Institute of Architecture of Application Systems
University of Stuttgart
Universitätsstraße 38
D–70569 Stuttgart

Master's Thesis Nr. 3572

Decision Support for Application Migration to the Cloud

Alexander Maximilan Darsow

Course of Study: Business Informatics

Examiner: Jun.-Prof. Dr.-Ing. Dimka Karastoyanova

Supervisor: Dr. Vasilios Andrikopoulos

Commenced: 5th September 2013

Completed: 5th March 2014

CR-Classification: D.2.7, D.2.11, H.3.4, H.3.5, H.4.2

Abstract

The Cloud Computing paradigm promises a major shift in providing computing resources and enterprises are encouraged to consider migrating existing applications to this new environment. In this regard various approaches of decision support for application migration to the Cloud exist to aid decision makers with this challenging multi-dimensional issue.
This Master's thesis considers the elaboration of a recent vision of a decision support system for application migration to the Cloud taking into account decisions to be made, and tasks that support decision-making. Based on a literature investigation this work constitutes a refined version of this approach by identifying several specific decisions and their relationships to other decisions and tasks. By means of a survey these extensions have been evaluated by peers in research and professional practice. Finally, a prototype based on current web technologies has been implemented to actually make the decision support approach available to decision makers considering application migration to the Cloud.

Table of Contents

List of Figures

List of Tables

1. Introduction

The Cloud Computing paradigm gained increasing attention and popularity in the field of information technology (IT) in recent years. The shift to provision computing resources like hardware capacities, storage, applications, and services in a virtual manner on-demand of the consumer instead of running internal data centers gained attention in the world of enterprises. On the one hand, resources are requested on-demand appropriate according to actual occurring workloads, and on the other those Cloud resources are charged in correspondence with their usage (i.e. pay-per-use). Due to the fact that actual costs associated with internal IT resources in terms of purchase, operation, and management are omitted Cloud Computing also enables a reduction of costs. However, downsides like security or availability issues inherent in Cloud Computing have also to be taken into account.

Nevertheless, analysts like Gartner ensure that "Cloud computing is set to have a considerable impact on business in the future" [1], which is also proven by the continued growth of the Cloud service market today, since enterprises consider *the Cloud* also in terms of productive applications rather than just for development or testing scenarios [2]. In this sense, a key interest of enterprises is to migrate their existing applications and even whole IT infrastructure resources from in-house to the Cloud [3] [4]. However, the approach of adopting Cloud Computing in terms of migrating existing applications can be difficult especially in case of business-critical applications, which have been developed over periods of time [5]. Adjustments to the application have to be faced with respect to the new environment (e.g. technical restrictions like new programming language versions, APIs, etc.) as well as conflicting aspects like related cost, performance qualities, security or legal concerns [6] [7]. Thus, due to such reasons some applications are not ready now, and they may never be in the future, to be moved to the Cloud at all [8].

Cloud service provider like IBM [9] or HP [10] address this issue by offering consultancy in Cloud Computing adoption and application migration beside their core Cloud services in order to assist enterprises in this challenging venture. Additionally, this field started to gain attention of research and in the meantime several approaches have been published to address the issue of migrating applications to the Cloud (see Section 2.3.3). However, the kind and degree of support differ: some approaches focus on migrating specific architectural parts of the application (e.g. web server) [11], or the migration to a certain kind of target Cloud environment (e.g. hybrid Clouds) [12], while others, for example, consider a more general view of identifying concerns of enterprises related to the Cloud migration [13] [14].

Jamshidi et al. [5] recently published a very comprehensive review of the State of the Art research in Cloud migration where existing research in this field has been identified, classified and compared to others. Based on an internal ranking by this review the approach of Andrikopoulos et al. [15] is assessed as one of the most related ones in this field. As one of the latest publications in this research field, [16] continues the work discussed in [15] and considers a vision of a *Decision Support System for Cloud Migration*. The approach addresses a more general view to support the Cloud migration rather than selecting a certain Cloud offering. It provides a conceptual view of which *decision points* have to be attended in the multi-dimension problem of application migration to the Cloud and which *tasks* are related to them. This work of Andrikopoulos et al. provides the basis this Master's thesis is build upon.

1.1. Problem Statement and Research Objectives

The previous section introduced how the research field of application migration to the Cloud has been developed. Furthermore, it pointed out the research approach of Andrikopoulos et al. as an important and also appropriate consideration for providing decision support. In the publication [16] the Decision Support System for Cloud migration is stated as a *conceptual vision* in which four major decision points with seven supporting tasks have been identified to address the multi-dimensional problem of application migration to the Cloud.

Since this work extends and refines the vision discussed by Andrikopoulos et al., and in order to avoid confusion to the reader, the conceptual view of the Decision Support System for Cloud Migration in [16] will be referred to as the *conceptual Decision Support Framework* (cDSF) in the following.

Currently, the State of the Art is missing an appropriate approach to provide decision support in application migration to the Cloud that addresses concerns of application developers as well as stakeholders in identifying important decision points like how an application is distributed, which scalability strategy is followed, how is multi-tenancy applied, and which service provider and offering is selected. The stated vision of Andrikopoulos et al. [16] approaches this issue and constitutes their future work. But to elaborate the cDSF to an actual decision support system its present elements like decision points and tasks need to be considered in more detail and further specified. For example the decision point about service provider and offering selection has to take into account different service and deployment models the Cloud Computing paradigm provides. Some of those suggestions on how decision points can be further considered have already been mentioned by the authors in their publication but an actual elaboration has not been conducted yet.

Based on the above, the main objective of this Master's thesis is focused on the *elaboration*, *refinement*, and *modeling* of the decision points and if necessary also tasks involved in the cDSF to improve this approach of application migration to the Cloud. This objective is subdivided into the following research objectives:

RO. 1 *Organize the State of the Art on application migration support and decision support systems in the context of Cloud Computing*

RO. 2 *Elaborate, refine, and evaluate the conceptual Decision Support Framework*

- *Investigate the cDSF based on the State of the Art and/or empirical techniques*
- *Extend the cDSF regarding completeness and suitability*
- *Model the decision points, extending decisions, tasks, and their relations to each other*

RO. 3 *Identify requirements for an implementation of the Decision Support Framework*

RO. 4 *Provide a prototypical implementation of the Decision Support Framework*

1.2. Thesis Outline

This Master's thesis follows the structure depicted in Figure 1.1. Chapter 1 introduces the topic to the reader and points out the problem statement accompanied by research questions this work intends to address. Chapter 2 provides an overview of related work beginning with basic terminology necessary in context of this work as well as the State of the Art (SotA) followed by an introduction of recent publications addressing the issue of application migration to the Cloud. This chapter ends with the presentation of the conceptual Decision Support Framework (cDSF) introduced as a conceptual view of a Decision Support System for Cloud Migration, which is the basis for this Master's thesis. Then, in Chapter 3, the main part of this thesis addresses the elaboration of the conceptual Decision Support Framework with first highlighting the process on how the elaboration is planned to be performed. Subsequently, the actual elaboration within the four decision points is described and additional tasks are pointed out. All elaboration and refinement work finally results in the elaborated Decision Support Framework (eDSF). Based on this, Chapter 4 shows the process and the findings of the eDSF evaluation conducted by a questionnaire-based survey. Regarding the realization aspect of this Master's thesis, Chapter 5 illustrates how the eDSF is prototypically implemented by means of a web-based application to provide an actual decision support system. Finally, this work concludes in Chapter 6 by answering the initially stated research questions and briefly discussing future work.

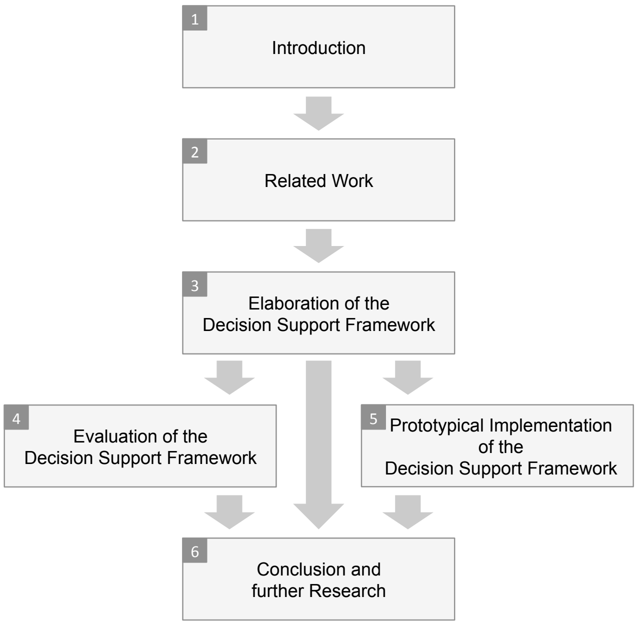


Figure 1.1.: Thesis Outline

2. Related Work

This chapter introduces the main terminology related to this thesis for a common understanding. This is followed by recent publications in the field of application migration to the Cloud and finally the research basis this thesis is built on is presented to the reader.

2.1. Cloud Computing

Cloud Computing describes a paradigm shift in provisioning computing resources like hardware capacities, storage, and applications. Instead of running such resources cost intensively on-premise they are provided on-demand of the consumer via networks (i.e. most commonly the internet) and only charged with regard to their actual usage [17] [18]. This approach promises a solution for the long-time ambition of turning computing in a utility such as water or electricity [18] [19]. Thereby Cloud Computing refers to both; the services delivered, as well as the hardware and software in the datacenter, which actually provide them and often called a *Cloud* [18].

Especially, in its beginnings, many different definitions have been raised [17] [18] [20] [21] with each of them focusing on different characteristics, aspects, and perspectives of the whole topic. In 2011, the National Institute of Standards and Technology (NIST) provided its definition for Cloud Computing:

> *„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.“* [22]

Given the fact that this definition is one of the clearest, most comprehensive ones, and in addition to that widely applied by governments and professionals [23] [24], this Master's thesis will also adopt it.

With further reference to the NIST definition, Cloud Computing is composed of five essential characteristics and mainly subdivided into three service models and four delivery models [22].

Essential Characteristics

- *On-demand self-service*: Computing capabilities such as hardware capacities, server, and storage are individually provisioned as needed to the consumer in an automatically manner without any human interaction.

- *Broad network access*: The provisioned computing resources are available for access over a network, e.g. the Internet, and based on standardized mechanisms and technologies.

- *Resource pooling*: On the provider's side computing resources are organized in resource pools to provision several consumers by using a *multi-tenant* model. Various physical and virtual resources are (re-)assigned dynamically depending on consumers demands. With pooling various resources the consumer has no control or knowledge of the detailed physical resources location in general, but in some cases a location specification (e.g. selecting a certain country or datacenter) might be possible.

- *Rapid elasticity*: Depending on the consumer's demands computing capabilities can be flexibly provisioned or released. This might be performed in an automatic way so that rapid scaling possibly leads to the illusion of unlimited resources to the consumer.

- *Measured Service*: A kind of cloud management automatically controls and optimizes resource usage by leveraging metering capabilities (e.g. pay-per-use or charge-per-use) appropriate to the type of service. Therefore, in terms of transparency for both provider and consumer the utilized resources can be monitored, controlled and reported.

Service Models

- *Software as a Service* (SaaS): The service provided to the consumer is an on-demand software application that can be accessed from different kinds of client devices, e.g. the web browser. The underlying infrastructure or individual application capabilities are not managed or controlled by the consumer. Only very limited user-specific configuration settings of the application might be possible. Examples for business SaaS are salesforce Sales Cloud[1] or SAP Business One Cloud[2].

- *Platform as a Service* (PaaS): The consumer is provided with a service to deploy consumer-created or acquired applications onto a provisioned Cloud infrastructure using programming languages, libraries, services, and tools supported by the provider. The consumer fully controls deployed applications and eventually limited configurations of the provided application environment, but has no management or control privileges in terms of the underlying Cloud infrastructure. Business examples for such services are SAP HANA Cloud Platform[3] and Microsoft Windows Azure[4].

- *Infrastructure as a Service* (IaaS): In this model the service provided to the consumer is the IT infrastructure sharped in provisioning processing, storage, network and other fundamental computing resources. The management or control of the underlying infrastructure is not accessible to the consumer but everything above this level like operating systems and deployed applications is controllable. Business related IaaS examples are Amazon EC2[5] or IBM SmartCloud[6].

Beside this most common differentiation of service models, the fast evolution of Cloud Computing exposes frequently new Cloud services and terminology like *Communication-as-a-Service* (CaaS), *Human as a Service* (HaaS), more generally consolidated under the term *Everything as a Service* (XaaS) [25] [26]. Since they are not part of the NIST definition, these models are solely mentioned for completeness.

Delivery Models

- *Private Cloud*: The Cloud infrastructure is exclusively provisioned for a single organization and its consumers (e.g. employees, business units). However, ownership, management and operation might be performed by the organization itself, a third party, or some combination of them. Regarding this, the cloud can exist on or off premises.

[1] salesforce Sales Cloud: http://www.salesforce.com/de/sales-cloud/overview/
[2] SAP Business One Cloud: http://www.sap.com/pc/tech/cloud/software/small-business-management-cloud/index.html
[3] SAP HANA Cloud Platform:http://www.sap.com/pc/tech/cloud/software/hana-cloud-platform-as-a-service/index.html
[4] Microsoft Windows Azure: http://www.windowsazure.com/en-us/
[5] Amazon EC2: http://aws.amazon.com/en/ec2/
[6] IBM SmartCloud: http://www.ibm.com/cloud-computing/us/en/iaas.html

- *Community Cloud*: The provisioned Cloud infrastructure is for exclusive use of a organizational community which is sharing a specific concern (e.g. mission, policy, security requirements, etc.). Ownership, management, and operation may be performed by one or more organizations of the community, a third party, or some combination of them. It may exist in an on or off premises manner.

- *Public Cloud*: The Cloud infrastructure is open for public use and usually shared by several public consumers. It is managed, operated and owned by a business, academic, or government organization, or a combination of them and it exists on premise of the cloud provider.

- *Hybrid Cloud*: The provisioned Cloud infrastructure is a combination of two or more delivery models mentioned before that remain unique, but are connected through standardized or proprietary technology.

Cloud Actors

According to the NIST definition different *actors* are involved on the supplier as well as on the receiver side while using any kind of Cloud service. The supplier refers to the provision of the service in terms of three perspectives *ownership*, *management* and *operation*. In principle, this leads to three individual actors (i.e. *Cloud owner*, *Cloud manager*, and *Cloud operator*), but more usually one covers several responsibilities — for example a cloud owner is in charge of management and operation efforts [22]. Since this distinction is fuzzy in case of public Clouds because a services provider may spread efforts over more than one party which has not necessarily be obvious to a service consumer, we will use the term *Cloud provider* for one party that provides Cloud services in general.

Considering the receiving side of Cloud services the above mentioned definition uses the term *consumer* whereby others like [18] talk about users. We will stick to the term *Cloud consumer* for the party that receives any kind of service capabilities provided by the Cloud. Further distinction on the receivers side is not made, although varying responsibilities especially in terms of enterprises could be distinguished, e.g. payment, management or orchestration of services.

Benefits and Risks for Enterprises

Adopting Cloud Computing in enterprises is associated with several benefits and opportunities, most of them related to cost reduction [8]. But nonetheless also risks and challenges have to be considered [18]. Both can be distinguished into technical and non-technical related aspects.

