those platforms, it is possible for a user to choose one or more available regions to host their data are. As a result, data are not limited to be in the same jurisdiction or in different jurisdiction that the enterprises operate (see Table 3.11).

## 3.5 Relations toward Outcomes of the decision Select Cloud Vendor

In section 3.4, a relation X is defined between each outcome A of the decision Select Cloud Vendor and each outcome B of another decision. This section reversely defines the relation Y pointing form outcome B toward outcome A. In most of the cases Y is of the same relationship type with X, which makes the bilateral relation between A and B. There are only some exceptions:

- The relations of type conditional allowing and allowing plus existing between A and B, because in such cases A is a part of a specific scenario that enables B. In the vice versa direction, when B is selected, A is simply allowed to be selected generally.
- When offerings of Amazon Web Services are selected, both data in the same jurisdiction and in different jurisdiction are allowed, because AWS gives its users the capability to freely select the regions and zones available. In the opposite direction, when customers require that data should be only within the same jurisdiction with the location where their enterprises locate, the requirement is not necessarily satisfied, because Amazon has a limited number of data centers

**Table 3.10:** Relations between outcomes of the decision Select Cloud Vendor toward outcomes of the decision Select Pricing Model

| | Free | Pay-Per-Use | Pay-Per-Unit | Charge-Per-Use (Subscription) |
|---|---|---|---|---|
| **Amazon services** | | **Select Pricing Model** | | |
| Amazon EC2 | ca | a | ex | a |
| Amazon ECS | a | ex | ex | ex |
| AWS Elastic Beanstalk | a | ex | ex | ex |
| Amazon VPC | ex | a | ex | ex |
| Amazon S3 | ca | ex | a | ex |
| **Azure services** | | | | |
| Azure Virtual Machines | ex | a | ex | ex |
| Azure Container Service | a | ex | ex | ex |
| Azure App Service | a | a | ex | ex |
| Azure Virtual Network | a | ex | ex | ex |
| Azure Blob Storage | ex | ex | a | ex |
| **Google services** | | | | |
| Google Compute Engine | ex | a | ex | ex |
| Google Container Engine | ex | a | ex | ex |
| Google App Engine | a | ex | ex | ex |
| Google Compute Engine Network | a | ex | ex | ex |
| Google Cloud Storage | ex | ex | a | ex |

**Table 3.11:** Relations between outcomes of the decision Select Cloud Vendor toward outcomes of the decision Define Resource Location

| | Data In Same Jurisdiction | Data In Different Jurisdiction |
|---|---|---|
| **Amazon services** | **Define Resource Location** | |
| Amazon EC2 | a | a |
| Amazon ECS | a | a |
| AWS Elastic Beanstalk | a | a |
| Amazon VPC | a | a |
| Amazon S3 | a | a |
| **Azure services** | | |
| Azure Virtual Machines | a | a |
| Azure Container Service | a | a |
| Azure App Service | a | a |
| Azure Virtual Network | a | a |
| Azure Blob Storage | a | a |
| **Google services** | | |
| Google Compute Engine | a | a |
| Google Container Engine | a | a |
| Google App Engine | a | a |
| Google Compute Engine Network | a | a |
| Google Cloud Storage | a | a |

which are located in North Virginia, Oregon, North California, Ireland, Frankfurt, Singapore, Tokyo, Sydney, Seoul, Mumbai and São Paulo, not always in the same jurisdiction with location of clients. As a result, this type of requirement can only be arbitrarily fulfilled. The same argument is applied to services of Microsoft Azure and Google Cloud Platform, because although Azure owns the biggest number of data centers among cloud providers, they are limited to some cities of America, Europe and Asia, Google also has data centers in the US, Western Europe and Eastern Asia.

Tables 3.12, 3.13 and 3.14 show the relations of other outcomes toward the services offered by each provider.

**Table 3.12:** Relations between other outcomes in the CloudDSF+2 knowledge base toward services of Amazon Web Services

| | **Amazon EC2** | **Amazon ECS** | **AWS Elastic Beanstalk** | **Amazon VPC** | **Amazon S3** |
|---|---|---|---|---|---|
| **Define Scalability Level** | | | | | |
| No Scaling | a | a | a | a | ex |
| VM Level Scaling | a | a | a | a | ex |
| Middleware Level Scaling | ex | ex | ex | ex | a |
| Application Level Scaling | ex | ex | a | ex | ex |
| VM + Middleware Level Scaling | ex | ex | ex | ex | ex |
| VM + Application Level Scaling | ex | ex | ex | ex | ex |
| Middleware + Application Level Scaling | ex | ex | ex | ex | ex |
| VM + Middleware + Application Level Scaling | ex | ex | ex | ex | ex |
| **Select Scaling Type** | | | | | |
| Vertical Scaling | a | ex | ex | ex | a |
| Horizontal Scaling | a | a | a | a | a |
| Hybrid Scaling | a | ex | ex | ex | a |
| **Select Elasticity Automation Degree** | | | | | |
| Manual Scaling | a | a | a | ex | ex |
| Semi-Automatic Scaling | a | ex | ex | ex | ex |

| | | | | | |
|---|---|---|---|---|---|
| Semi-Automatic Third-Party Scaling | ex | ex | ex | ex | ex |
| Automatic Scaling | a | a | a | a | a |
| Automatic Third-Party Scaling | ex | ex | ex | ex | ex |
| **Select Scaling Trigger** | | | | | |
| No Trigger | a | a | a | ex | a |
| Event-Driven Trigger | a | a | a | a | ex |
| Proactive Trigger | ex | ex | ex | ex | ex |
| **Select Multi-Tenancy Level** | | | | | |
| Shared Hardware Multi-Tenancy | a | a | a | a | a |
| Shared OS Multi-Tenancy | ex | a | ex | ex | ex |
| Shared Middleware Multi-Tenancy | ex | ex | ex | ex | ex |
| Shared Application Multi-Tenancy | ex | ex | ex | ex | ex |
| **Select Cloud Deployment Model** | | | | | |
| Public Cloud | a | a | a | a | a |
| Private Cloud | ex | ex | ex | a | ex |
| Community Cloud | a | a | a | a | a |
| Hybrid Cloud | a | a | a | a | a |
| **Select Cloud Service Model** | | | | | |
| IaaS | a | a | ex | a | a |
| PaaS | ex | ex | a | ex | ex |
| SaaS | ex | ex | ex | ex | ex |
| IaaS + PaaS | a | a | a | ex | ex |
| IaaS + SaaS | a | a | ex | ex | ex |
| PaaS + SaaS | ex | ex | a | ex | ex |
| IaaS + PaaS + SaaS | a | a | ex | ex | ex |
| **Define Cloud Hosting** | | | | | |
| On-Premise Hosting | ex | ex | ex | ex | ex |
| Off-Premise Hosting | a | a | a | a | a |
| Hybrid Hosting | in | in | in | in | in |
| **Select Pricing Model** | | | | | |
| Free | a | a | a | ex | a |
| Pay-Per-Use | a | ex | ex | a | ex |
| Pay-Per-Unit | ex | ex | ex | ex | a |
| Charge-Per-Use (Subscription) | a | ex | ex | ex | ex |
| **Define Resource Location** | | | | | |
| Data In Same Jurisdiction | aa | aa | aa | aa | aa |
| Data In Different Jurisdiction | a | a | a | a | a |