One major technical benefit can be derived from the NIST definition and its characteristics in terms of rapid elastic provisioning of perceived infinite resources on-demand of the Cloud consumer. Furthermore, with delivering Cloud services over broad networks and standardized interfaces will lead to more flexibility and efficiency in the IT department. In contrast, risks concerning for example service availability, performance, data lock-in, data privacy, etc. arise and have to be addressed properly before any Cloud adoption approach [3] [18].

At a first glance, enterprises obviously focus more on their business, thus, on non-technical aspects rather than technical ones. Perhaps the most common benefit associated with the adoption of Cloud Computing is cost reduction [17]. Reducing or even replacing physical on-premise computing infrastructure and their associated up-front and operating costs by receiving computing capabilities on demand based on a pay-per-use payment model is realized by converting capital expenses (CAPEX) into operational expenses (OPEX) [8] [18] [20] [27]. However, since the usage of Cloud services has evolved over the years and today core business operations and personal data may be affected, it is of significant importance to address for example legal and confidential issues [28].

Taking into account the overall benefits and risks in terms of Cloud Computing it is important to be aware of the relationship between technical and non-technical aspects. For example, elasticity and on-demand resource provisioning can lead to cost savings over running traditional datacenters but in contrast the multi-tenant characteristic possibly raise data privacy and confidential concerns. Cloud providers try to ease these concerns of Cloud consumers by ensuring Service Level Agreements (SLAs) related to specific characteristics or performance key figures of the provided service [20].

To sum up, the Cloud Computing paradigm can provide comprehensive advantages but it also raises a series of challenges to be considered by a Cloud consumer. And may be not every enterprise or their specific applications are appropriate to be moved to the Cloud.

2.2. Application Migration to the Cloud

In most of today's enterprises software systems (i.e. applications) are supporting core business processes with essential functionalities like cost analysis, shipping tracking, or accounting and the fact those systems probably have been evolved and adjusted over a long time make it a tough and very limited consideration of changing them. According to the software maintenance standard of the International Organization for Standardization (ISO) and the International Electrotechnical Commission (IEC) [29], the term adaptive maintenance is defined as "the modification of a software product, performed after delivery, to keep a software product usable in a changed or changing environment". In these terms software migration or retirement are two possible conclusions of the ISO/IEC maintenance process circle and application migration can be viewed as a special case of adaptive maintenance [29]. Thus, to keep existing applications capable for development and future-proof they are required to be migrated to efficient and up-to-date architectures [30].

In general, such a „Software migration is viewed as a transformation of software systems into another or new target environment, without changing its functionality" [30]. But once all functionalities of the source system are successfully shifted to the new target environment, it is possible to consider also enhancements and additional functionalities to address further business needs [31].

Transferring this general perspective of application migration to the context of Cloud Computing basically means "[...] to port an application from a local data center to a selected Cloud platform with no changes in functionalities or compromises in performance" [7]. However, in most migration cases at least some application modifications are needed to accomplish a sufficient and appropriate adaptation to the cloud [32], for example an application migrated to an IaaS solution may need adjustments to utilize elasticity in resource provisioning.

Drivers for Migration to the Cloud

Interests of enterprises are aligned to do best business the most efficient way and in terms of IT departments this is often associated with lowering their costs. Given the overall promising benefits of Cloud Computing identified in Section 2.1, it is not surprising the major driver for enterprises is an expected reduction of costs [28] [32] [33]. This mainly entails serving existing workloads while lowering expenses in infrastructure investment, maintenance and operation in terms of computing resources [32].

Scalability is another significant advantage enterprises aim for, because rapid and elastic provisioning is one of the main characteristics of Cloud Computing (see Section 2.1) promises more responsiveness and flexibility for the IT department and for the whole organization as well [6] [32] [33] [34]. Beside these two primary enterprise drivers for Cloud migration other drivers also include green computing through reducing energy consumption [32] or optimize business processes to improve overall quality for their services and products [6].

A current review of Cloud migration research in [5] has also identified a list of drivers for Cloud migration emphasizing cost savings and scalability as the most common drivers for enterprises as mentioned above.

Despite, all attraction to Cloud Computing benefits its associated risks depicted in Section 2.1 have to be considered beforehand and in practice many organizations struggle while considering a Cloud adoption approach [6] [33]. For instance, an existing application that violates environmental constraints of the Cloud provider suddenly turns the Cloud migration into an inherently complex issue [6].

Cloud Migration Types

Therefore, in case of moving enterprise application to the Cloud the advised types of candidate applications are related to pilot projects, non-essential tasks, development and test or solely the compensation of peak workloads [33]. Arguably the most accessible kind of Cloud application migration for enterprises is moving to IaaS, which offers fast and convenient benefits because this allows a Cloud migration without any kind of major changes to the application [35]. New Cloud services like Databases as a Service, however, may grant more flexible and individual approaches of application migration on the level of single application layers or even components. In [15] Andrikopoulos et al. investigated how the business and data layer of common three layered enterprise applications can be migrated to the Cloud. The authors identified different *migration types* discussing how an application migration to the Cloud can be performed. Figure 2.1 shows those migration types, which are introduced in the following.

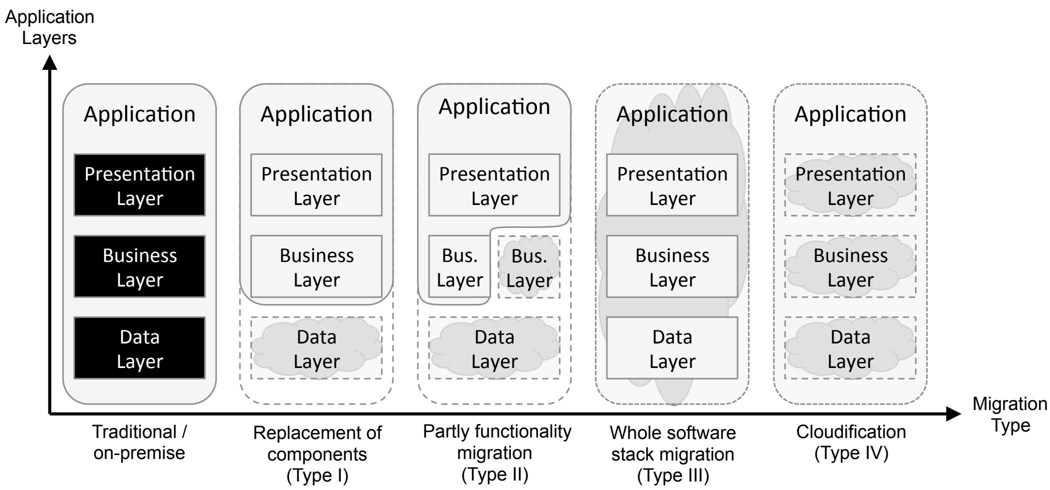


Figure 2.1.: Cloud Migration Types [15]

- *Type I - Replacement of components*: In this case, one or more components of an application are substituted by Cloud services. According to the authors this "is the least invasive type of migration" where activities such as configuration, rewriting and adaption accrue to cope with possible migration incompatibilities. As an example the move from a local MySQL database to a Google App Engine Datastore is stated.

- *Type II - Partial functionality migration*: In this case whole application functionalities are moved to the Cloud, which may involve several architectural components on one or more application layers. An example for this case is the combination of an Amazon SimpleDB and EC2 instances to host data and business logic of an application.

- *Type III - Whole software stack migration*: This case considers the migration of a whole software stack where an existing application is encapsulated in a virtual machine (VM) and then deployed on a IaaS Cloud provider. As mentioned before this is assumed as the most accessible migration approach to the Cloud for enterprises [15] [35].

- *Type IV - Cloudify the application*: In this type of Cloud migration the application is completely re-constructed by implementing the applications functionalities out of a combination of cloud services. Also, like in migration type I, this scenario needs the migration of data and business logic to the Cloud and further actions to cope with possible incompatibilities.

2.3. Decision Support for Application Migration to the Cloud

2.3.1. Decision Support

Each of us is faced to make decisions in various situations in daily life. Thereby, every decision contains a guess about the future and the way of either solving a problem or achieving a goal is to estimate a certain situation in terms of anticipating possible actions to finally achieve a desired objective [36]. The means of decision-making is a non-random process of selecting a possible course out of the alternatives available [36] [37]. As decision-making gains in complexity with an increasing number of considered elements and relationships, this in turn requires today's enterprise decision-making processes to be as efficient as even possible [36].

In the majority of such situations a *decision maker* (DM) is confronted with multiple criteria when judging possible alternatives [37]. Such problems composing multiple criteria are denoted as *Multiple Criteria Decision Making* (MCDM) problems [37] [38]. Referring to [38], characteristics of MCDM comprise fundamental components like a finite or infinite number of alternatives (i.e. potential actions, solutions, courses, etc.), at least two criteria, a stated problematic, and at least one DM. Furthermore, the problematic can either be a choice problematic, sorting problematic or ranking problematic and there are several known methods to address these different kinds of MCDM problems [38].

In case of considering decision support for application migration to the Cloud the problem we face will probably be a MDCM choice problematic in order to select the Cloud solution best suitable for the application. To address these MCDM problems there are two major approaches in the field, namely multi-attribute utility and value theories (e.g. the Analytical Hierarchical or Network Process) and outranking methods (e.g. ELECTRE or PROMETHEE Methods), which are briefly introduced in the next section.

However, the objective of this Master's thesis is not to determine a certain Cloud solution but rather to point out decisions, tasks, and their relationships in terms of an application migration to the Cloud. Hence, decision support in this case is provided by an effective representation of MCDM data (i.e. decisions, tasks, and relationships) for the decision-making process as emphasized by Ward et al. in [39].

Analytical Hierarchical and Analytic Network Process

The *Analytic Hierarchy Process* (AHP), introduced by Saaty in [40], presents an approach to solve multiple criteria decision-making when its criteria are hierarchically structured. The structure is ordered in descending levels from the overall goal of the decision making process at the top down to criteria, subcriteria and alternatives [40]. The process uses an absolute scale of judgments to reach a relative measurement based on pairwise comparisons of all criteria to determine a decision either by a single value indicating the best outcome or by a priority vector representing a ranking of possible outcomes [40] [41].

However, most real world decision problems usually do not fit in a simple hierarchical and top-down-oriented structure, because criteria and subcriteria are partly related on different hierarchical levels and even alternatives themselves possibly affect criteria as well [41] [42]. Therefore, Saaty introduced the *Analytic Network Process* (ANP), which is a generalization of the AHP and represents a network structure of nodes, clusters and loops [41] [42]. ANP is mainly based on two parts: a control network of criteria, subcriteria, and alternatives representing the flow of criteria influences with the help of source and sink notes indicating either the origin or the destination of influence [42], and a second network (the supermatrix) indicating influences among the various elements [43]. Furthermore, ANP relates to the fundamentals of AHP and its approach of pairwise comparison judgments to determine a decision [42].

Both approaches and their different structure of criteria, subcriteria and alternatives are depicted in Figure 2.2.

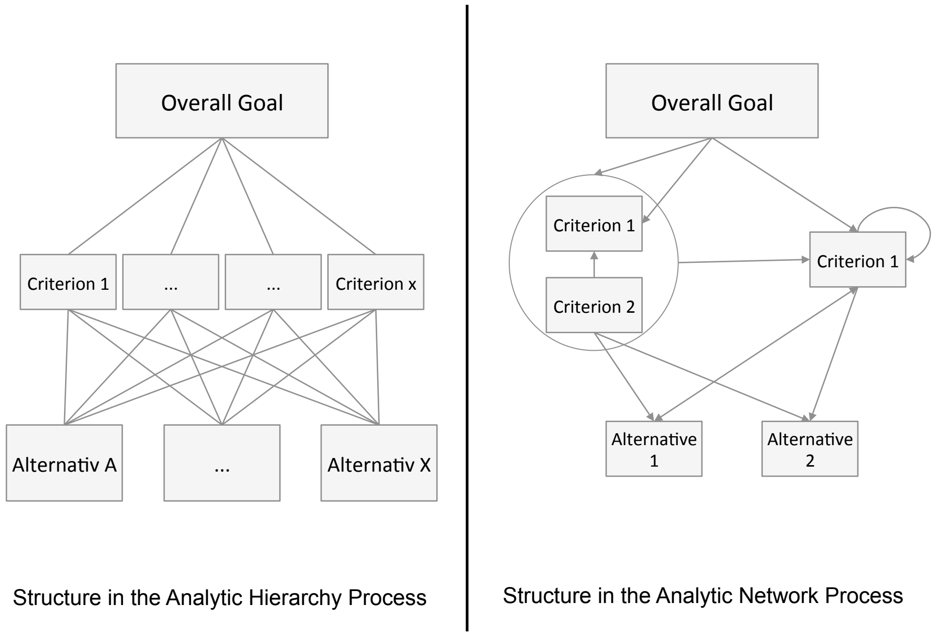


Figure 2.2.: Criteria Structure of AHP vs ANP [42]

ELECTRE methods

ELECTRE (ELimination Et Choix Traduisant la REalité - ELimination and Choice Expressing the REality) methods describe a methods family aiding MCDM problems based on outranking relations on the set of alternatives [38] [44]. The origin research approach (ELECTRE I) goes back to 1965 dealing with a real world multiple criteria problem of choosing between marketing activities of an enterprise. Based on other real world problems ELECTRE II was introduced as the first method of the family designed especially to address ranking problems. All ELECTRE methods focus on two main procedures: First, the construction of one or more outranking relations and second, an exploitation procedure. Through the exploitation procedure recommendations are elaborated with respect to results of the first procedure. Depending on the addressed problematic (choosing, ranking or sorting) determined recommendations aid the decision maker. For all three mentioned problematics individual ELECTRE methods are designed [44]. While these methods were fundamental for other outranking methods (e.g. PROMETHEE) developed in the past, ELECTRE methods reveled a widespread attention in operational research or in areas like water management, finance, transportation and project selection [38].

2.3.2. Decision Support Systems

A *decision support system* (DSS) is an information system (IS) to support the decision-making process of humans by utilization of data and models [45] [46]. Through the assistance of a computer-based system a decision maker is able to be more productive, agile, innovative, and reputable on his decisions made [47]. Within the history of information systems, DSS are described as one of the core elements in the IS field which extend and step far beyond management information systems [36]. Basically, in context of decision support systems the following five types of systems can be distinguished [45]:

- *Model-driven DSS*: Models with limited data and parameters provided by the DM regarding financial, optimization, and/or simulation are used, e.g. in production planning management DSS

- *Data-driven DSS*: This type access and manipulated a time series of internal and external company data and also real-time data, like for example Data warehouses systems

- *Communication-driven DSS*: By using network and communications technologies these type of DSS enable support in decision-relevant collaboration and communication, for example in groupware or video conferencing systems

- *Document-driven DSS*: These DSS provide support based on retrieving documents like scans, images, videos, and webpages and then analyzing them for relevant information

- *Knowledge-driven DSS*: Such a DSS supports by suggesting and recommending actions to the DM referring to knowledge about the problem domain

The knowledge-driven DSS uses a specialized knowledge base for problem-solving which refers to a particular domain, understanding of problems within this domain, and certain skill in solving problems of this kind [45]. The actual support of the system is represented by a suggestion or recommendation of actions to the DM [45]. In terms of decision support of application migration to the Cloud in case of this Master's thesis, a knowledge-driven DSS addresses the stated problem of representing a given knowledge base, i.e. in the end the elaborated Decision Support Framework. In addition, this DSS type would also be appropriate in order to determine a certain Cloud provider of an application migration.

Architecture of a Decision Support System

With reference to Holsapple [48] a DSS architecture in general consists of four essential components as depicted in Figure 2.3.