**Table 3.13:** Relations between other outcomes in the CloudDSF+2 knowledge base toward services of Microsoft Azure

| | Azure Virtual Machines | Azure Container Service | Azure App Service | Azure Virtual Network | Azure Blob Storage |
|---|---|---|---|---|---|
| **Define Scalability Level** | | | | | |
| No Scaling | a | a | a | a | ex |
| VM Level Scaling | a | a | a | a | ex |
| Middleware Level Scaling | ex | ex | ex | ex | a |
| Application Level Scaling | ex | ex | a | ex | ex |
| VM + Middleware Level Scaling | ex | ex | ex | ex | ex |
| VM + Application Level Scaling | ex | ex | ex | ex | ex |
| Middleware + Application Level Scaling | ex | ex | ex | ex | ex |
| VM + Middleware + Application Level Scaling | ex | ex | ex | ex | ex |
| **Select Scaling Type** | | | | | |
| Vertical Scaling | a | ex | a | ex | a |
| Horizontal Scaling | a | a | a | a | a |
| Hybrid Scaling | a | ex | ex | ex | a |
| **Select Elasticity Automation Degree** | | | | | |
| Manual Scaling | a | a | a | ex | ex |
| Semi-Automatic Scaling | a | ex | ex | ex | ex |
| Semi-Automatic Third-Party Scaling | ex | ex | ex | ex | ex |
| Automatic Scaling | a | a | a | a | a |
| Automatic Third-Party Scaling | ex | ex | ex | ex | ex |
| **Select Scaling Trigger** | | | | | |
| No Trigger | a | a | a | ex | a |
| Event-Driven Trigger | a | ex | a | a | ex |
| Proactive Trigger | ex | ex | ex | ex | ex |
| **Select Multi-Tenancy Level** | | | | | |
| Shared Hardware Multi-Tenancy | a | a | a | a | a |

| | | | | | |
|---|---|---|---|---|---|
| Shared OS Multi-Tenancy | ex | ex | ex | ex | ex |
| Shared Middleware Multi-Tenancy | ex | ex | ex | ex | ex |
| Shared Application Multi-Tenancy | ex | ex | ex | ex | ex |
| **Select Cloud Deployment Model** | | | | | |
| Public Cloud | a | a | a | a | a |
| Private Cloud | a | a | a | a | a |
| Community Cloud | ex | ex | ex | ex | ex |
| Hybrid Cloud | a | a | a | a | a |
| **Select Cloud Service Model** | | | | | |
| IaaS | a | a | ex | a | a |
| PaaS | ex | ex | a | ex | ex |
| SaaS | ex | ex | ex | ex | ex |
| IaaS + PaaS | a | a | a | ex | ex |
| IaaS + SaaS | a | a | ex | ex | ex |
| PaaS + SaaS | ex | ex | a | ex | ex |
| IaaS + PaaS + SaaS | a | a | ex | ex | ex |
| **Define Cloud Hosting** | | | | | |
| On-Premise Hosting | a | a | a | a | a |
| Off-Premise Hosting | a | a | a | a | a |
| Hybrid Hosting | in | in | in | in | in |
| **Select Pricing Model** | | | | | |
| Free | ex | a | a | a | ex |
| Pay-Per-Use | a | ex | a | ex | ex |
| Pay-Per-Unit | ex | ex | ex | ex | a |
| Charge-Per-Use (Subscription) | ex | ex | ex | ex | ex |
| **Define Resource Location** | | | | | |
| Data In Same Jurisdiction | aa | aa | aa | aa | aa |
| Data In Different Jurisdiction | a | a | a | a | a |

**Table 3.14:** Relations between other outcomes in the CloudDSF+2 knowledge base toward services of Google Cloud Platform

| | Google Compute Engine | Google Container Engine | Google App Engine | Google Compute Engine Network | Google Cloud Storage |
|---|---|---|---|---|---|
| **Define Scalability Level** | | | | | |
| No Scaling | a | a | a | a | ex |
| VM Level Scaling | a | a | a | a | ex |
| Middleware Level Scaling | ex | ex | ex | ex | a |
| Application Level Scaling | ex | ex | a | ex | ex |
| VM + Middleware Level Scaling | ex | ex | ex | ex | ex |
| VM + Application Level Scaling | ex | ex | ex | ex | ex |
| Middleware + Application Level Scaling | ex | ex | ex | ex | ex |
| VM + Middleware + Application Level Scaling | ex | ex | ex | ex | ex |
| **Select Scaling Type** | | | | | |
| Vertical Scaling | a | a | ex | ex | a |
| Horizontal Scaling | a | a | a | a | a |
| Hybrid Scaling | a | ex | ex | ex | a |
| **Select Elasticity Automation Degree** | | | | | |
| Manual Scaling | a | a | ex | ex | ex |
| Semi-Automatic Scaling | a | ex | ex | ex | ex |
| Semi-Automatic Third-Party Scaling | ex | ex | ex | ex | ex |
| Automatic Scaling | a | a | a | a | a |
| Automatic Third-Party Scaling | ex | ex | ex | ex | ex |
| **Select Scaling Trigger** | | | | | |
| No Trigger | a | a | ex | ex | a |
| Event-Driven Trigger | a | a | a | a | ex |
| Proactive Trigger | ex | ex | ex | ex | ex |

| | | | | | |
|---|---|---|---|---|---|
| **Select Multi-Tenancy Level** | | | | | |
| Shared Hardware Multi-Tenancy | a | a | a | a | a |
| Shared OS Multi-Tenancy | ex | a | ex | ex | ex |
| Shared Middleware Multi-Tenancy | ex | ex | a | ex | ex |
| Shared Application Multi-Tenancy | ex | ex | ex | ex | ex |
| **Select Cloud Deployment Model** | | | | | |
| Public Cloud | a | a | a | a | a |
| Private Cloud | ex | ex | ex | ex | ex |
| Community Cloud | ex | ex | ex | ex | ex |
| Hybrid Cloud | ex | ex | ex | ex | ex |
| **Select Cloud Service Model** | | | | | |
| IaaS | a | a | ex | a | a |
| PaaS | ex | ex | a | ex | ex |
| SaaS | ex | ex | ex | ex | ex |
| IaaS + PaaS | a | a | a | ex | ex |
| IaaS + SaaS | a | a | ex | ex | ex |
| PaaS + SaaS | ex | ex | a | ex | ex |
| IaaS + PaaS + SaaS | a | a | ex | ex | ex |
| **Define Cloud Hosting** | | | | | |
| On-Premise Hosting | ex | ex | ex | ex | ex |
| Off-Premise Hosting | a | a | a | a | a |
| Hybrid Hosting | ex | ex | ex | ex | ex |
| **Select Pricing Model** | | | | | |
| Free | ex | ex | a | a | ex |
| Pay-Per-Use | a | a | ex | ex | ex |
| Pay-Per-Unit | ex | ex | ex | ex | a |
| Charge-Per-Use (Subscription) | ex | ex | ex | ex | ex |
| **Define Resource Location** | | | | | |
| Data In Same Jurisdiction | aa | aa | aa | aa | aa |
| Data In Different Jurisdiction | a | a | a | a | a |

## 3.6 Summary of the Extension

In order to specify the relations between outcomes of the decision Select Cloud Vendor toward outcomes of other decisions in the CloudDSF+2 knowledge base and vice versa, three more relationship types namely allowing plus, conditional allowing and arbitrary allowing were defined, which together with allowing, including and excluding relations make a set of total six relations between outcomes. The occurrence of those relations are listed in Table 3.15, Table 3.16 and Table 3.17. Each of these tables is divided into two halves, the left one shows the relations from outcomes of the decision Select Cloud Vendor toward outcomes of other decision, the right one shows the opposite direction.

It is obvious that the relations listed in those tables are unilateral. However, the number of relations in the left hand side and in the right hand side of the same line are in most of the cases equal, which means that there usually exists a bilateral relation between the listed outcomes. There are also cases of actual unilateral relations, for example between outcomes of the decision Define Resource Location and services of the considered platforms, as explained in Section 3.5.

These three tables of the section provide the overview of each platform in consideration. It is possible to compare the number of relations from and to services of Amazon Web Services with that of Microsoft Azure and Google Cloud Platform regarding each decision. However, the difference in the number of relations does not translate into the powerfulness of one platform over the others, but only present the different functionalities and features that each platform provides. Further information about such features are described in detail in Section 3.4.

A total number of 1285 relations of all relationship types are replacing twenty affecting and binding relations of the previous CloudDSF+ knowledge base, including 489 allowing relations, 20 including relations and 732 excluding relations. The newly defined relation type conditional allowing was used only 12 times, allowing plus 22 times and arbitrary allowing 15 times. Due to the definition of those relation types, such a limited occurrence tells about the independence and reusability of offered services.

The added relations regarding the decision Select Pricing Model provide a coarse-grained comparison between providers about the way the fees are calculated. It should be mentioned that a direct comparison on the price of the same storage volumes and resources used in a specific scenario is not provided in CloudDSF+2 framework.

**Table 3.15:** Quantification of outcome relations(Amazon Web Services)

| | AWS toward decisions | | | | | | Decisions toward AWS | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Decisions** | **a** | **ex** | **ap** | **ca** | **aa** | **in** | **a** | **ex** | **ap** | **ca** | **aa** | **in** |
| Define Scalability Level | 5 | 30 | 5 | 0 | 0 | 0 | 10 | 30 | 0 | 0 | 0 | 0 |
| Select Scaling Type | 8 | 6 | 1 | 0 | 0 | 0 | 9 | 6 | 0 | 0 | 0 | 0 |
| Select Elasticity Automation Degree | 8 | 16 | 1 | 0 | 0 | 0 | 9 | 16 | 0 | 0 | 0 | 0 |
| Select Scaling Trigger | 3 | 7 | 2 | 3 | 0 | 0 | 8 | 7 | 0 | 0 | 0 | 0 |
| Select Multi-Tenancy Level | 5 | 14 | 1 | 0 | 0 | 0 | 6 | 14 | 0 | 0 | 0 | 0 |
| Select Cloud Deployment Model | 16 | 4 | 0 | 0 | 0 | 0 | 16 | 4 | 0 | 0 | 0 | 0 |
| Select Cloud Service Model | 13 | 22 | 0 | 0 | 0 | 0 | 13 | 22 | 0 | 0 | 0 | 0 |
| Define Cloud Hosting | 5 | 5 | 0 | 0 | 0 | 5 | 5 | 5 | 0 | 0 | 0 | 5 |
| Select Pricing Model | 6 | 12 | 0 | 2 | 0 | 0 | 8 | 12 | 0 | 0 | 0 | 0 |
| Define Resource Location | 10 | 0 | 0 | 0 | 0 | 0 | 5 | 0 | 0 | 0 | 5 | 0 |

| | **A** | **Ex** | **AP** | **CA** | **AA** | **in** | **Total** |
|---|---|---|---|---|---|---|---|
| **Occurrences** | **168** | **232** | **10** | **5** | **5** | **10** | **430** |

**Table 3.16:** Quantification of outcome relations (Microsoft Azure)

| | Azure toward decisions | | | | | | Decisions toward Azure | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Decisions** | **a** | **ex** | **ap** | **ca** | **aa** | **in** | **a** | **ex** | **ap** | **ca** | **aa** | **in** |
| Define Scalability Level | 5 | 30 | 5 | 0 | 0 | 0 | 10 | 30 | 0 | 0 | 0 | 0 |
| Select Scaling Type | 9 | 5 | 1 | 0 | 0 | 0 | 10 | 5 | 0 | 0 | 0 | 0 |
| Select Elasticity Automation Degree | 9 | 16 | 0 | 0 | 0 | 0 | 9 | 16 | 0 | 0 | 0 | 0 |
| Select Scaling Trigger | 5 | 8 | 0 | 2 | 0 | 0 | 7 | 8 | 0 | 0 | 0 | 0 |
| Select Multi-Tenancy Level | 5 | 15 | 0 | 0 | 0 | 0 | 5 | 15 | 0 | 0 | 0 | 0 |
| Select Cloud Deployment Model | 15 | 5 | 0 | 0 | 0 | 0 | 15 | 5 | 0 | 0 | 0 | 0 |
| Select Cloud Service Model | 13 | 22 | 0 | 0 | 0 | 0 | 13 | 22 | 0 | 0 | 0 | 0 |
| Define Cloud Hosting | 10 | 0 | 0 | 0 | 0 | 5 | 10 | 0 | 0 | 0 | 0 | 5 |
| Select Pricing Model | 6 | 14 | 0 | 0 | 0 | 0 | 6 | 14 | 0 | 0 | 0 | 0 |
| Define Resource Location | 10 | 0 | 0 | 0 | 0 | 0 | 5 | 0 | 0 | 0 | 5 | 0 |

| | **A** | **Ex** | **AP** | **CA** | **AA** | **In** | **Total** |
|---|---|---|---|---|---|---|---|
| **Occurrences** | **177** | **230** | **6** | **2** | **5** | **10** | **425** |