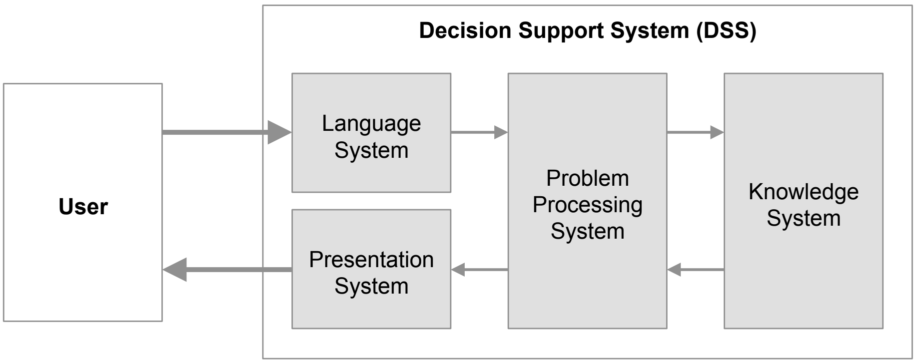


Figure 2.3.: Generic Decision Support System Architecture [48]

First, the *Language System* comprises all messages, which are understood and accepted by the DSS. Second, the *Presentation System* represents all messages the DSS emits and sends to the user. Third, the *Knowledge System*, which contains data the DSS establishes its decision-making on with respect to the DSS type (e.g. model-driven, knowledge-driven, etc.). The fourth element, the *Problem Processing System*, is the key component recognizing and solving the decision by utilizing particular decision-making methods or algorithms. The system typically interacts with users like DM, administrator or developer.

Sprague and Carlson [49] suggest a DSS architecture more related to an information system architecture (e.g. the three layer pattern [50]) and is also referred as *the* architecture of DSS [48]. Their model shows three basic components, a database, a model base and a user interface, illustrated in Figure 2.4.

Regarding [49] the database comprises information and data required to perform analysis and process the actual decision-making method. These methods or decision analysis tools

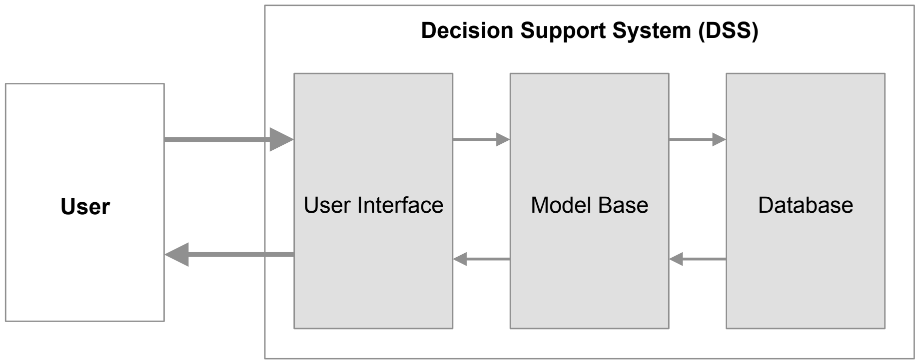


Figure 2.4.: Information System Components of a Decision Support System [48]

are deposited in the model base, which is directly related to the database for performing data queries. The third component, the user interface, is a key component because it represents the actual implementation of the system and handles all communication and interaction between user and DSS.

With special focus on DSS in the field of application migration to Cloud Computing environments, Andrikopoulos et al. [51] identified system requirements based on existing literature and available tools which are applicable on actual Cloud service or provider decision-making.

- Ability to match user-provided requirements with available Cloud service provider offerings and calculate the cost of using each offering for a given period.

- Ability to rank the proposed offerings based on different criteria beyond cost.

- Ability to support variable requirements in terms of computational resources over periods of time to better match the varying demands of the users.

- Existence of a knowledge base for Cloud service providers containing the information regarding their offerings and pricing models.

- Availability of a user-friendly interface that allows user to navigate the system easily.

2.3.3. Decision Support for Application Migration to the Cloud

The consideration of migrating existing applications to Cloud Computing environments receives increasing attention by enterprises. This non-trivial challenge of selecting the most appropriate solution fulfilling individual needs is addressed by commercial consultancy business as well as research whose decision support approaches meanwhile go beyond simple cost comparison.

Table 2.1 gives an overview of current research approaches supporting the enterprise decision-making of application migration to the Cloud. Those approaches have been considered highly relevant in terms of this Master's thesis and will be introduced in the following.

(Name), Author(s), Title	***Year***	***Reference***
„Cloudward Bound", M. Hajjat, X. Sun, Y. E. Sung, D. Maltz, S. Rao, K. Sripanidkulchai, M. Tawarmalani, „Process for Migration Software Applications to the Cloud"	2010	[12]
„The Cloud Adoption Toolkit", A. Khajeh-Hosseini, D. Greenwood, J. W. Smith, I. Sommerville, „The Cloud Adoption Toolkit: supporting cloud adoption decisions in the enterprise"	2012	[13] [14]
„Cloudstep", P. V. Beserra, A. Camara, R. Ximenes, A. B. Albuquerque, N. C. Mendonça, „Cloudstep: A Step-by-Step Decision Process to Support Legacy Application Migration to the Cloud"	2012	[6]
„CloudGenius", M. Menzel, R. Ranjan, „CloudGenius: Decision Support for Web Server Cloud Migration"	2012	[11][52]
M. A. Chauhan, M. A.Babar, „Towards Process Support for Migrating Applications to Cloud Computing"	2012	[53]

Table 2.1.: Current Approaches in Decision Support for Application Migration to the Cloud

Cloudward

The *Cloudward* framework was developed by Hajjat et al. [12] in collaboration between academic and corporate business research aiming to migrate enterprise services to hybrid Cloud environments. Cloudward addresses the enterprise challenges of whether migrating a certain application is beneficial at all, and if, how to consider which application components should be migrated in order to achieve the best solution for defined objectives. In addition, Cloudward also devotes to how security policies have to be reconfigured to ensure correctness after the application, or single components, is migrated.

As a result of today's high complex enterprise applications the framework basically considers a hybrid migration approach where application layers and single application components at those layers are migrated from a local data center to the Cloud. The designated hybrid environment consists of on-premise elements inside the enterprise local data center and a single Cloud data center. To explore benefits of the hybrid migration approach, Cloudward takes into account aspects like cost savings, communication costs, transaction delays, and general enterprise constraints (e.g. data privacy and security).

The framework part concerning support decision-making is based on constructing an optimization problem which corresponds to an integer programming problem which is formulated to maximize migration benefits. On the other hand, an algorithm for automatic and insurable reconfiguration of reachability policies for the migrated application based on access control lists is part of the Cloudward framework.

The approach described by Cloudward pays special attention to real world enterprise application architectures and supports decision-making on how they can be migrated most beneficially to a hybrid environment. The framework was applied and evaluated based on real applications shown in case studies [12] and limitations like considering just a single Cloud data center are targeted for future work.

The Cloud Adoption Toolkit

The Cloud Adoption Toolkit was introduced by Khajeh-Hosseini et al. [13] and shows a framework supporting decision makers in adopting Cloud Computing within their organization. Its main approach is to identify distinct concerns of enterprises by exploring various areas in terms of Cloud adoption and match them to techniques and tools provided by the framework to address them appropriately. As a result of processing the framework particular requirements for the individual enterprise Cloud adoption should be provided. Furthermore, the framework is not limited to a special kind of existing application because it assesses its general suitability for moving to the Cloud. Different perspectives consider aspects like costs, technical, non-technical and organizational influences as well as socio-technical effects with impact on the migrated application solution.

The framework is divided into four components that are meant to be used in sequence. At the beginning, *Technology Suitability Analysis* assesses the general suitability of Cloud

Computing technology for the existing application. For this purpose the framework provides a checklist with questions related to technical and organizational circumstances to explore obstacles, which increase risks or even hinder the Cloud adoption at all. For instance, elasticity and availability aspects as well as security and confidential issues are considered.

Next, *Cost Modeling & Energy Consumption Analysis* and *Stakeholder Impact Analysis* are performed subsequently on one level (i.e. they can be performed simultaneously). The Cost Modeling part is the origin and also the most mature tool in the whole toolkit and a free online implementation[7] is provided. While Cost Modeling takes into account different scenarios (e.g. public or hybrid Cloud solutions) and corresponding future system demands, Energy Consumption Analysis focuses on determining an optimal energy consumption limited to the perspective of an own private Cloud infrastructure. The later component Stakeholder Impact Analysis addresses decision makers with assessing the organizational fit of the Cloud system in terms of its socio-political impact. Related benefits and risks are exposed by investigating the Clouds impact on stakeholders work activities (e.g. processing time), social factors (e.g. employee interest and satisfaction), and political factors like fairness and reasonability within the decision making procedures). By this operational feasibility of using the Cloud should be ensured.

Within the part of *Responsibility Modeling* the Cloud systems responsibilities are considered on how systems operation, maintenance, and management is spread over entities like enterprise departments and service providers. This should also address socio-political acceptance of the future Cloud solution.

Depending on individual initial situations of an enterprise, framework parts can be utilized if they are relevant in a certain case of application migration like it is demonstrated in a case study [13].

Cloudstep

In their paper [6], Beserra et al. present a decision process supporting the migration of legacy applications into the context of Cloud Computing. This step-by-step decision process is called *Cloudstep* and tries to fill the gap of a missing general process for migrating existing applications to the Cloud. Thereby, the main objective is the identification and analysis of key characteristics regarding Cloud selection and possible migration tasks.

[7] PlanForCloud: http://www.planforcloud.com

In general, Cloudstep aims to support the organizational point of view as well as application developer with a process consisting of nine defined activities, which utilize mainly three template-based profiles. Profiles are created for the organization, the legacy application and one for each candidate Cloud provider. Each profile gathers information through questions regarding its particular subject, e.g. the organization profile questions the motivation for the Cloud migration or the current allocation of computing resources and further legal and administrative aspects. For all three types of profiles a number of questions are given as guidance for their individual definition.

Then, these profiles are cross-analyzed iteratively on different process levels to reveal constraints in terms of the Cloud migration purpose. The exposed constraints can further be categorized in seven main areas regarding financial, organizational, security, communication, performance, availability and suitability constraints, which depend on an hierarchically structure. After evaluating all identified constraints several outcomes are possible: In case of non violating constraints a migration is recommended and the next step is to choose an appropriate migration strategy. Otherwise, either the problematic constraints have to be addressed in the legacy application context, or a new candidate Cloud provider has to be examined, or in case irresolvable constraints the migration purpose has to be aborted.

In conclusion, Cloudstep illustrates a step-by-step decision process for the migration of legacy applications to the Cloud. Both, technical and non-technical aspects are addressed based on gathering information those created profiles which are cross-analyzed to reveal constraints that might affect the overall Cloud migration purpose. Although, the process steps are well-structured, the creation of detailed profiles, their structure and exact questions are only broadly defined.

CloudGenius

In 2012 Menzel and Ranjan introduced in [11] a framework called *CloudGenius*, which claims to support an automated process of decision-making for web server migration to the Cloud. The framework mainly addresses web engineers supporting them to select the right combination of IaaS Cloud offering and Cloud virtual machine (VM) image for a single-tier web application migration.

The process is initiated based on the known decision of moving a web application to an IaaS Cloud and describes a straightforward designed procedure starting with requirements definition. In this step a web engineer is required to define requirements (numerical and non-numerical) regarding the expected Cloud infrastructure (i.e. the required web server characteristics). Next, input is required in terms of performance goals regarding specific criteria of the Cloud VM and IaaS Cloud offer which are gathered based on selecting and weighting the criteria. All inputs are processed and initially stored in the frameworks database, which will be the source for following process steps. Those are filtering and evaluating VM images and Cloud infrastructure services separately by leveraging the decision-making framework $(MC^2)^2$ [52].

$(MC^2)^2$ is an evaluation method of multi-criteria decision-making problems. Contrarily to the MCDM method ANP suggested in the original $(MC^2)^2$, CloudGenius uses the AHP method (see Section 2.3.1), which has a lower complexity in building the hierarchical weighted criteria structures for the two evaluation objects (i.e. Cloud VM and IaaS Cloud offering).

After evaluating VM images and IaaS Cloud offering, combined solutions of images and services are built which are finally presented ranked by a value for each VM image and offering combination influenced by user-defined weights. Further process steps consider preparation of the existing web server and application for deployment and migration, execution of the software deployment and configuration and a final customization of the designated solution. If the elaborated web server configuration is not satisfying at this point the framework loops back to the solution selection step or even back to the very beginning of defining requirements. However, if the elaborated solution is satisfying the last step is to plan and execute the migration strategy.

The CloudGenius framework presents a generic process approach to support decision-making in application migration to the Cloud. Caused by its pre-defined focus on solely single-tier web applications and IaaS Cloud offerings up-front efforts and decisions have to be made by enterprises before the framework can be used beneficially. In addition, the framework almost entirely regards to technical requirements of the web server and lacks in considering organizational consequences and enterprise challenges. The usage of AHP is adequate for the presented elaboration, but not sufficient for more comprehensive decision support since much more criteria and possible interconnections have to be considered.

Process Support for Migrating Applications to Cloud Computing

In their 2012 paper [53], Chauhan and Babar present a process framework to support the migration of software applications to the Cloud. The seven-step process is elaborated based on considering concepts for migrating legacy systems to Software Oriented Architecture (SOA) as Cloud Computing arguably embraces many of its characteristics. By extracting insights of SOA methodologies and enhancing them to address specific key issues regarding the migration to the Cloud, like individual requirements for Cloud environment and influencing characteristics of different Cloud services, this process attempts to provide comprehensive guidance to evaluating the most appropriate target Cloud environment.

First, general business requirements are identified by questioning the migrations main motivation and objectives, which are purposed to be achieved. Further, these requirements are subdivided into more detailed requirements (e.g. functional requirements) and analyzed in either a qualitatively or quantitatively manner. As a result, this step provides a set of requirements processed by business analysts.

Next, potential Cloud environments are taking into account, determining their specific features and explore their appropriateness to previously stated requirements like for example data confidentiality, sensitivity, or budget restrictions. At this stage project managers and system architects provide a list of potential Cloud environments and their features.

Given the potential Cloud environments their compatibility with the existing application is analyzed and a trade-off analysis is provided for each potential Cloud environment assessed by system analysts and architects.

In a more detailed manner the next step investigates each potential Cloud environment with regards to functional and qualitative requirements satisfaction to emphasize the best architecture solutions. The results are high-level designs of potential architecture solution elaborated by system architects.

At the same stage potential Cloud environments and their specific quality features (e.g. multi-tenancy or interoperability) are analyzed. The result is a trade-off analysis of supported quality requirements by the potential Cloud environments.

Results of the three previous steps are then cross-analyzed to evaluate the Cloud environment which satisfies requirements of the purposed target environment best. In case of any incompatibilities the two previous two steps may be reprocessed. The outcome describes the finalized design of the purposed system architecture.

In the final step the elaborated system is implemented and deployed on the desired Cloud environment by system architects and developers.

The introduced process to support migrating applications to the Cloud offers an approach, which is not specialized on certain legacy applications nor Cloud architectures and mentions both technical and organizational related requirements. However, at its current state considered process steps are only described in a very general manner and even the provided case study [53], examining the movement of an open source application (Hackystat[8]) to two cloud environments, lacks in stating more concrete information (e.g. metrics, detailed requirements) for each step. Hence, an actual decision maker is only superficially supported and has to invest vast effort to tune this process for his needs.