**Table 3.17:** Quantification of outcome relations (Goolge Cloud Platform)

| | **Goolge toward decisions** | | | | | | **Decisions toward Goolge** | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Decisions** | **a** | **ex** | **ap** | **ca** | **aa** | **in** | **a** | **ex** | **ap** | **ca** | **aa** | **in** |
| Define Scalability Level | 5 | 30 | 5 | 0 | 0 | 0 | 10 | 30 | 0 | 0 | 0 | 0 |
| Select Scaling Type | 10 | 5 | 0 | 0 | 0 | 0 | 10 | 5 | 0 | 0 | 0 | 0 |
| Select Elasticity Automation Degree | 5 | 17 | 0 | 3 | 0 | 0 | 8 | 17 | 0 | 0 | 0 | 0 |
| Select Scaling Trigger | 5 | 8 | 0 | 2 | 0 | 0 | 7 | 8 | 0 | 0 | 0 | 0 |
| Select Multi-Tenancy Level | 6 | 13 | 1 | 0 | 0 | 0 | 7 | 13 | 0 | 0 | 0 | 0 |
| Select Cloud Deployment Model | 5 | 15 | 0 | 0 | 0 | 0 | 5 | 15 | 0 | 0 | 0 | 0 |
| Select Cloud Service Model | 13 | 22 | 0 | 0 | 0 | 0 | 13 | 22 | 0 | 0 | 0 | 0 |
| Define Cloud Hosting | 5 | 10 | 0 | 0 | 0 | 0 | 5 | 10 | 0 | 0 | 0 | 0 |
| Select Pricing Model | 5 | 15 | 0 | 0 | 0 | 0 | 5 | 15 | 0 | 0 | 0 | 0 |
| Define Resource Location | 10 | 0 | 0 | 0 | 0 | 0 | 5 | 0 | 0 | 0 | 5 | 0 |

| | **A** | **Ex** | **AP** | **CA** | **AA** | **In** | **Total** |
|---|---|---|---|---|---|---|---|
| **Occurrences** | **144** | **270** | **6** | **5** | **5** | **0** | **430** |

# 4 Implementation and evaluation of CloudDSF+2 Prototype

This chapter introduces the visualization of of newly added relationship types as well as relations between additional outcomes and existing outcomes of the CloudDSF+2 knowledge base. Then CloudDSF+2 is evaluated in terms of improvements and limitations in comparison with its previous version CloudDSF+ provided in [Bal15].

## 4.1 Implementation of CloudDSF+2

This latest implementation of CloudDSF+2 does not modify the architecture of CloudDSF+, but simply utilizes the same technologies and libraries mentioned in [Bal15] to visualize the additional relationship types and relations described in its knowledge base.

Following modifications were made to the CloudDSF+ prototype:

**KB Visualizer**

The static visualization rendered in KB Visualizer of the CloudDSF+ prototype includes three layouts namely Hierarchical Layout, Decision Relations Layout and Outcome Relations Layout. In Hierarchical Layout of CloudDSF+2, the placeholder outcome of the decision Select Cloud Vendor was replaced by fifteen new outcomes, as shown in Figure 4.1.

In the Decision Relations Layout page, the styles of the arrows representing different relationship types at the decision level were changed to a unified style and visualized in different colors to make them more descriptive and distinctive (see Figure 4.2). The same alteration was applied to arrows that depict relationship types at the outcome level. In this static environment, it is possible to render up to thousands of relations between outcomes at the same time, which results in a very high density and the colorful arrows support the visualization better, as seen in Figure 4.3.

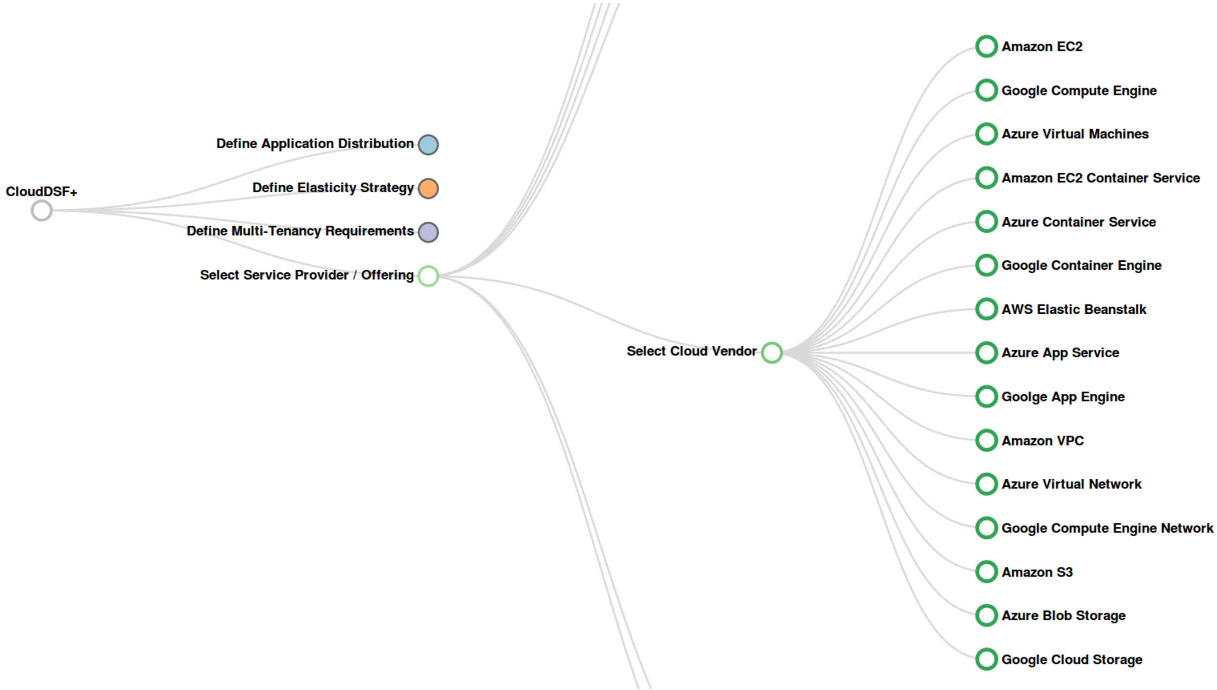


**Figure 4.1:** New outcomes in the hierarchical layout


**Figure 4.2:** Arrows presenting relations between decisions in new style

### KB Navigator

In this page, the placeholder outcome of the decision Select Cloud Vendor was also replaced by fifteen actual outcomes of the three selected vendors Amazon, Microsoft and Google. The functionalities of the KB Navigator remain mainly the same as described in [Bal15], with an adjustment. Normally the selection of an outcome excludes other outcomes of the same decision from further selections by marking them gray (inactive). The user is questioned if the action is really intentional when another outcome of the same decision is selected. If the answer is "yes", the newly selected outcome will become active. This rule is however not applied to newly added outcomes of the decision Select Cloud Vendor, because it is semantically and therefore technically possible to select multiple services at the same time.


**Figure 4.3:** Arrows presenting relations between outcomes in new style

**CloudDSF+2 Parser**

As discussed in [Bal15], the parser reads the CloudDSF+ knowledge base in form of an excel file and transforms it into two JSON files, which serve as input for the CloudDSF+ visualizer. Because the knowledge base is updated with fifteen outcomes of the decision Select Cloud Vendor, the rules that validate the updated knowledge base should also be modified. For the convenience of the readers, these rules are listed here, some of them remain the same as in [Bal15] and some are modified to fit with the updated knowledge base.