2.4. The Conceptual Decision Support Framework for Application Migration to the Cloud

The previous section introduces recent approaches considering decision support for application migration to the Cloud, which can be referred to as State of the Art work of this field of research. One of the latest works, however, is a conceptual view of a Decision Support System for Cloud Migration by Andrikopoulos et al. [16] expressing challenges and a vision on how decision support in context of application migration to the Cloud can be provided. This, in terms of this Master's thesis so-called, conceptual Decision Support Framework (cDSF) is depicted in Figure 2.5 and introduced in the following.

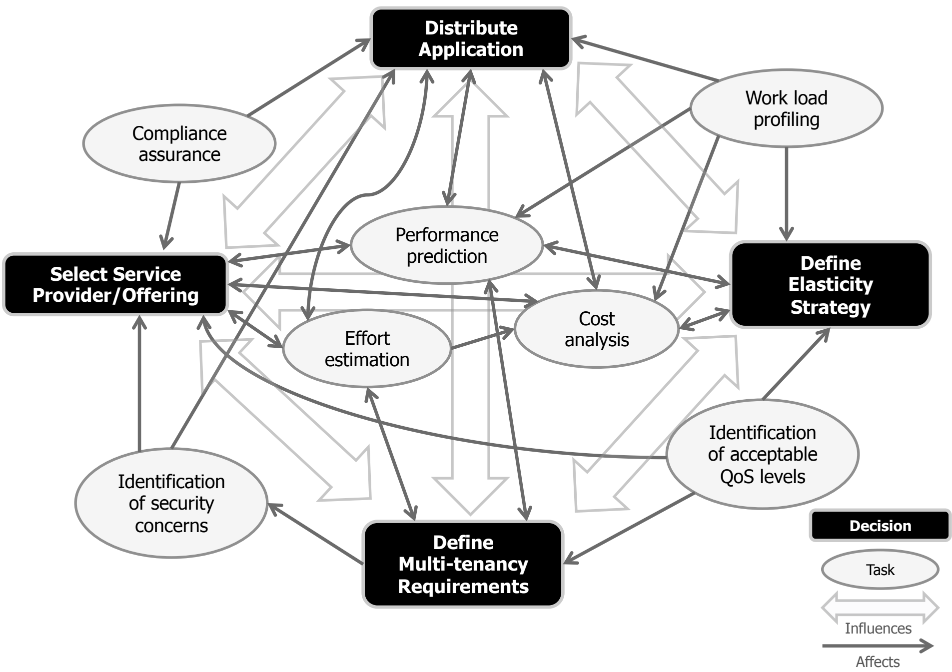


Figure 2.5.: The Conceptual Decision Support Framework for Application Migration to the Cloud [16]

8 Hackystat: https://code.google.com/p/hackystat/

Unlike other State of the Art approaches introduced above, which for instance support decision markers with process suggestions to perform partial application migration or consider only selective kinds of applications and/or Cloud Computing models and service providers, the cDSF focuses on a more general approach to address the intention of migrating an application to the Cloud.

The Cloud migration approach mainly distinguishes *decision points* and *tasks*: four major decision points have to be considered when intending an application migration to the Cloud. Additionally, seven tasks are defined to support proper decision-making in each decision point. All decision points are influenced among each other and task results affect one or more decision point at a time. Figure 2.5 illustrated the overall network structure created by all cDSF elements and their influencing or affecting relationships.

2.4.1. Decision Points

Application Distribution

The decision how an application aims to migrate to the Cloud is distributed in terms of logical and physical placement is one of the central decisions. This refers to how the legacy application can be divided into logical components and/or layers in the first place and which of them could be migrated to a Cloud Computing model beneficially with respect to potential concerns and/or restrictions. Addressing this issue different Cloud migration types [15] can be considered.

The decision, which layer(s) of an application should move to the Cloud, for instance, influences the choice of the Cloud delivery model, but it can also be influenced by for example the preference of a certain Cloud service model. Depending on the distribution choice task like performance prediction, cost analysis and effort estimation are affected. On the other hand expected workloads, compliance restrictions and security concerns affect this decision point.

Define Elasticity Strategy

To cope with existing, possible future, and unpredictable demands in terms of application workload and to reach required Service Level Agreements (SLAs) a decision on how to ensure application elasticity has to be made. At a first glance, providing elasticity is related to service providers and their offerings (Decision Point *Select Service Provider / Offering* in the cDSF), but works like [15] [17] [54] [55] have also exposed application prerequisites and characteristics as influencing factors. Essential questions to be asked affiliated to an elasticity strategy are according to [15] [16] with reference to Suleiman et al. [54]: What (application parts) to scale? How much to scale with respect to limitations (e.g. VM licensing constrains)? How to scale / which type of scaling (vertical and/or horizontal)? When and how fast to scale?

Such questions indicate the relationship of this decision point to service provider characteristics on the one hand but also dependencies to how an application is distributed and which multi-tenancy requirements have to be addressed on the other. For instance, vertical scaling of a single application tier moved to the Cloud may only be achieved if the remaining tiers are able to communicate with multiple instances in the Cloud. Otherwise, appropriate application adjustments have to be taken into account.

Define Multi-tenancy Requirements

As multi-tenancy is one of the essential characteristics of Cloud Computing (see Section 2.1) it is also considerable to define requirements for a existing application moved to the Cloud paradigm which likely is not multi-tenant aware so far. Multi-tenancy awareness of an application comprises two fundamental aspects [15]: communication in terms of ensuring isolated message exchange for each tenant, and administration and management to allow tenant-based configuration and individual management of the application. Possible questions related to this decision point are: What extent of multi-tenancy should be supported? And what multi-tenancy aspects does the application support so far?

Select Service Provider / Offering

How to select a service provider and appropriate offering to fulfill application requirements is, similar to application distribution, another major decision to take in the course of migrating an application to the Cloud. Conditions, like pricing model, delivery or service model, and physical location come along within this decision point. The latter for example has an exceeding influence on other decision points as well as on several tasks like performance prediction, compliance requirements, and security concerns. As one contemporary example related to EU enterprises the physical location of stored data has to be within the EU borders to ensure requirements in terms of legal constrains and data privacy [56].

2.4.2. Tasks

Workload profiling

Based on today's demands of an existing application and its future demand requirements the expected to-be workload profile has to be estimated as an essential input affecting both central cDSF Tasks (i.e. performance prediction and cost analysis). Also workload profiling delivers information support for challenges on how to distribute the application, e.g. deciding which layer(s) or component(s) to move to the Cloud as well as for defining the elasticity strategy most suitable.

Performance prediction

Performance prediction is one of the central tasks in the DSF caused by its responsibility in terms of application behavior after the Cloud migration. Hence, all decision points are affected by this task. Besides, workload profiling provides inputs to estimate performance metrics.

Effort estimation

Effort estimation addresses the amount of work associated with possible adaptions to the existing application depending on choices of how it is distributed, which service provider and offering is selected, and if multi-tenancy requirements are exposed. Results of this task affect cost analysis in the first place but can also lead to reconsideration of previous made decisions, e.g. if adaptation efforts exceed overall benefits of the migrations proposal.

Cost analysis

Similar to the aforementioned performance prediction, cost analysis is also one of the central tasks within the cDSF. On the one hand costs depend on decisions made in the areas of application distribution, service provider and offering selection, and elasticity strategy definition. But on the other hand estimated and analyzed costs may cause reconsideration of previously made decisions and lead to adjustments of some kind. In addition cost analysis is affected by the Tasks workload profiling and effort estimation. The latter one also connects the Decision Point Define Multi-Tenancy Requirements indirectly to the cost analysis Task.

Identification of acceptable QoS levels

This task takes into account existing and also planned Service Level Agreements to identify the needed level of quality of service (QoS) characteristics. A required QoS metric like service availability affects service provider and offering selection based on what availability ratio can be ensured. Furthermore, the task affects both decision points elasticity strategy definition in terms of determine appropriate QoS assurance as well as application requirements in terms of multi-tenancy features.

Compliance assurance

Especially, in case of Cloud Computing when data possibly is transferred beyond enterprise boundaries it is important to ensure compliance to legal and internal regulations, e.g. date privacy regulations for enterprises in the European Union. Results of this task affect the to-be state of the application in terms of how it will and can be distributed. Hence, also the selection of the service provider and offering is directly affected.

Identification of security concerns

As security concerns are typical obstacles related to Cloud Computing they have to be identified with regards to the application moved to the Cloud. Considering critical data and communications, which need protection, exposing such security concerns as well as multi-tenancy characteristics, e.g. data isolation, may affect the task results in an additional way. Then, all these concerns obviously affect both the selection of a service provider and its offering as well as how the application can possibly be distributed.

3. Elaboration of the Decision Support Framework

This chapter now focuses on the elaboration and refinement of the conceptual Decision Support Framework (cDSF) presented in the previous chapter. The process, executed steps, and results of this elaboration is discussed in the following.

3.1. Process of Elaborating the Decision Support Framework

The overall elaboration of the cDSF presented in this chapter follows the process depicted in Figure 3.1.

The initial input for the elaboration process is the conceptual Decision Support Framework [16] on which this Master's thesis is based on. With reference to the cDSF and stated research objectives of this Master's thesis (see Section 1.1) the research scope *Elaboration and refinement of the conceptual Decision Support Framework* has been defined.

- *Decision Support for Application Migration to the Cloud*: This domain specifically focused on literature in the field of supporting decision-making when migrating an existing application to the Cloud Computing environment. This is the most promising research domain as this Master's thesis and the basis publication of Andrikopoulos et al. is assigned to this field of research. For example, all literature introduced in Section 2.3.2 is associated with this domain.

- *Cloud Computing*: Cloud Computing is the second literature research domain since idiosyncrasies specifically related to the paradigm of Cloud Computing have to be considered in case of an application migration to such an environment (e.g. multi-tenancy and security and/or compliance constraints). Furthermore a Cloud service

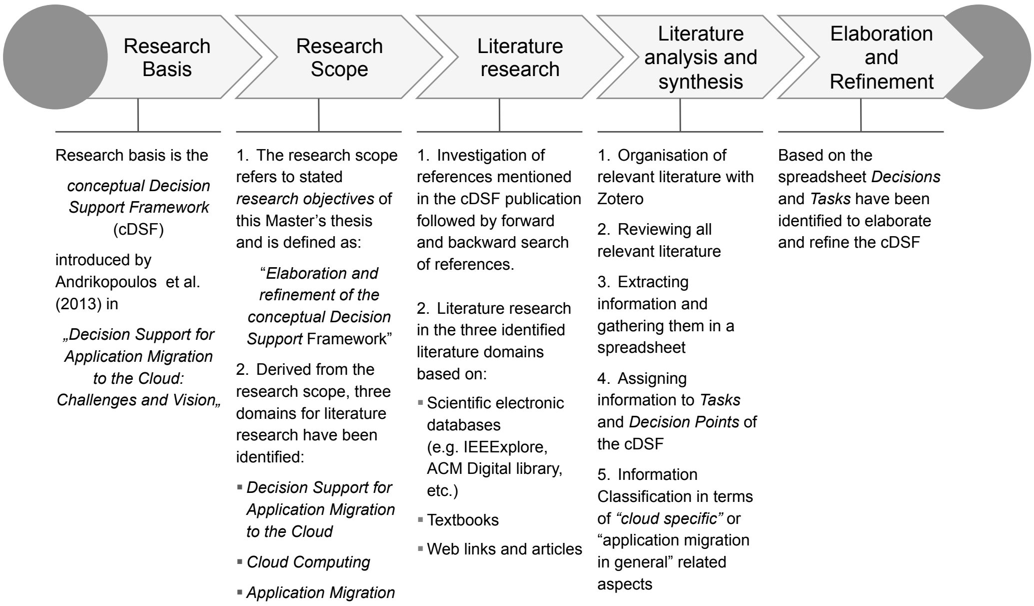


Figure 3.1.: Elaboration Process of the Conceptual Decision Support Framework

provider represents an additional party compared to general application migrations which has to be considered since various offering, SLAs, pricing models, etc. may influence decisions made while application migration considerations.

- *Application Migration*: The third research domain identified addresses the challenge of migrating an application to a new environment. In general, this is not a new task in the IT field and thus decisions and task associated with application migration in general are relevant or can be transferred to the special case of migration to the Cloud.

The next process phase of literature research starts with reviewing and investigating references and literature mentioned in in the cDSF publication [16] with a follow-up forward and backward search of these references. Therefore, a first list of relevant literature for the elaboration and refinement has been created. Further literature search has been done with regards to the three identified literature research domains based on using electronic scientific databases, traditional textbooks as well as Internet web links and article references. The electronic scientific databases used for searching electronic paper and articles are IEEE Xplore[1], ACM Digital library[2], and ScienceDirect[3]. Also Google

[1] IEEE Xplore: http://ieeexplore.ieee.org
[2] ACM Digital Library: http://dl.acm.org
[3] ScienceDirect: http://www.sciencedirect.com

scholar[4] has been used because it offers a centralized and easy overview of scientific contents with additional information (e.g. number of citations) for contents provided by the previous mentioned databases.

Then, in the next process step, relevant contents of the literature research have been organized, analyzed and information considerable for refinement of the cDSF have been extracted. For organization of all different kinds of research sources and contents (e.g. paper and textbook references, web links, etc.) zotero[5] has been used. After all sources and contents have been reviewed relevant information in terms of refining the cDSF have been extracted and gathered in a spreadsheet. Subsequently they have been assigned to the existing decision points and tasks of the cDSF (see Section 2.4) if possible. Otherwise a first draft for a new task or decision has been formalized for later elaboration. In addition, all relevant information in the spreadsheet have been tried to be classified as related to either *application migration in general* or *cloud specific* migration aspects.

Finally, this created spreadsheet provides the basis for the elaboration and refinement of the conceptual Decision Support Framework based on the various information gathered and extracted out of relevant literature and then organized and assigned with regards to existing decision points and tasks of the cDSF.

3.2. Elaboration of Decisions

Based on the prerequisite steps of the elaboration process this section illustrates the elaboration of the conceptual Decision Support Framework. At first, elaborated decisions and their possible outcomes and associated tasks are introduced grouped by their superior decision point (e.g. application distribution, define elasticity strategy, etc.). Consequently, an overview of how decisions are related to each other within the overall framework is given, followed by an introduction of additional tasks elaborated in case of missing input or support of a specific decision.

The elaboration of decisions is limited to their effect on specific outcomes and does not consider how a selection in one decision effects other decisions in terms of limited selections. However, the question if decisions are generally related to others in the framework is addressed after each decision point has been elaborated.

[4] Google scholar: http://scholar.google.com
[5] zotero: https://www.zotero.org

3.2.1. Decision Point 1 - Distribution Application

The following section describes the elaborated decisions in *Decision Point 1 - Application Distribution* as they are shown in Figure 3.2.

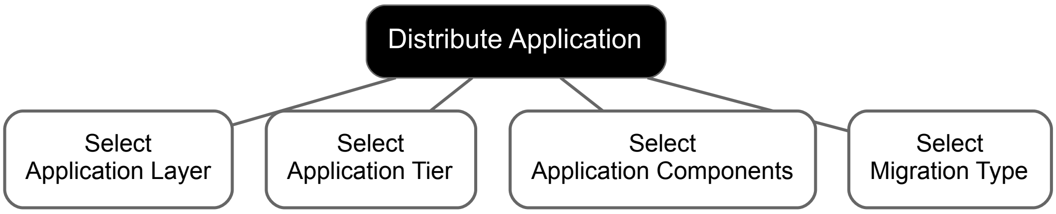


Figure 3.2.: Elaboration of Decision Point 1 - Distribute Application

Decision 1.1 - Select Application Layer(s)

Description

Today's application development typically follows a layer architecture, i.e. the three layers pattern [50] in building software applications. The three layered application architecture encapsulates application functionality to obtain logical separation, which is especially beneficial in case of complex enterprise applications. The pattern comprises a *presentation layer* where functionality is located handling interactions and requests between user and application, a *business layer*, which contains the actual application logic, e.g. calculations or sales analysis, and a *data layer* where databases and functionality related to database communication and interaction are arranged [12] [15] [50].

This encapsulation in a logical manner leads to the decision on how to subdivide existing applications and how they are possibly distributed in terms of Cloud migration. However, a simple replacement of application layers with a Cloud Computing service is not applicable in most cases without at least entailing application adjustments [57]. When considering databases (logically located at the data layer, or more precisely at the database layer) even if Cloud service provider offer databases commonly used on premise, like Oracle Database[6], SQL Server[7], and MySQL[8] [58], there may occur incompatibilities based on different database versions or characteristics and functionalities, which are not implemented by the service provider [15]. Furthermore, the level of data access is important to be addressed appropriately due to its responsibility of ensuring appropriate data access functionality. To address this importance, Strauch et al. [59]

[6] Oracle Database: http://www.oracle.com/us/products/index.html#cn02-Database
[7] Microsoft SQL Server: https://www.microsoft.com/en-us/sqlserver/default.aspx
[8] MySQL: http://www.mysql.com

[60] developed several *Cloud Data Patterns* concerning functional, non-functional, and confidentiality challenges and Hajjat et al. developed an algorithm in [12] to ensure data access after migration to the cloud.

Due to the extend of how single layers may impact the application migration as well as the layered approach for software application architecture is usually used for building today's applications, the decision *Select Application Layer(s)* is identified as a decision in Decision Point 1 - Application Distribution.

Possible outcomes According to the three layers pattern by Fowler [50] described above, the possible outcomes in this decision is either one of these three layers or a combination of multiple layers.

- *Presentation Layer*
- *Business Layer*
- *Data Layer*
- *Multiple Layers*

Relations with Tasks

The outcome selection in this task is affected of several task of the cDSF but in turn also affects certain tasks. How this decision is related to tasks is depicted in Figure 3.2.

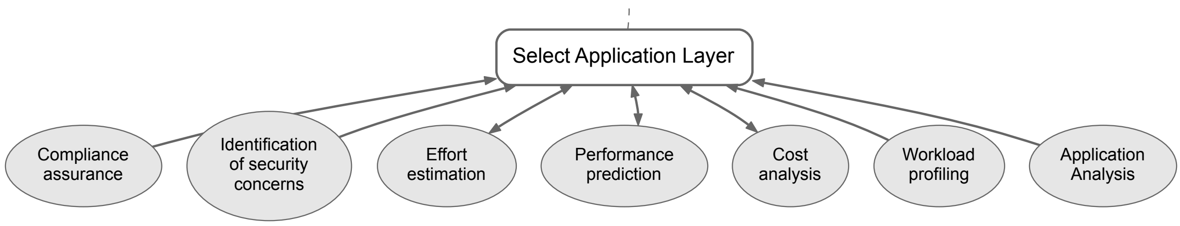


Figure 3.3.: Tasks Related to Decision 1.1 - Select Application Layer

To decide which application layer(s) are possible to consider for the Cloud migration implicates assurance of enterprise compliance in terms of internal and legal regulations as stated in Section 2.4.2. Besides, classified data on the application data layer that might hinder migration approaches also application logic on the business layer can possibly expose problems in case it contains crucial enterprise processes, which constitute competitor advantages. In this sense, security concerns can be raised regarding critical data and communications. For example, such concerns are especially related to data communication occurring between application layers in case of a hybrid cloud migration approach. Data security concerns in term of data privacy regarding the data layer have also to be considered. Approaches like confidentiality Cloud Data Patterns [59] [60] aid to overcome concerns like these.

The Task Effort estimation and this decision are mutually affected. The former is more obvious because moving not all layers requires at least rewiring the connections between layers and in some cases (e.g. moving the business layer with additional multi-tenancy requirements) a higher amount of adjustment effort will occur. On the other hand capacity limitations may affect the decision and impose changes. Performance prediction also affects in both ways as if for instance single or multiple layers are migrated to achieve a hybrid solution this may entail higher response times or delays in data connections. In contrast to that, defined performance values can lead to over-thinking already made decisions. As for example the hybrid cloud approach is stated as probably the most cost-effective one for an organization [61] this decision provides input for cost analysis. In turn, based on a given cost budget a total migration of all layers might not be possible. The identification of which application layers are probably the most beneficial ones to be migrated to the cloud can be done based on workload profiling regarding each specific layer.

In addition to those tasks stated in the cDSF the new Task *Application analysis* has been identified regarding this decision. This fact is also captured by the approach of Cloudstep (see Section 2.3.3) where an application profile is created to identify aspects like for example architecture, technologies, programming language, etc. of the existing application. Regarding this decision, the identification of the application architecture is a necessary prerequisite before a selection of application layers can even be done.

Decision 1.2 - Select Application Tier

Description

While the layers pattern in Decision 1.1 comprises the logical architecture of application functionality also the physical infrastructure where those are actually running (e.g. application and database servers) have to be considered. For this purpose the term *tier* is used instead of layer despite the fact that they are often used in an interchangeable manner [62] [63]. The following tiers can be distinguished [63]: the *data tier* hosts components corresponding to the data layer (e.g. database servers), the *application tier* comprising business logic components deployed on application server(s) and in case of talking about web applications also web server(s) are arranged on this tier, and finally the *client tier* hosts components according to the presentation layer, i.e. in case of web applications this is the users devices running a web browser. This classification is approximately straightforward except of the location of web servers as their functionality is logically arranged on the presentation layer but physically they are assigned to the application tier [12] [63].

Based on this classification it is possible to neglect the client tier in terms of a web application migration to the Cloud as it comprises the users device and web browser to access the actual application and this part cannot be moved to the Cloud. Possibly this consideration is also the reason why the Cloud Migration Types (see Section 2.2) especially focuses on migration of the business a data layer.

Such a distinction of application tiers considering physical resources (i.e. application and/or database servers) encapsulating application parts and implemented functionality is relevant in terms of hybrid application migrations to the Cloud as workloads or performance metrics can easy be estimated on the level of existing servers and whole overloaded servers can be moved to the Cloud more easily. Considering this importance of the application tier perspective the decision *Select Application Tier* has been identified as in Decision Point 1 - Application Distribution.

Possible outcomes

According to the different application tiers introduced previously the outcome of this decision is one of the following:

- *Client Tier*
- *Application Tier*
- *Data Tier*
- *Multiple Tiers*

Relations with Tasks

The tasks related to this decision are shown in Figure 3.4. The view of application tiers is closely connected with the one of application layers hence functionality of a logical layer (e.g. business layer) is commonly located on the belonging physical tier (i.e. application tier). Consequently the same tasks are related to this decision as they are described for Decision 1.1.

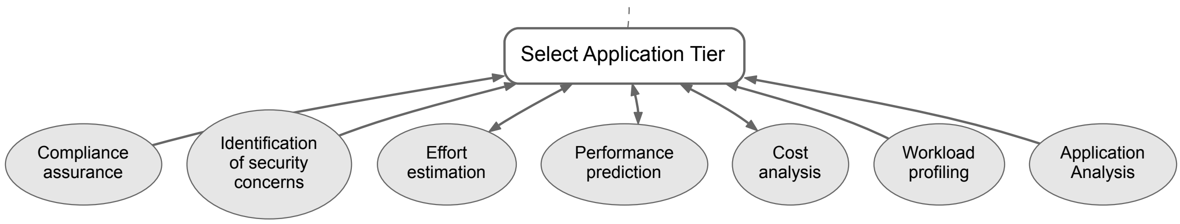


Figure 3.4.: Tasks Related to Decision 1.2 - Select Application Tier

Decision 1.3 - Select Application Components

Description

In some circumstances the granularity of considering applications in terms of application layers or tiers may appear as not appropriate enough since a more detailed separation is required. The layers pattern [50] provides the assumption of multiple subjects, i.e. components, on single layers. For example a component could represent a database server or an instance located on the data tier or a small application functionality like payroll calculation. Such a fine separation into components is beneficial especially in terms of large and complex application scenarios where decisions might have to be made component-wise [11]. So, for instance, a productive database with high workloads is more beneficial and important to be migrated to an elastic and high-available environment than another database dedicated solely to analyzing and/or reporting tasks.

Which components to migrate or to keep locally is a non-trivial decision [12] due to the fact that selected components may influence scalability characteristics in order to support scaling on different application levels to achieve performance goals [34]. But depending on the actual application architecture this decision may experience more or less attention in this decision point. Nevertheless, a selection of single application components might be necessary in terms of application distribution and thus, this decision *Select Application Component(s)* has been identified as a part of Decision Point 1 - Application Distribution.

Possible outcomes

The outcome of this decision depends on the individual architecture of the application considered for migration to the Cloud and which components can be determined in each case. Then the decision outcome can be either a single component or multiple components.

- *Single component*
- *Multiple components*

Relations with Tasks
Tasks affecting this decision are depicted in Figure 3.5. As this decision can be considered as a successor of Decision 1.1 or Decision 1.2 the already mentioned and described tasks of Decision 1.1 are related to this decision and provide input for decision-making.

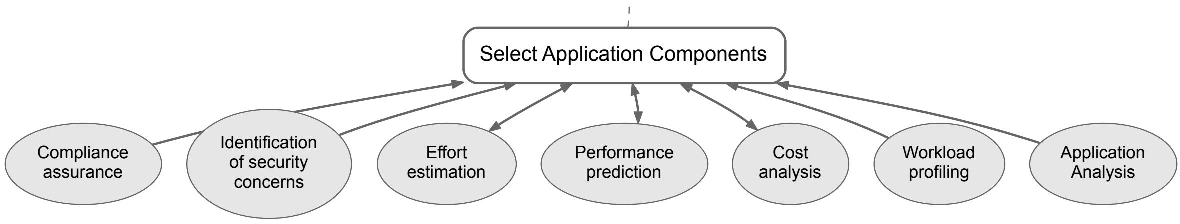


Figure 3.5.: Tasks Related to Decision 1.3 - Select Application Component(s)

Decision 1.4 - Select Migration Type

Description
Considering the migration of an existing application to the Cloud there are different scenarios regarding its architecture and nature on how this can be achieved. The previously introduced decisions in Decision Point 1 already concerned with the application's architecture in terms of components, logical layers, and physical tiers which in turn can either be seen as predecessors or also as successors of this decision about migration types.

In [15] Andrikopoulos et al. examined different possibilities on how existing applications can be enabled for the adoption of Cloud Computing and they identified four *migration types* representing how such an adoption is possible to be performed. In the beginning, each migration type assumes an existing application hosted on-premise in a non-Cloud environment. Based on this prerequisite a migration type is applied, thus a migration of existing Cloud applications between different Cloud provider and offerings is beyond the categorization scope. The four migration types are illustrated in Figure 2.1 and have been described in more detail earlier in Section 2.2.

In the cDSF vision [16] those migration types have already been highlighted as a prospect consideration in terms of the Decision Point 1, because different application topologies are provided how an application might be distributed among the Cloud and a local data center. Such aspects majorly impact other decisions for instance decisions in the context of Decision Point 4 - Select Service Provider / Offering, e.g. the Cloud delivery model selection and can obviously be vice versa as well.

The suggestion of different migration types holds a relevant classification in terms of how an application can possibly be distributed in case of Cloud migration. Hence, it has been identified as a decision in Decision Point 1 - Application Distribution.

Possible outcomes

The approach delivers four different Cloud migration types introduced in Section 3.6. One of these migration types can be chosen as outcome of this decision.

- Type I - Replacement of component(s) with Cloud offerings
- Type II - Partially migration of application functionality
- Type III - Whole software stack migration
- Type IV - Application cloudification

Relations with Tasks

This decision is affected by results of several tasks as depicted in Figure 3.2. Linked to the aforementioned this decision can be either seen as a predecessor or a successor of others decision in this decision point. Accordingly the same tasks as already mentioned in the previous decisions and described in Decision 1.1 are relevant in terms of selecting a migration type.

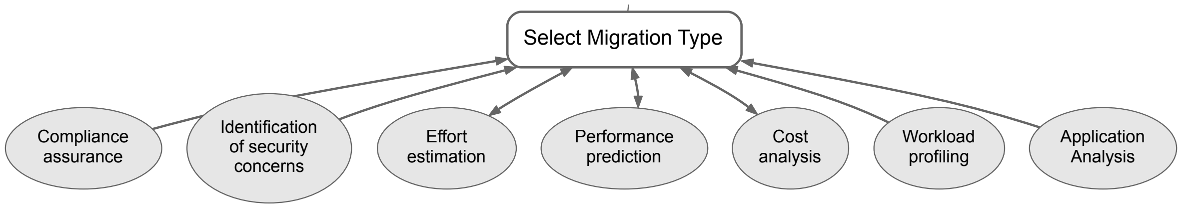


Figure 3.6.: Tasks Related to Decision 1.4 - Select Migration Type

Decision Classification

Decisions identified in Decision Point 1 - Distribute Application can be classified as mainly related to *Application Migration in General* as Table 3.1 points out. In case of the elaborated Decision 1.1 - Decision 1.3 the description above has shown those decisions are concerned with architectural distinctions of an application in terms of either logical or physical distribution. Based on input of tasks like workload profiling, performance prediction, and compliance assurance parts of the application or even the whole application is desired to be migrated to either (re-)continue compliance assurance or reaching performance goals.

According to a general consideration of software migration stated by the ISO/IED [29] these three decisions can be assigned to the migration activity named “Requirements analysis and definition of migration”. In this sense, these decisions are assumed independent of the target environment (i.e. in this case a Cloud Computing environment) and instead be relevant in general application migration cases.

In turn, Decision 1.4 - Select Migration Type focuses specifically on how an application can be distributed in terms of Cloud Computing and thus it is assumed to be *Cloud Migration Specific*. In conclusion, the overall Decision Point 1 cannot be classified in only one distinction precisely and hence it is stated as vague.

Decision	***Classification***
Decision 1	☁ ▷
Decision 1.1	▷
Decision 1.2	▷
Decision 1.3	▷
Decision 1.4	☁

☁ : Cloud Migration specific
▷ : Application Migration in General

Table 3.1.: Decision Classifications in Decision Point 1

3.2.2. Decision Point 2 - Define Elasticity Strategy

In the following section the elaborated decisions in *Decision Point 2 - Define Elasticity Strategy* are described. An overview is illustrated in Figure 3.7.

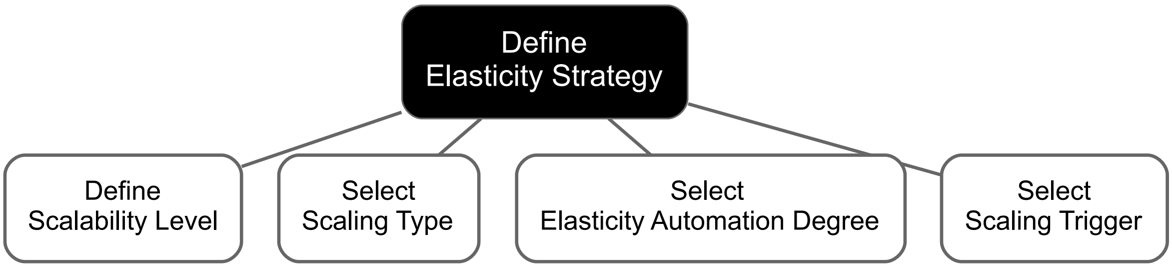


Figure 3.7.: Elaboration of Decision Point 2 - Define Elasticity Strategy

Decision 2.1 - Define Scalability Level

Description

Scalability is a prerequisite in order to speak about elasticity of an application comprising all hardware, virtualization, and software layers associated with the application [64]. Based on this assumption scalability can be performed on different levels to address identified bottlenecks in performance of cloud-based applications. However, an appropriate and comprehensive general classification of scalability levels for Cloud Computing is currently not provided by the literature.