- On the level of decisions, only influencing, affecting, binding and requiring relationship types are allowed.
- On the level of outcomes, only including, excluding, allowing, allowing plus, prerequisite allowing and arbitrary allowing relationship types are allowed.
- On the level of decisions only requiring relations can be combined with other relationship types. Therefore, influencing, allowing and binding relationships cannot coexist from one decision to another.
- If a relation from decision A to decision B exists, there must also be relationships from any outcome of decision A to any of the outcomes of decision B.
- If a relation from outcome A to outcome B exists, there must also be a relationship from the respective decision of outcome A to the respective decision of outcome B.
- Binding and affecting relations are complimentary to each other. Logically, if a binding relation from decision A towards decision B exists, in the reverse case, an affecting relationship must be present and vice versa.
- If an including relation from outcome A to outcome B exists, in the case a relation exists in reverse, it must also be of the including or allowing relationship type. Otherwise a contradiction would exist.
- Any given outcome can only have one relation towards another outcome.
- Between outcomes of the same decision an exclusive or (XOR) relation were specified. Hence, as soon as an outcome is selected all others of the respective decision are not applicable anymore. Therefore, defined relations between outcomes of the same decision never apply and would unnecessarily pollute the knowledge base. As a consequence, any given outcome is only allowed to have relations toward outcomes of other decisions.

Those rules are respectively implemented by taking the available code provided together with [Bal15] and modifying it accordingly.

The CloudDSF+2 Prototype as well as the CloudDSF+2 Parser has been forked from the respective code repository[1]. The source code is publicly available at GitHub[2] under the Apache 2.0 license[3].

## 4.2 Evaluation of CloudDSF+2

In [Bal15], the efficacy of CloudDSF+ is evaluated through a real world task of deriving a migration strategy for the information technology (IT) infrastructure of the School of Computer Science at the University of St Andrews. The full and detailed evaluation process is provided in [Bal15], and additionally summarized here for the sake of continuity:

#### Description of use case

The IT system of the School of Computer Science is providing seven services for 60 staff members and 340 students.

- **Archive**: provides archiving functionality to all of the storage services of the school with 560 Gigabyte of data. It is hosted on a storage server.
- **StaffRes and StudRes**: enable staff and students to manage or procure the materials for courses/lectures, respectively. Both services are predominantly used at the start and end of a term. It can be assumed that both systems access the same resources and can be actually treated as a single application. Each service is hosted on an application server whereas the necessary data are hosted on a storage server.
- **Website**: is outdated and suffers from performance problems that might occur due to excessive loads in the university network. It is hosted on an application server.
- **WebDev**: is used as a testing ground for the aforementioned website or as a backup in case the main server for the website is not available. This service is logically hosted in the same location as the website but is very rarely used.

[1] https://github.com/bametz/clouddsfPlus
[2] https://github.com/minhthudo/clouddsfPlus
[3] http://www.apache.org/licenses/LICENSE-2.0

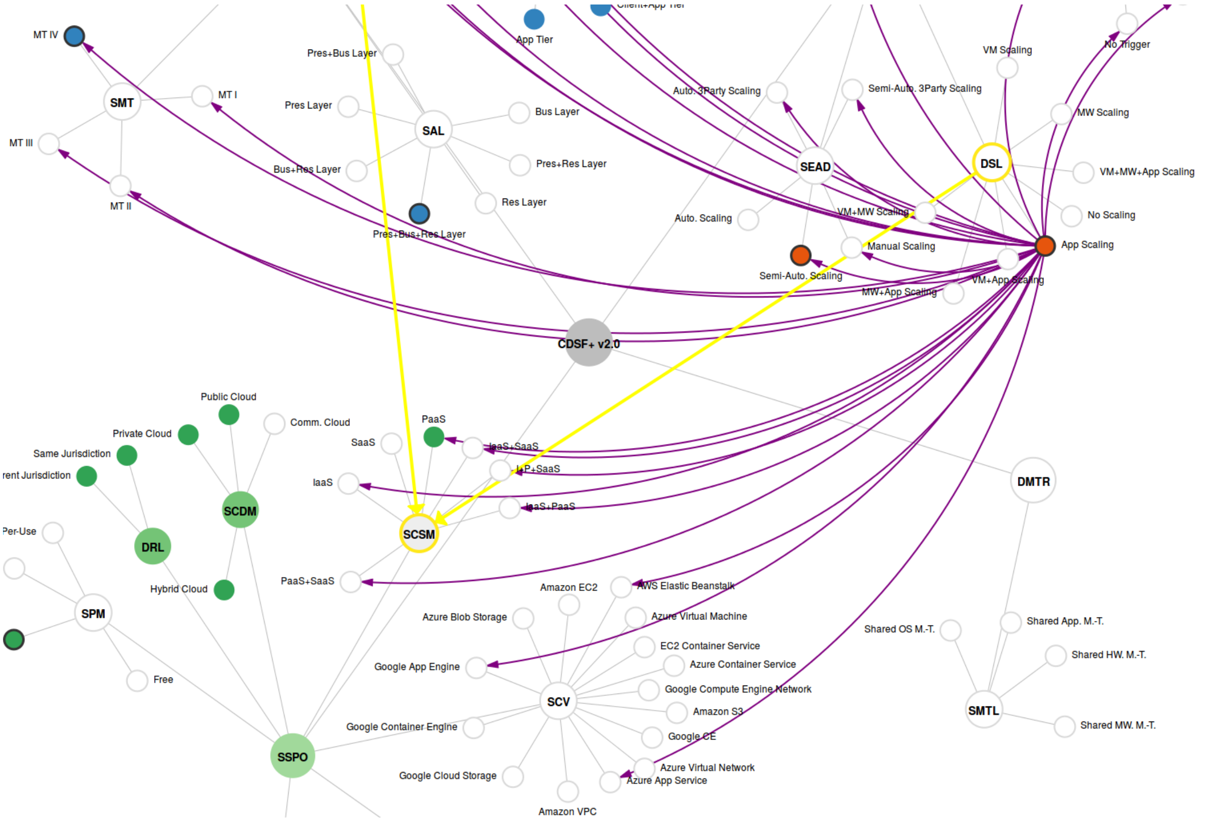


**Figure 4.4:** Selection of platform services for migrating Website

- **WebApps**: provides services like blogs, public wikis and software downloads, which are virtually hosted on a non-dedicated Apache server because of their very small usage and resource consumption.
- **Home directories mirror**: replicates the home directories for all students and staff. It is hosted on a storage server.
- **Teaching**: is a server that hosts student projects. The service runs 24 weeks annually since it is in use only during the terms.

### Migration strategy with updated decision Select Cloud Vendor

In this paragraph, the existing migration strategy described in [Bal15] is briefly summarized. Then the selection of suitable services and providers in accordance with preselected outcomes is added. An example on the selection of suitable services is shown in Figure 4.4.