Nevertheless, Vaquero et al. [65] describe levels of application scalability in the Cloud, which recognizes server scalability regarding VMs, virtual network scaling, and platform scaling considering container and explicitly database scalability. In addition, [66] specifies scalability patterns especially in terms of PaaS, which are strongly related to multi-tenancy scenarios.

Based on the above, Figure 3.8 shows an illustration of scalability levels, which are derived based on the previous mentioned literature and inspired by the computing stack used for the multi-tenancy layer classification of Pors [67] (see also Decision 3.2).

The depicted approach in Figure 3.8 is divided in three main areas namely *application level*, *virtualization level*, and *hardware level*. Generally, on each scalability level one of the scaling type introduced in the following Decision 2.2 (i.e. vertical, horizontal, or hybrid scaling) can be applied. For example, vertical scaling implies assigning additional resources and horizontal scaling means replicating and adding instances [54] [66]. As the application level does not provide resources in terms of computing resources then vertical scaling can only be performed on levels where actual physical or virtual computing resources are underlying. For example vertical scaling on application instance level could not be performed since more than one application servers can not host one single instance and furthermore the underlying level (i.e. the virtual machine) would not necessarily

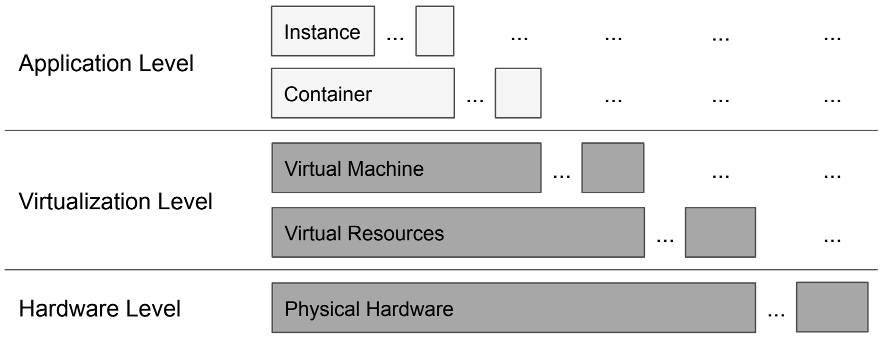


Figure 3.8.: Cloud Computing Scalability Levels

change in its amount of computing resources.

The highest level where scalability of a Cloud application can occur is the *instance level*. On this level workload bottlenecks are addressed by starting and adding new application instances to the existing application server (horizontal scaling), e.g. per application instance only 50 concurrent requests can be handled and with every new instance the total number of possible requests handled increases by 50. As previously mentioned, vertical scaling cannot be performed at application level.

The same holds true for the *container level* in terms of vertical scaling and in case of horizontal scaling; for example new application servers are added which host at least one application instance to address increased workloads. Obviously on the upper level of the scaled out application server application instances can again scale horizontally whereas this will not be sufficient anymore at some point due to resource shortage of the VM.

On the upper first virtualization level the *virtual machine level* is positioned, which instantiates new virtual machines commonly using predefined VM images [11] on horizontal scaling and obtain additional virtual resources in case of vertical scaling. The same holds for the bottom virtualization level, namely *virtual resources level*, which can also perform both vertical and horizontal scaling.

The *physical hardware level* is placed as the grounding level which scales horizontally through appending additional physical hardware (e.g. a server unit) and vertically by replacing a physical server with a more powerful one. As a matter of fact this level has to be further subdivided into a kind of “server level” with underlying “hardware component level” to be fully accurate. However, this is not further considered due to the assumption that datacenters in principle do not scale on the level of single hardware components, neither horizontally (i.e. appending single RAM or disk storage units to a server) nor vertically by e.g. replacing certain CPUs, but rather would simply add a whole new server accomplishing demanded resources instead.

Scalability can be applied to multiple levels of an application and each level may utilize a different scaling type. For example if the VM level scales horizontally, on each VM the container level (e.g. application server) can additionally scale horizontally.

Due to the level where application scalability is applied can influence cost analysis and performance prediction and are assumed relevant in terms of an elasticity strategy, this classification has been identified as a decision in Decision Point 2 - Define Elasticity Strategy.

Possible outcomes
As an outcome in this decision either one or more levels of application scalability can be selected.

- *Instance Level*
- *Container Level*
- *Virtual Machine Level*
- *Virtual Resource Level*
- *Physical Hardware Level*
- *Multiple Levels*

Relations with Tasks
Tasks of the cDSF related to this decision are depicted in Figure 3.9.

In terms of considering the scalability strategy for an application in the Cloud expected workloads are important to be known upfront. Thereby it is possible to determine workloads for different areas of the application regarding the application architecture (e.g. requests at the applications front end may differ from requests at a certain database of the application). Based on workload characteristics the appropriate level to apply scalability can be selected to satisfy defined QoS levels as well as expected performance measures [55]. Consequently performance prediction also affects the selection of scalability levels and vice versa a selected level has to be taken into account by that task.

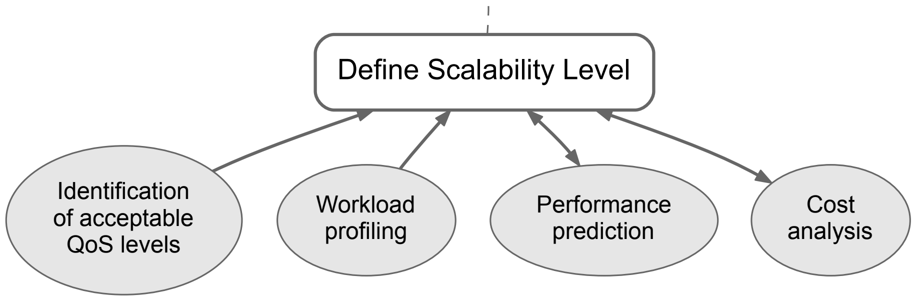


Figure 3.9.: Tasks Related to Decision 2.1 - Define Scalability Level

In the end, a made selection will affect cost analysis hence additional or more powerful resources applied to the application will either be charged by a service provider or expenses on new hardware have to be considered in case of an on-premise Cloud.

Decision 2.2 - Select Scaling Type

Description
Scaling can be performed on different levels according to Decision 2.1. In addition, scaling can be subdivided into mainly two types, which describe how resource under- or over-provisioning is actually addressed. Those two are namely *vertical scaling* (scaling up-down) and *horizontal scaling* (scaling out-in) and a third one *hybrid scaling* represents a combination of the former ones [54] [65]. Vertical scaling is performed by assigning (or removing) additional resources to a single machine [16] [54] [66], e.g. adding more storage resources to a VM. In turn horizontal scaling is utilized by replicating (or removing) instances or new machines on the same level [16] [54], for instance replicating application instance on a virtual machine. The third case comprises scaling in both directions vertical as well as horizontal, i.e. application instances are replicated on a VM while also additional resources are attached to it.

However, which type of scaling can be applied to an application depends on various characteristics e.g. of the application itself, the Cloud Computing service model, or the specific offering of a service provider [16]. Whereas vertical scaling is mostly unrelated to the application and depends more on service provider offerings in turn horizontal scaling largely involves applications characteristics, e.g. stateless components featuring REST are easier to be replicated [68].

This detailed view on how scaling of an application is actually performed as well as other decisions are influenced by the selected type of scaling, this decision has been identified as relevant in Decision Point 2 - Define Elasticity.

Possible outcomes

One of the scaling types described above can be selected as an outcome for this decision.

- *vertical scaling* (scaling up / down)
- *horizontal scaling* (scaling in / out)
- *hybrid scaling* (a combination of vertical and horizontal scaling)

Relations with Tasks

This decision is related to the same tasks like Decision 2.1 as shown in Figure 3.10. An appropriate scaling type to ensure right and fast resource provisioning to the application on variable workloads is especially crucial in terms of costs because additionally demanded resources that remain idle cause unnecessary costs.

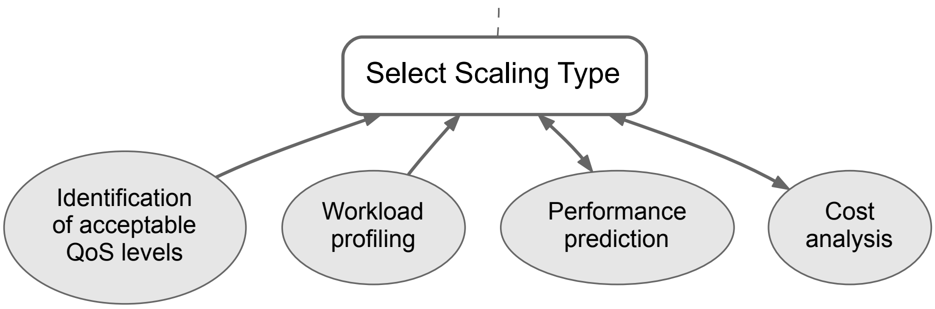


Figure 3.10.: Tasks Related to Decision 2.2 - Select Scaling Type

Decision 2.3 - Select Elasticity Automation Degree

Description

In the context of elasticity the term *automatic scaling* (auto-scaling) is often mentioned as a characteristic of Cloud services, for example in services like Amazon Elastic Compute Cloud (Amazon EC2) [69] or those provided by Rackspace [70]. However, according to

NIST the essential characteristic of Cloud Computing "rapid elasticity", i.e. appropriate provision and release of demanded resources, needs to be performed automatically only *in some cases* [22]. In literature the elastic adaptation process is described as usually automated but manual steps are conceded [64].

Therefore, automation in scaling is typically attained by a set of defined rules, either partially predefined rules of a service provider or user-defined rules, which are periodically checked in terms of certain metrics have exceeded defined thresholds to govern how the service scales up or down to adapt to variation in workloads [55] [65]. In general, rules in this perspective consider infrastructure or platform metrics (e.g. CPU, memory, etc.) whereas in some cases also server-level metrics (e.g. cost to benefit ratio) or even more complex condition settings (e.g. arithmetic and logic combinations of simple rules) are possible to be expressed as rules [65].

Given this, truly auto-scaling with no human intervention necessary [71] is not offered in practice and literature uses the term auto-scaling even though it is more a *semi-automatic scaling* where automation is achieved by periodically checking user-defined rules. A scaling even less automated can be considered in case a Cloud owner adds server instances in an Amazon Cloud manually or a more reasonable case when new physical servers are bought and added to a datacenter. In such a situation, adaptations are mostly done by *manual scaling*. But despite these two possibilities actual *automatic scaling* is mentioned only in literature and the approach of proactive scaling by Sallam and Li [71] introduces a possible attempt.

In order to address differences in the degree of elasticity automation in an application elasticity strategy for its Cloud migration, this decision *Select Elasticity Automation Degree* has been identified in as part of Decision Point 2 - Define Elasticity Strategy.

Possible outcomes

According to the above explanations about different automation degrees in elasticity, one of the following three outcomes can be selected in terms of this decision.

- *Manual scaling*
- *Semi-automatic scaling*
- *Automatic scaling*

Relations with Tasks

In this particular decision the tasks depicted in Figure 3.11 have to be considered. As mentioned by previous decisions in this decision point also this decision is related to workload profiling and performance prediction which entail effects to cost analysis. But particularly in case of manual scaling special capabilities for actually performing scaling activities might be necessary and that is why the new Task Workforce capabilities identification is additionally applied to this decision.

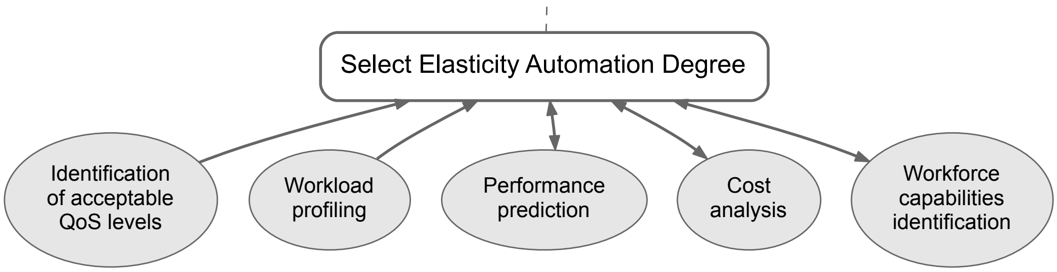


Figure 3.11.: Tasks Related to Decision 2.3 - Select Elasticity Automation Degree

Decision 2.4 - Select Scaling Trigger

Description

In [54] Suleiman et al. highlight the factor of time, which is needed until scaling actions take their full effect, i.e. the delay between initiating and actual availability of a new added VM for example. This time has been defined as *scaling latency* according to [72] or also referred as the instance *spin-up time* [55] affecting performance and cost [54].

The authors of [54] describe two possible approaches: *event-driven* scaling, which means reactive triggering based on monitoring scaling rules and reacting on exceeding thresholds, and in contrast to that a *proactive* scaling approach which considers log files and real-time measures to predict scaling actions right in time. For example, Rackspace provides event-driven scaling by using the Rackspace Cloud Monitoring to trigger certain defined policies based on given thresholds [70].

This shows the importance of considering the scaling latency in order the achieve best elasticity for an application in the Cloud and thus this decision *Select Scaling Trigger* has been identified in Decision Point 2 - Select Elasticity Strategy.

Possible outcomes
The possible outcome in this decision is one of the above described trigger options.

- *Event-driven*
- *Proactive*

Relations with Tasks
This decision considers the scaling latency in order to determine how fast scaled resources are available to use. As mentioned above this mainly affects performance and costs, which is, why only the tasks Performance predication and Cost analysis are related to this decision.

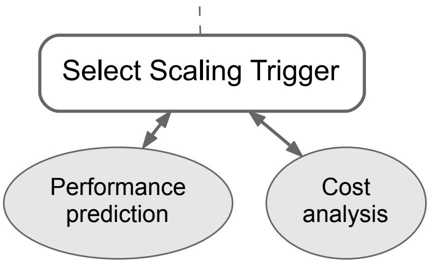


Figure 3.12.: Tasks Related to Decision 2.4 - Select Scaling Trigger

Decision Classification

Decisions in Decision Point 2 - Define Elasticity Strategy are classified as *Cloud Migration Specific* as well as related to *Application Migration in General* like Table 3.2 shows. Due to the fact that scalability by itself is not an exclusive characteristic provided by Cloud services, but rather a fashion which deals with how resources can be provided fast, easy and appropriate in terms of varying demand situations also non-Cloud environments can be characterized as scalable.

Nevertheless, today the term scalability is widely used and related to Cloud Computing terminology in terms of describing its characteristic of rapid elasticity as mentioned in the Cloud Computing definition in Section 2.1.

Decision	***Classification***
Decision 2	☁ ▷
Decision 2.1	☁ ▷
Decision 2.2	☁ ▷
Decision 2.3	☁ ▷
Decision 2.4	☁ ▷

☁ : Cloud Migration Specific
▷ : Application Migration in General

Table 3.2.: Decision Classifications in Decision Point 2

3.2.3. Decision Point 3 - Define Multi-Tenancy Requirements

The following section presents elaborated decisions in *Decision Point 3 - Define Multi-Tenancy Requirements* illustrated in Figure 3.13.