**Migration decisions for the whole system** Upon consideration of the overall context and technical situation of the system to be migrated, the following options are selected for the whole system:

- Decision Define Cloud Hosting: Off-premise Hosting
- Decision Define Roles of responsibility: Management
- Decision Select Cloud Deployment Model: Private Cloud
- Decision Define Resource Location: Data In Same Jurisdiction

*Note: In the previous strategy, the default cloud vendor was Amazon. Now it can be Amazon, Microsoft or Google. However at this point, it is not yet possible to select any specific services.*

**Archive and Home Directory Mirror** The following options are selected for these two services:

- Decision Define Elasticity Automation Degree: Automatic Third-Party Scaling (Amazon is considered the third party)
- Decision Define Scalability Level: Middleware Level Scaling
- Decision Select Scaling Type: Hybrid Scaling
- Decision Select Application Layer: Resource Layer
- Decision Select Application Component: Middleware Components
- Decision Select Migration Type: Migration Type II
- Decision Select Cloud Service Model: IaaS
- Decision Select Pricing Model: Pay-Per-Use

*Note: With the selection of Middleware Level Scaling, Hybrid Scaling and the cloud service model IaaS, it is possible to choose one of three storage services Azure Blob Service, Amazon S3 or Goolge Cloud Storage. However, these services are not supporting the pricing model Pay-Per-Use, but providing Pay-Per-Unit, which is also another type of charging the actual consuming resources. As a result, the outcome Pay-Per-Use should be changed to Pay-Per-Unit, otherwise none of exiting services can be in use.*

**Website** The following options are selected for the service:

- Decision Select Migration Type: Migration Type IV
- Decision Select Application Layer: Presentation + Business + Resource Layer
- Decision Select Application Component: Application + Middleware Components
- Decision Select Cloud Service Model: PaaS
- Decision Define Scalability Level: Application Level Scaling
- Decision Select Scaling Type: Horizontal Scaling
- Decision Define Elasticity Automation Degree: Semi-Automatic Scaling
- Decision Select Scaling Trigger: Event-Driven Trigger
- Decision Select Pricing Model: Charge-Per-Use

*Note: With the selection of the cloud service model PaaS, Application Level Scaling, Horizontal Scaling, Semi-Automatic Scaling, Even-Driven Trigger, it is possible to choose one of three services Azure App Service, AWS Elastic Beanstalk or Google App Engine. However, as discussed in the previous section of Archive and Home Directory Mirror, the selected pricing model should be changed to Pay-Per-Unit, otherwise none of these services can be selected.*

**Teaching** The following options are selected for the service:

- Decision Select Cloud Service Model: IaaS
- Decision Select Migration Type: Migration Type III
- Decision Define Scalability Level: VM Level Scaling
- Decision Select Scaling Type: Hybrid Scaling
- Decision Define Elasticity Automation Degree: Manual Scaling
- Decision Select Scaling Trigger: No Trigger
- Decision Select Pricing Model: Pay-Per-Unit

*Note: With the selection of the cloud service model IaaS, VM Level Scaling, Hybrid Scaling, Manual Scaling, No Trigger, it is possible to choose one of three services Amazon EC2, Azure Virtual Machines or Google Compute Engine. However, these services are supporting the pricing model Pay-Per-Use, and the selected model is Pay-Per-Unit.*

**StaffRes and StudRes** The following options are selected for the service:

- Decision Select Cloud Service Model: PaaS
- Decision Select Migration Type: Migration Type IV
- Decision Define Scalability Level: Application Level Scaling
- Decision Select Scaling Type: Horizontal Scaling
- Decision Define Elasticity Automation Degree: Semi-Automatic Scaling
- Decision Select Scaling Trigger: Event-Driven Trigger
- Decision Select Pricing Model: Pay-Per-Use/Charge-Per-Use

*Note: The selection of the cloud service model PaaS together with Application Level Scaling, Horizontal Scaling and the pricing model Pay-Per-Use lead to three services Azure App Service, AWS Elastic Beanstalk or Google App Engine. However, these services are not supporting Semi-Automatic Scaling, but Manual Scaling or Automatic Scaling.*

**WebApps** The following options are selected for the service:

- Decision Select Cloud Service Model: IaaS
- Decision Select Migration Type: Migration Type III
- Decision Define Scalability Level: No Scaling

*Note: Due to the simple selection of other outcomes, it is possible to use one of three services Amazon EC2, Azure Virtual Machine or Google Compute Engine.*

**Discusion**

The validation of the CloudDSF+2 knowledge base has been carried out by an implementation that satisfies both existing and newly defined rules. As stated in [Bal15], these rules have been inferred based on the assumptions and definitions stated during the refinement and extension of the knowledge base and need to be evaluated as well.

The refinement of the decision Select Cloud Vendor in the knowledge base and the implementation of the CloudDSF+2 prototype has made the framework more capable by allowing decision makers to select different dimensions of the migration strategy and at the same time determine the existing cloud services that satisfy all of these selections.

However, it has been shown through the use case that in many cases not only one but usually multiple competitive services provided by different cloud vendors satisfy a specific migration strategy. In such situation, there are several tasks of the initial framework CloudDSF [Daw14] that would need to be implemented to provide the vendor comparison. The task Vendor Benchmark inspects cloud vendors in terms of their reputation and capability [GJN13], while the two tasks Cost Analysis and Identification of acceptable QoS Levels consider their technical and functional aspects.

Furthermore only three cloud vendors and fifteen cloud services are taken into the knowledge base, which is a limited number in comparison with number of services provided and vendors in the cloud market. While more and more services are being launched every year, the extension of the knowledge base is still done manually. As a result, an automatic mechanism to update the outcomes of the pricing and vendor decisions should be built in order to make the framework stronger and more up-to-date.

# 5 Conclusion and future work

The challenge of migrating legacy applications to the cloud in a financially and technically efficient way requires the consideration of multiple technical and organizational aspects which are not always covered by the available decision support approaches described in Chapter 2. As a result, the extended CloudDSF+ was built based on the CloudDSF that aims at enabling decision makers to gather a sound information basis by means of tasks and then make necessary decisions prior to a migration. However, CloudDSF+ still needs to be completed to fully achieve this goal. This thesis addresses one of deficiencies of CloudDSF+, the missing specific cloud computing services offered by actual cloud vendors, and extends the framework into CloudDSF+2.

To this end, the placeholder outcome of the decision Select Cloud Vendor in the CloudDSF+ knowledge base, namely Evaluated Cloud Vendor, has been replaced with fifteen popular cloud computing services provided by the three leading vendors in the cloud computing market:

- Amazon Elastic Compute Cloud (Amazon EC2), Amazon EC2 Container Service (Amazon ECS), AWS Elastic Beanstalk (Amazon ESB), Amazon Virtual Private Cloud (Amazon VPC) and Amazon Simple Storage Service (Amazon S3) by Amazon Web Services (AWS)
- Azure Virtual Machine, Azure Container Service, Azure App Service, Azure Virtual Network and Azure Blob Storage by Microsoft Azure
- Google Compute Engine, Google Container Engine, Google App Engine, Google Compute Engine Network and Google Cloud Storage by Google Cloud Platform

The cloud vendors as well as their services have been briefly introduced, aiming at an overview on the functionalities of each service. In order to define the relations between the newly added outcomes and the existing outcomes of CloudDSF+, the two general relationship types binding and affecting at the outcome level have been replaced by six specific relationship types, namely including, excluding, allowing, allowing plus, prerequisite allowing and arbitrary allowing. While the first three relationship types have been formerly defined in CloudDSF+, the last three ones have been newly added. Quantitatively, a total number of 1285 relations of all relationship types are replacing twenty affecting and binding relations of the previous CloudDSF+ knowledge base, including 489 allowing relations, 20 including relations and 732 excluding relations. The newly defined relationship type conditional allowing was used only 12 times, allowing plus 22 times and arbitrary allowing 15 times.