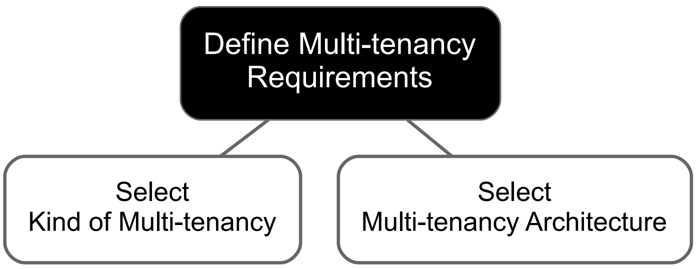


Figure 3.13.: Elaboration of Decision Point 3 - Define Multi-Tenancy Requirements

Decision 3.1 - Select Kind of Multi-tenancy

Description
In general, the characteristic of multi-tenancy can be distinguished into two kinds of patterns regarding Guo et al. [73]. The first *multiple instances multi-tenancy* describes a pattern where each tenant is supported with a dedicated instance (e.g. application or database instance) located in a hosting environment consisting of either a shared operation system (OS) and middleware layer on a shared hardware or in a more isolated manner on a dedicated OS and middleware (i.e. partition based on virtual machines) on a shared hardware. The second pattern shows *native multi-tenancy* in which case tenants share a single application based on shared resources below.

Multiple instances mulit-tenancy usually supports tenants up to dozens whereas native multi-tenancy is purposed to support even hundreds or thousands of tenants [73]. But with an increasing number of tenants sharing the same environment, it is crucial to ensure QoS for each tenant so that for example increased workloads of one tenant do not affect performance and QoS of other tenants in the same environment.

The decision, which kind of multi-tenancy to choose depends on enterprise requirements connected to the application migrated to the Cloud as well as on characteristics of the application itself. For example, small and medium-sized businesses (SMBs) may choose lower costs with native multi-tenancy and accept limitations in QoS whereas large enterprises are likely willing to pay a premium for more isolation (e.g. dedicated resources) in terms of preventing potential risks related to shared resources. Furthermore, the application characteristics in terms of multi-tenant awareness determine the outcomes in this decision, but in many cases where multi-tenancy requirements are treated it is a matter of how much effort an enterprise is willing to spend for application adjustments.

Since this is a general distinction between two different types of multi-tenancy, this decision *Select Kind of Multi-tenancy* has been identified in Decision Point 3 - Define Multi-Tenancy Requirements.

Possible outcomes

The possible outcome in terms of this decision is one of the two multi-tenancy patterns described above.

- *Multiple instances multi-tenancy*
- *Native multi-tenancy*

Relations with Tasks

In Figure 3.14 the tasks are depicted with a relationship to this decision.

With regards to the cDSF, the decision point of multi-tenancy affects the identification of security concerns. Multi-tenancy is about the trade-off between isolated and shared resources which in turn defines the basis for the identification of security concerns. For example, data stored on an dedicated server is generally suspected to be more secure than on a server shared among multiple users. However, choosing one kind of multi-tenancy mentioned above will expose adjustment efforts in case of an non-multi-tenancy

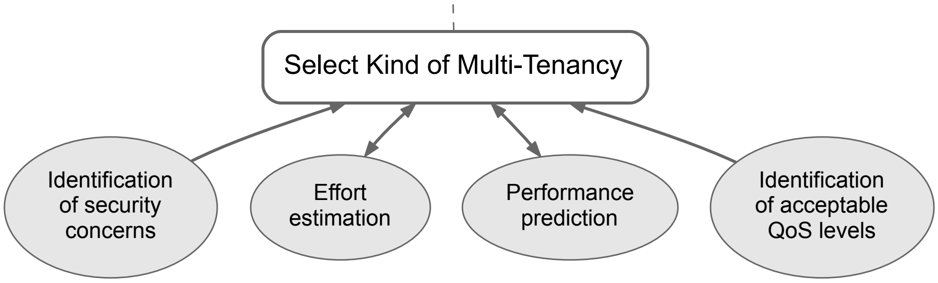


Figure 3.14.: Tasks Related to Decision 3.1 - Select Kind of Multi-Tenancy

aware application, e.g. a totally shared native multi-tenancy approach will entail much higher efforts than a separated instances per tenant. If a given threshold is exceeded the Task Effort estimation may also indicate the need of reconsidering this decision. Same holds for estimated performance goals that can be input for this decision. In turn the number of expected tenants needed to be served concurrently can also affect this task and an appropriated kind of multi-tenancy has to be selected. In this case, also requirements in terms of QoS impact influence the selection of the multi-tenancy kind.

Decision 3.2 - Select Multi-Tenancy Architecture

Description

Multi-tenancy can be applied to different application parts. In [67] a comprehensive analysis of current literature aimed to identify those levels, which are shown in Figure 3.15. Basically, the findings divide multi-tenancy regarding the physical architecture of an enterprise application (introduced in Decision 1.2 - Select Application Tier) into an application and database perspective and extend them by underlying levels where multi-tenancy can be applied to certain resources (e.g. virtual machines, etc.). According to this differentiation multi-tenancy can be applied independently on several levels of the actual application business logic (i.e. on the application tier) and of the persistence components like databases (i.e. on the data tier) of an application. Depending on how the application is desired to be distributed only one or both perspectives have to be considered if multi-tenancy requirements are exposed for the case of application migration at all.

Figure 3.15 presupposes that the multi-tenancy continuum of sharing versus isolation is oriented bottom-up where the lowest level (i.e. hardware) indicates the least of sharing and thus the highest level of isolation. This applies vice versa for the highest layer.

Consequently, when multi-tenancy is applied to a certain level all lower levels are shared among tenants and in turn isolation arise on those levels above. For instance if multi-tenancy is applied on the level of virtual machines the virtual machine itself and resources below (i.e. hardware resource) are shared among tenants, but levels above (e.g. an application or database server) are provided isolated for each tenant.

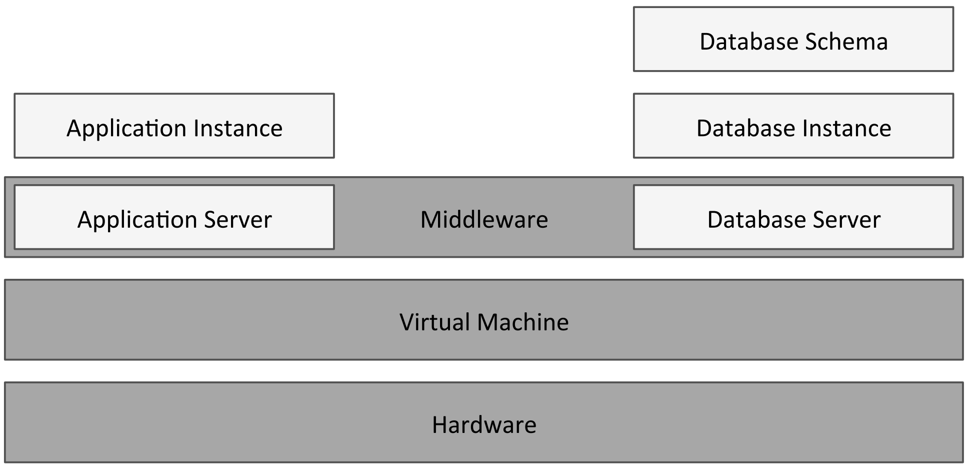


Figure 3.15.: Stack of Multi-Tenancy Levels

Based on the evaluation of different multi-tenancy levels [67] also constructed several so-called *multi-tenancy architectures* (MTAs). Those MTAs are composed according to the multi-tenancy levels shown in Figure 3.15. In terms of the application tier four levels can be counted, regarding the data tier five levels and each perspective is extended by one case where no multi-tenancy is applied. Hence, a total number of 30 (five application levels times six data levels) possible multi-tenancy architectures can be distinguished as illustrated in Figure 3.16. The bottom-left case of no multi-tenancy in both perspectives is excluded, because at least any requirements for multi-tenancy are assumed to be stated if multi-tenancy is considered in the application migration.

Not only does the introduction of multi-tenancy architectures show a comprehensive classification of various multi-tenancy scenarios for an application migration approach to the Cloud, but also it shares the view of application architecture distinction according to Decision 1.2. Furthermore, a multiple instance multi-tenancy selection in the previous Decision 3.1 can be specified in more detail by an MTA. For this reason this decision *Select Multi-Tenancy Architecture* has been identified as a part of Decision Point 3 - Define Multi-Tenancy Requirements.

Possible outcomes

The possible outcome in this decision is one of the 29 distinguished multi-tenancy architectures depicted in 3.16. At this place a further description of each single MTA is omitted, because most of them are described in detail by Pors [67] (because his research focuses specifically on service providers and their development of SaaS applications so only half of the MTAs have been relevant for his consideration). But the purpose and construction of MTAs is very straightforward with the aid of the MTA overview in Figure 3.16.

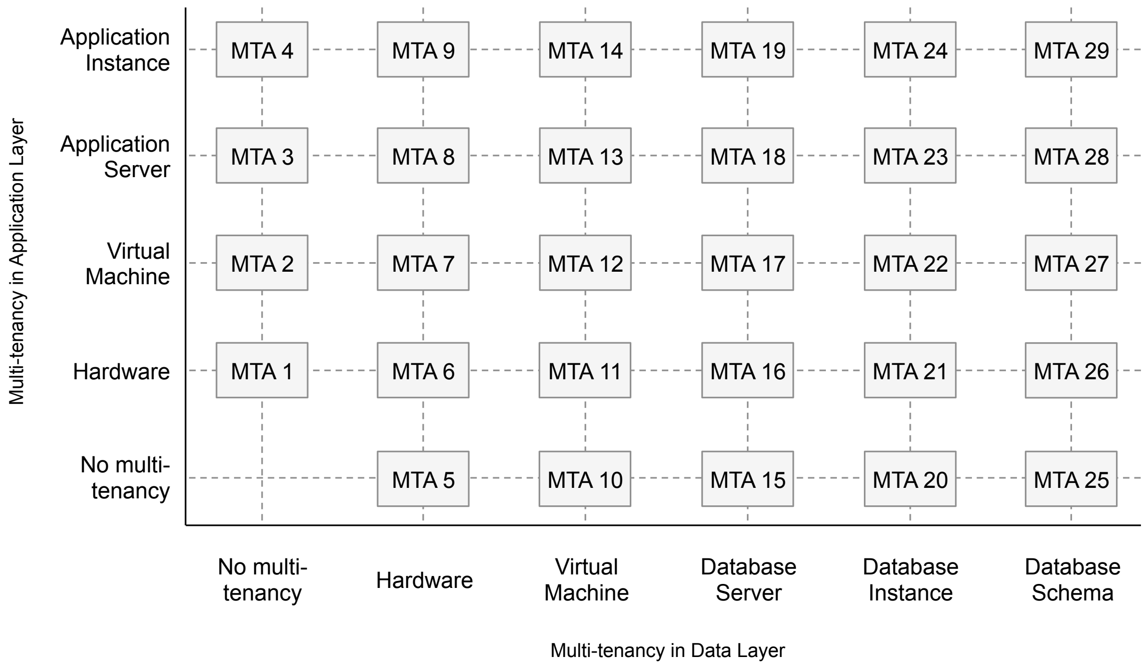


Figure 3.16.: Possible Multi-Tenancy Architectures

Relations with Tasks

This decision is affected by the same tasks as Decision 3.1 as illustrated in Figure 3.17. Assuming the more detailed distinction of MTAs the inputs from tasks like Identification of acceptable QoS levels or Performance prediction also have to be delivered more precisely, i.e. if multi-tenancy should be applied on the level of application and/or data inputs metrics have also to be provided on this granularity level. But in turn also more accurate information in terms of outcomes are delivered by this decision.

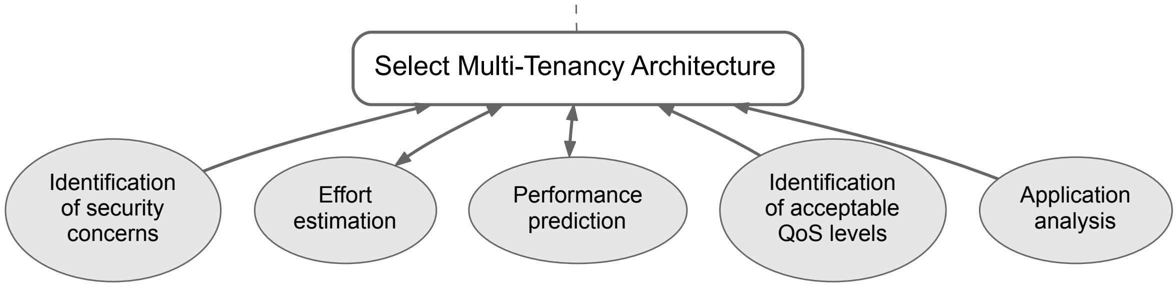


Figure 3.17.: Tasks Related to Decision 3.2 - Select Multi-Tenancy Architecture

Decision	***Classification***
Decision 3	☁
Decision 3.1	☁
Decision 3.2	☁

☁ : Cloud Migration Specific
▷ : Application Migration in General

Table 3.3.: Decision Classifications in Decision Point 3

Decision Classification

If one considers Decision Point 3 concerning multi-tenancy, i.e. provisioning pooled resources to several consumers, two decisions have been identified and their classification is presented in Table 3.3.

With reference to [74], Bezemer and Zaidman pointed out multi-tenancy as an approach for SaaS applications and a relatively unexplored field of research first mentioned in 2005. Furthermore, they delimited multi-tenancy from the concepts of multi-user and multi-instance. In addition, multi-tenancy is stated in the NIST Cloud Computing definition introduced in Section 2.1 as a part of one of its essential characteristics as well as [75] mentioned it as a key attribute of SaaS applications.

Based on this characterization decisions regarding multi-tenancy are assumed to be solely classified as *Cloud Computing Specific*.

3.2.4. Decision Point 4 - Select Service Provider / Offering

The following section considers the elaborated decisions of *Decision Point 4 - Select Service Provider / Offering* that are shown in Figure 3.18.

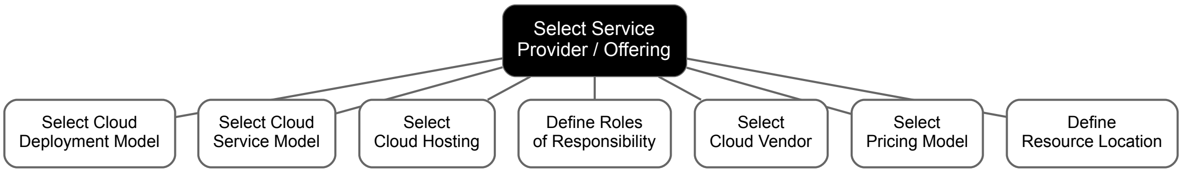


Figure 3.18.: Elaboration of Decision Point 4 - Select Service Provider / Offering

Decision 4.1 - Select Cloud Deployment Model

Description

The view of deployment models is one of the key differentiations in Cloud Computing taking into account with *whom* Cloud resources are shared. Chapter 2.1 introduces those deployment models in more detail, which are distinguished in literature and also used for offering descriptions by some service provider, for example Rackspace [76]. According to the characteristic of resource sharing from minimal to maximal degree of sharing the following deployment models are known: *Private Cloud*,*Community Cloud*, and *Public Cloud*. Additionally, a composition of two or more models is known as *Hybrid Cloud*.