The visualization of the CloudDSF+2 knowledge base has been modified accordingly. Fifteen outcomes of the decision Select Cloud Vendor have been added into the hierarchical structure of the knowledge base. In the static environment KB Visualizer that renders relations between decisions and between outcomes, the format of relation arrows have been changed to make them more descriptive and distinctive. Previously in the dynamic environment KB Navigator, users were allowed to select only one outcome of a decision at one point of time. However this restriction has not been applied to newly added outcomes of the decision Select Cloud Vendor, therefore it has been possible to select multiple cloud services at the same time.

In the final step, an evaluation of the CloudDSF+2 has been carried out. First, the validation of the CloudDSF+2 knowledge base has been performed. The previous version CloudDSF+ specified eleven rules that define a valid and consistent knowledge base. Some of those rules have become obsolete and some new rules have been added, making a new set of nine rules, which are part of the implementation of the CloudDSF+2 Parser. Second, the use case described in [Bal15] has been revisited in order to prove the efficacy of the CloudDSF+2 framework by deriving migration strategies and at the same time determining specific cloud services for the migration.

Although the refinement of the decision Select Cloud Vendor makes the CloudDSF+2 framework more capable, several shortcomings are identified. Although many cloud service providers are present in the cloud market, only three of them have been mentioned in this thesis. While each provider is providing tens of cloud services, it has been limited to five services to be added into the CloudDSF+2 knowledge base. Consequently, there might be situations when it is not possible to select a cloud service to carry out the available migration strategy of a system, because the suitable service is simply not included in the knowledge base.

There are also some deficiencies that were present prior to this work. A more comprehensive evaluation including the business domain should be carried out to verify the accuracy of the refined and extended knowledge base. The decision Select Pricing Model should be refined to offer users with price comparison in specific utilization scenario, which is useful when multiple cloud services equally satisfy the same migration strategy. In order to increase its scope of application, CloudDSF+2 could also be adapted for the decision support for engineering cloud native applications, which is a potential field of further research.

Besides these limitations, the CloudDSF+2 framework has advantages of versatility and intuitive interface that support decision makers smoothly through the decision process.

As a result, the CloudDSF+2 framework and its implementation is suitable to derive migration strategies and select cloud services, making a step toward more sophisticated decision support for application migration to the cloud.

# Bibliography


[Aa10] M. Armbrust, et al. “A view of cloud computing.” In: *Communications of the ACM* 53.4 (2010), pp. 50–58. ISSN: 0001-0782. DOI: 10.1145/1721654.1721672 (cit. on p. 13).

[Aa13] J. Alonso, et al. “Cloud modernization assessment framework: Analyzing the impact of a potential migration to Cloud.” In: Einhoven, Netherlands: IEEE, 2013, pp. 64–73. ISBN: 978-1-4673-4889-8. DOI: 10.1109/MESOCA.2013.6632736 (cit. on p. 15).

[Aa14] V. Andrikopoulos, et al. “CloudDSF - The Cloud Decision Support Framework for Application Migration.” In: *Proceedings of the Third European Conference on Service-Oriented and Cloud Computing*. Vol. 8745. Manchester, UK, 2014, pp. 1–15. ISBN: 987-3-662-44878-6 (cit. on pp. 9, 11, 20, 21).

[AB14] A. Ahmad, M. A. Babar. “A framework for Architecture-driven Migration of Legacy Systems to Cloud-enabled Software.” In: *Proceedings of the WICSA 2014 Companion Volume*. WICSA ’14 Companion. Sydney, Australia: ACM, 2014, 7:1–7:8. ISBN: 978-1-4503-2523-3. DOI: 10.1145/2578128.2578232. URL: http://doi.acm.org/10.1145/2578128.2578232 (cit. on pp. 15, 18, 19).

[ABLS13] V. Andrikopoulos, T. Binz, F. Leymann, S. Strauch. “How to Adapt Applications for the Cloud Environment.” In: *Computing* 95.6 (2013), pp. 493–535. DOI: 10.1007/s00607-012-0248-2. URL: http://dx.doi.org/10.1007/s00607-012%20-0248-2 (cit. on pp. 9, 14, 22).

[ASL13] V. Andrikopoulus, S. Strauch, F. Leymann. “Decision Support for Application Migration to the Cloud: Challenges and Vision.” In: *Proceedings of the 3rd International Conference on Cloud Computing and Service Science (CLOSER 2013)*. Aachen, Germany: SciTePress, 2013, pp. 149–155 (cit. on pp. 19, 20, 22).

[Ba12a] L. Badger, et al. “Cloud computing synopsis and recommendations.” In: *NIST Special Publication*. Vol. 800-146. 1. Gaithersburg, MD, USA: NIST, 2012. ISBN: 978-1-4776-2105-9. URL: http://csrc.nist.gov/publications/nistpubs/800-145/SP800-146.pdf (cit. on pp. 11, 12, 14).

[Ba12b] P. V. Beserra, et al. "Cloudstep: A step-by-step decision process to support legacy application migration to the cloud." In: *2012 IEEE 6th International Workshop on the Maintenance and Evolution of Service-Oriented and Cloud-Based Systems (MESOCA)* (Sept. 2012), pp. 7–16. DOI: 10.1109/MESOCA.2012.6392602. URL: http://doi.acm.org/10.1145/1851275.1851212 (cit. on p. 15).

[Bal15] M. Balduin. "Refinement and Extension of the Cloud Decision Support Framework for Application Migration to the Cloud." MA thesis. Institute of Architecture of Application Systems, University of Stuttgart, 2015, 155pp. (Cit. on pp. 10, 11, 14, 15, 24, 26, 30–32, 53, 81–85, 88, 92).

[BBG11] R. Buyya, J. Broberg, A. M. Goscinski. *Cloud Computing: Principles and Paradigms*. Wiley, 2011. ISBN: 978-0-470-88799-8 (cit. on pp. 11, 14).

[BH08] F. Burstein, C. W. Holsapple. "Handbook on Decision Support Systems." In: *1. International Handbooks on Information Systems*. Berlin, Germany: Springer, 2008. ISBN: 978-3-540-48712-8. DOI: 10.1007/978-3-540-48713-5 (cit. on p. 14).

[CA12] M. Chauhan, M. Alibabar. "Towards Process Support for Migrating Applications to Cloud Computing." In: *Proceedings of the 2012 International Conference on Cloud Computing and Service Computing (CSC)* (2012), pp. 80–87. DOI: 10.1109/CSC.2012.20 (cit. on p. 15).

[Clo13] Cloud Security Alliance. *The notorious nine: Cloud Computing Top Threats in 2013*. 2013. URL: http://downloads.cloudsecurityalliance.org/initiatives/top_threats/The_Notorious_Nine_Cloud_Computing_Top_Threats_in_2013.pdf (cit. on p. 13).

[Daw14] A. Dawson. "Decision Support for Application Migration to the Cloud." MA thesis. Institute of Architecture of Application Systems, University of Stuttgart, 2014, 157pp. (Cit. on pp. 10, 11, 14, 15, 20, 22, 24, 25, 29, 53, 89).

[FGE05] J. Figueira, S. Greco, M. Ehrgott. "Multiple Criteria Decision Analysis: State of the Art Surveys." In: *International Series in Operations Research & Management Science*. New York, NY, USA: Springer, 2005. ISBN: 0-387-23081-5 (cit. on p. 14).

[Fre14] S. Frey. "Conformance Checking and Simulation-based Evolutionary Optimization for Deployment and Reconfiguration of Software in the Cloud." PhD thesis. Faculty of Engineering, Kiel University, Department of Computer Science, CAU Kiel, 2014. ISBN: 978-3-7322-9734-4 (cit. on pp. 15, 17).

[Gar13] Gartner Inc. *Gartner says Cloud Computing will become the Bulk of new IT Spend by 2016*. 2013. URL: http://www.gartner.com/newsroom/id/2613015 (cit. on pp. 9, 14).

[GJN13] S. Gudenkauf, A. Josefiok M.and Goring, O. Norkus. "A reference architecture for cloud service offers." In: *17th IEEE International Enterprise Distributed Object Computing Conference (EDOC)*. 2013, pp. 227–236 (cit. on p. 89).

[HSS+10] M. Hajjat, X. Sun, Y.-W. E. Sung, D. Maltz, S. Rao, K. Sripanidkulchai, M. Tawarmalani. "Cloudward Bound: Planning for Beneficial Migration of Enterprise Applications to the Cloud." In: *SIGCOMM Comput. Commun. Rev.* 40.4 (Aug. 2010), pp. 243–254. ISSN: 0146-4833. DOI: 10.1145/1851275.1851212. URL: http://doi.acm.org/10.1145/1851275.1851212 (cit. on p. 15).

[IBM10] IBM. *Cloud Computing insights from 110 implementation projects, IBM Academy of Technology Survey*. 2010. URL: http://www-935.ibm.com/services/us/leveragingit/learnings_from_100_early_cloud_adopters.pdf (cit. on pp. 12, 13).

[Ins06] Institue of Electrical and Electronics Engineer, International Organization for Standardization, International Electrotechnical Commissions. *Software Engineering - Software Life Circel Processes - Maintenance: International Standard ISO/IEC 14764 IEEE Std 14764-2006.2nd ed.* New York, NY, USA, 2006. DOI: 10.1109/IEEESTD.2006.235774 (cit. on p. 12).

[Ja14] A. Juan-Verdejo, et al. "InCLOUDer: A Formalised Decision Support Modelling Approach to Migrate Applications to Cloud Environments." In: *40th EUROMICRO Conference on Software Engineering and Advanced Applications (SEAA)*. Verona, Italy: Conference Publishing Services, 2014, pp. 467–474. ISBN: 978-1-4799-5794-1. DOI: 10.1109/SEAA.2014.55 (cit. on pp. 15, 18).

[JAP13] P. Jamshidi, A. Ahmad, C. Pahl. "Cloud migration research: a systematic review." In: *IEEE Transactions on Cloud Computing* 1.2 (2013), pp. 142–157. ISSN: 2168-7161. DOI: 10.1109/TCC.2013.10 (cit. on pp. 9, 14, 15).

[Ka12] A. Khajeh-Hosseini, et al. "The Cloud Adoption Toolkit: supporting cloud adoption decisions in the enterprise." In: (2012), pp. 447–265. ISSN: 0138-0644. DOI: 10.1002/spe.1072 (cit. on p. 15).

[KPM11] KPMG. *The Cloud - Changing the Business Ecosystem*. 2011. URL: http://www.kpmg.com/AU/en/IssuesAndInsights/ArticlesPublications/pages/thecloud-changingthebusinessecosystem.aspx (cit. on pp. 13, 14).

[KPM12] KPMG. *Modelling the economic impact of Cloud Computing*. 2012. URL: http://www.kpmg.com/AU/en/IssuesAndInsights/ArticlesPublications/Documents/modelling-economic-impact-cloud-computing.pdf (cit. on pp. 12, 13).

[KWC98] R. Kazman, S. G. Woods, S. J. Carriere. "Requirements for Integrating Software Architecture and Reengineering Models: CORUM II." In: *Fifth Working Conference Reverse Engineering*. 1998, pp. 154–163 (cit. on p. 18).

[Ma13] A. Menychtas, et al. "ARTIST Methodology and Framework: A Novel Approach for the Migration of Legacy Software on the Cloud." In: *15th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC)*. Ed. by N. Björner, et al. Timisoara, Romania: IEEE, 2013, pp. 424–431. ISBN: 978-1-4799-3035-7. DOI: 10.1109/SYNASC.2013.62 (cit. on pp. 15, 16).

[Ma14] M. Menzel, et al. "CloudGenius: A Hybrid Decision Support Method for Automating the Migration of Web Application Clusters to Public Clouds." In: *IEEE Transactions on Computers*. 2014, pp. 1–14. DOI: 10.1109/TC.2014.2317188 (cit. on p. 15).

[MG11] P. Mell, T. Grance. "The NIST Definition of Cloud Computing." In: *NIST Special Publication. Gaithersburg, MD, USA: NIST 2011* 800-145.1 (2011). URL: http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf (cit. on p. 11).

[MR12] M. Menzel, R. Ranjan. "CloudGenius: Decision Support for Web Server Cloud Migration." In: WWW '12 95.6 (2012), pp. 979–988. DOI: 10.1145/2187836.2187967. URL: http://doi.acm.org/10.1145/2187836.2187967 (cit. on p. 15).

[Pow02] D. J. Power. *Decision Support Systems: Concepts and Resources for Managers. Westport, CT, USA*. Quorum Books, 2002. ISBN: 978-1-56720-497-1 (cit. on p. 14).

[Rig14] RightScale Inc. *State of Cloud Report*. 2014. URL: http://wwww.rightscale.com/2014-cloud-report (cit. on pp. 9, 13, 14, 36–38).

[Sa05] S. Strauch, et al. "Decision Support for the Migration of the Application Database Layer to the Cloud." In: *IEEE 5th International Conference on Cloud Computing Technology and Science (CloudCom2013)*. Vol. 1. Bristol, UK: IEEE Computer Society, 2005, pp. 639–646. ISBN: 978-0-7695-5095-4. DOI: 10.1109/CloudCom.2013.90 (cit. on p. 14).

[Sa11] B. Suleinman, et al. *On understand the economics and elasticity challenges of deploying business applications on public cloud infrustructure*. Journal of Internet Services and Applications 3, 2011. ISBN: 1867-4828. DOI: 10.1007/s13174-011-0050-y (cit. on pp. 12, 13).

[VA12] Q. H. Vu, R. Asal. "Legacy Application Migration to the Cloud: Practicability and Methodology." In: *IEEE Eight World Congress on Services*. Honolulu, HI, USA: IEEE, 2012, pp. 270–277. ISBN: 978-0-7695-4756-5. DOI: 10.1109/SERVICES.2012.47 (cit. on pp. 9, 12).

All links were last followed on November 25, 2016.

**Declaration**

I hereby declare that the work presented in this thesis is entirely my own and that I did not use any other sources and references than the listed ones. I have marked all direct or indirect statements from other sources contained therein as quotations. Neither this work nor significant parts of it were part of another examination procedure. I have not published this work in whole or in part before. The electronic copy is consistent with all submitted copies.

place, date, signature