As a consequence of being a key characterization in Cloud Computing deployment models are considered also by other publications regarding decision support for application migration to the cloud (see Chapter 2.3). In context of this decision support approach this decision also determines several other decisions and tasks across the framework. Hence, this decision *Select Cloud Deployment Model* has been identified in Decision Point 4 - Select Service Provider / Offering.

Possible outcomes

The possible outcomes of this decision are determined by the characterization of deployment models stated in the NIST definition of Cloud Computing [22].

- *Private Cloud*
- *Community Cloud*

- *Public Cloud*
- *Hybrid Cloud*

Relations with Tasks

While the decision context is related to almost all existing tasks of the cDSF this specific decision is related to only a subset of tasks depicted in Figure 3.19.

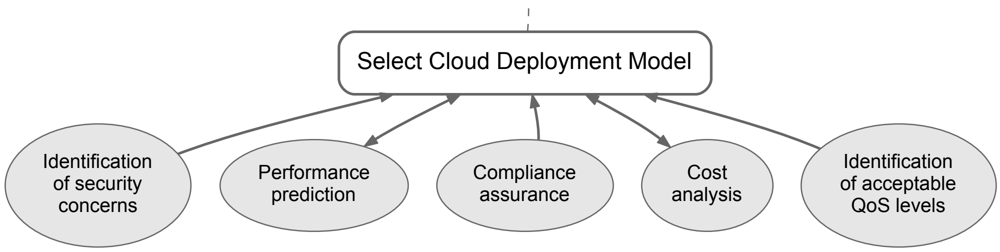


Figure 3.19.: Tasks Related to Decision 4.1 - Select Cloud Deployment Mode

The Cloud deployment models differ in their amount of resource sharing which is concerned by the Task Compliance assurance that provides inputs for this decision. Regulations in terms of legal requirements such as data privacy constraints but also individual enterprise regulations affect the deployment model selection. In addition, security concerns may also affect because for example a private Cloud provides more resource isolation than a public Cloud. This in turn will probably entail higher costs and analyzing them may require reconsideration of selected outcomes. The identification of acceptable QoS levels provides input associated with performance prediction for instance if demanded QoS and performance levels can not be ensured by a public Cloud, because resources are shared and variable workloads of other tenants more likely affect those confirmations. In turn both tasks also affect vice versa, i.e. service provider state specific QoS and performance levels in their offering regarding Cloud deployment models (e.g. Rackspace [76]).

Decision 4.2 - Select Cloud Service Model

Description

Concerning Cloud Computing terminology the term service model is regarded to classify *what* kind of resource is provided as a Service. Since different Cloud Computing definitions exist (see Section 2.1) this classification can be made in diverse dimensions but according to the widespread and commonly used NIST definition [22], the following three service models are known: *Infrastructure as a Service (IaaS)*, *Platform as a Service (PaaS)*, and *Software as a Service (SaaS)*. Service providers do not usually name their offerings according to this terminology as for example Amazon serves its Cloud services under the name Amazon Web Services and IaaS virtual services for instance are called Amazon EC2 [77]. But some providers like the IBM Cloud offerings stick to the terminology [9].

When considering the migration of an application to the Cloud the selection of an appropriate service model is likely the most crucial decision in terms of being consistent with the desired composition of how the application will be distributed in the Cloud (e.g. which migration type is selected in Decision 1.4, etc.). Other decision support approaches for Cloud migration considered in Section 2.3 share this distinction and also mention the consideration of the service model.

The classification of service models is one of the major distinctions in Cloud Computing theoretically, as well as used in terms of service provider offerings in practice. Furthermore, its relevance in decision support for application migration is mentioned by other decision support approaches in this field, which is why this decision *Select Cloud Service Model* has been identified as a part of Decision Point 4 - Select Service Provider / Offering.

Possible outcomes

According to the service model distinction above the possible selected outcome of this decision is one of the following:

- *Software as a Service (SaaS)*
- *Platform as a Service (PaaS)*
- *Infrastructure as a Service (IaaS)*

Relations with Tasks

To which tasks of the cDSF this decision is related to is shown in Figure 3.20.

All tasks mentioned by the previous Decision 4.1 are affecting the selection of a service model as well, whereas this decision highly depends on how the desired application will be distributed between the Cloud and in-house resources, which Decision Point 1 is concerned with. In this regard the task effort estimation is additionally affected because

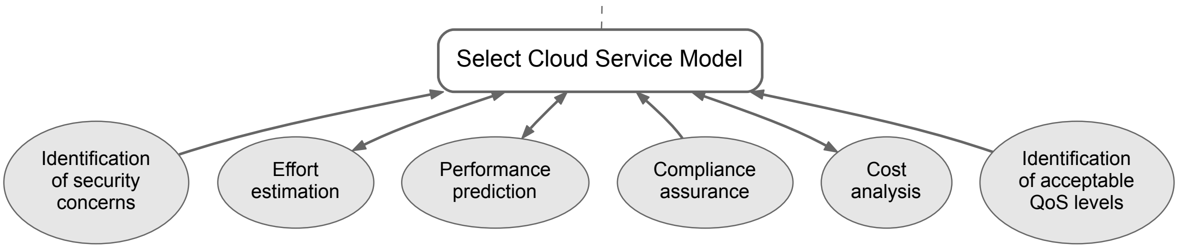


Figure 3.20.: Tasks Related to Decision 4.2 - Select Cloud Service Model

shifting an application encapsulated to an IaaS is assumed to consume less effort in terms of application adjustments than reconstructing a non-Cloud application to be a *cloudified* SaaS.

Decision 4.3 - Define Cloud Hosting

The fact *from whom* the service and resources are provided and hosted is another major consideration in terms of Cloud Computing and therefore relevant for this decision point. From the perspective of the Cloud consumer the Cloud service can either be hosted on-premise (in-house) or off-premise (i.e. external) by a service provider [3] [22]. In this sense an in-house private Cloud IaaS would for example mean an infrastructure service (IaaS) is provided exclusively for the same organization (private Cloud) and the resources themselves are hosted in the organization's datacenter, whereas any public Cloud services are provided off-premise by service providers [22]. Additionally, a hybrid combination of both types would be achieved for instance if the previous scenario of in-house private Cloud IaaS is extended by public infrastructure resources to cover possible unpredictable workloads.

The hybrid approach of combining on-premise and off-premise Clouds is today probably the most cost-efficient and commonly used solution in enterprises because needed resources can easily be extended and it addresses especially security concerns in terms of preventing the risk of losing control of sensitive data when keeping them on-premise [61] [78]. On the other hand further challenges are related to such a solution like increasing latency in data transfer between in-house and external hosted resources and additional costs for those data transfers [79].

This consideration presents the hosting aspect of a Cloud service and its resources as a relevant point of view in application migration to the Cloud, thus this decision *Define*

Cloud Hosting has been identified as a part of Decision Point 4 - Select Service Provider / Offering.

Description
According to the classification described above, the following outcomes of this decision are possible:

Possible outcomes

- *On Premise Cloud Hosting*
- *Off Premise Cloud Hosting*
- *Hybrid Cloud Hosting*

Relations with Tasks
The tasks associated with this decision are depicted in Figure 3.21.

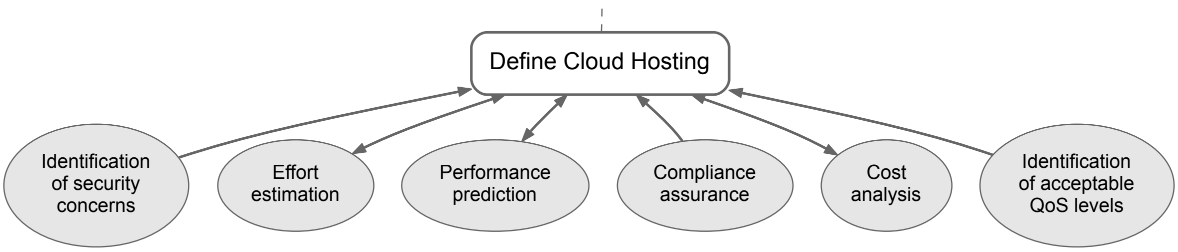


Figure 3.21.: Tasks Related to Decision 4.3 - Define Cloud Hosting

Regarding this decision the same tasks like in Decision 4.2 are related. This decision mainly deals with the situation of data leaving enterprise boundaries or not in which case both tasks compliance assurance and identification of security concerns affect this decision. Furthermore, those tasks concerning performance prediction and QoS affect this decision since external resources are likely associated with increasing response times and latency. In turn external resources are supposed to be less cost intensive due to service providers achieve economies of scale better than on premise resources. In addition, effort estimations can be affected because following a hybrid approach will entail application adjustments.

Decision 4.4 - Define Roles of Responsibility

Description
Associated with Cloud service infrastructures the three kinds of responsibility roles *ownership*, *operation*, and *management* are distinguished according to the NIST definition [22]. Furthermore, those responsibility roles can be held by either the organization (i.e. the enterprise), which actually uses the Cloud service infrastructure, a third party, or some responsibility combination of both. For example an organization using a private Cloud can be the resource owner (e.g. owner of physical servers) but due to IT management and maintenance is outsourced to some third party responsibilities in terms of operation and management are delegated.

Consideration and clarification which of those roles and responsibilities can be fulfilled and held by the organization or have to be delegated to some third party is crucial and should be done before any Cloud adoption [80]. This decision obviously can affect cost analysis in terms of possible costs for additional third party support. Beside costs it may be more important to consider if IT resources (i.e. IT staff) and/or capabilities are sufficient enough to fulfill activities related to those respective responsibilities to ensure service availability. Likely in such a case the more effective and efficient decision is to obtain a service where responsibilities are totally covered by a third party service provider.

The distribution of responsibilities related to a Cloud service infrastructure is crucial in case of considering the migration of an application to pretty much new Cloud Computing environments and hence this decision *Define Roles of Responsibility* has been identified in Decision Point 4 - Select Service Provider / Offering.

Possible outcomes
According to the three kinds of responsibilities mentioned above and how they can be held by either the organization itself or by a third party service provider, Figure 3.22 shows the alternative *Role Set* combinations representing all possible outcomes of this decision.

Relations with Tasks
Considering related tasks of the cDSF this decision can be affected by tasks addressing compliance assurance, security concerns as well as identification of QoS level. However, in this case it is more important to consider capabilities the organization is able to provide regarding activities associated with certain responsibility roles. For example not every enterprise runs its own IT department with existing Cloud Computing knowledge

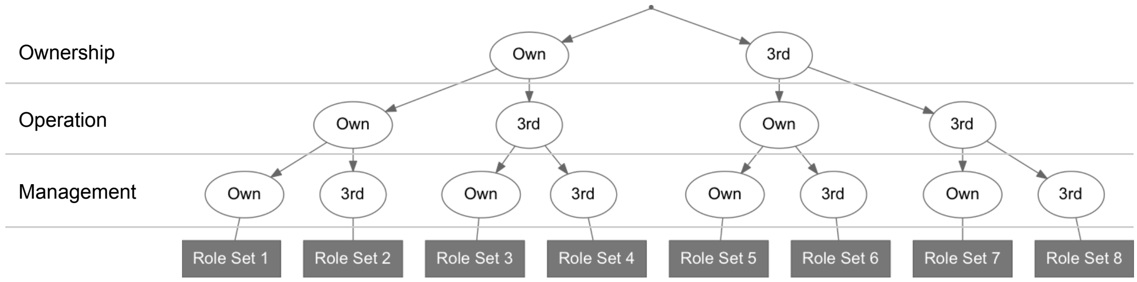


Figure 3.22.: Cloud Service Responsibility Role Sets

to fulfill hardware management and maintenance and to operate or manage a Cloud. Even if an IT department with appropriate resources is present their available capacity has to be checked, because holding more responsibilities is associated with activities that need to be addressed by a corresponding amount of workforce capacity. As a result a new Task Workforce capabilities identification is introduced and affected by this decision. An overview of all tasks related to this decision is given in Figure 3.23.

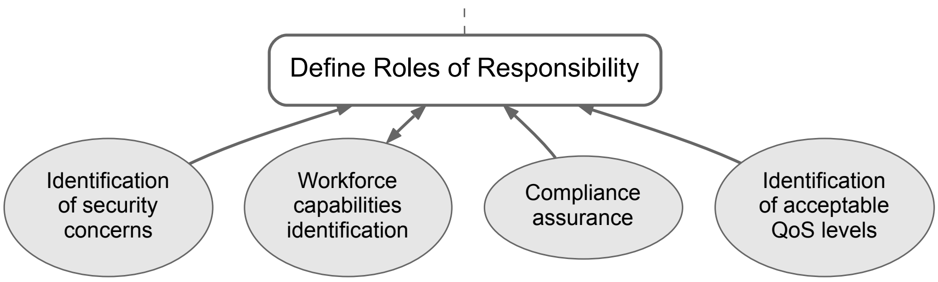


Figure 3.23.: Tasks Related to Decision 4.4 - Define Roles of Responsibility

Decision 4.5 - Select Cloud Vendor

Description

At a glance, aspects like costs and functional suitability are the most important criteria in enterprise selection of a Cloud provider and its offering but an evaluation will likely and should involve the vendors general reputation [81] [82]. To address this issue [81] consider several characteristics or attributes like reference projects, benchmarks, reports, etc. regarding a vendor's reputation and also aspects like resources, knowledge, skills, etc. in terms of the vendor's capabilities. The Info-Tech Research Group [83] created an

Excel-based tool to shortlist Cloud vendors (currently from a list of eight vendors) based on answering seven question regarding expected service characteristics and subsequently ranking the importance of Cloud vendor criteria like for example Affordability, Support, or Vendor Viability.

Since enterprises are commonly interested in long-term relationships with external parties involved in the application migration to the Cloud , the selection of a Cloud vendor should comprise such “soft” facts mentioned above beside cost and functional requirements. To address this issue this decision *Select Cloud Vendor* has been identified as a part of Decision Point 4 - Select Service Provider / Offering.

Possible outcomes
The possible outcome of this decision depends on an evaluation of different Cloud vendors, which are appropriate and suitable for the individual requirements of the desired application migration.

- *Evaluated Cloud Vendor*

Relations with Tasks
The decision to choose an appropriate Cloud vendor is beyond functional and technical aspects that are taken into account by the cDSF so far because it considers “soft facts” like reputation, reference customers, surveys and publications, etc. to evaluate suitable Cloud vendors for the application migration. As a result the new Task *Vendor evaluation* has been elaborated to address this issue and provide inputs for this decision, illustrated in Figure 3.24.

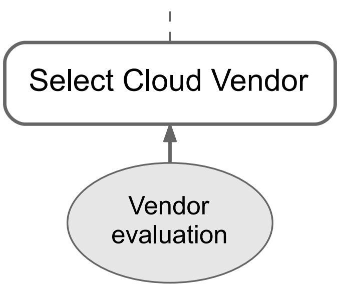


Figure 3.24.: Tasks Related to Decision 4.5 - Select Cloud Vendor

Decision 4.6 - Select Pricing Model

Description

In terms of cost consideration the adoption of Cloud Computing is often related to conversion of capital expenses (CapEx) to operating expenses (OpEx) because expenses on physical IT resources are now replaced by obtaining those resources on-demand and charged by using a "pay as you go" pricing approach [84]. This generally constrains the consideration of pricing models to the assumption that Cloud resources are owned and charged by a service provider and not owned by the Cloud user itself.

Pricing the consumption of Cloud resources uses various terminologies and can be done by different measures, which might not always be in line with the model of "pay as you go" [22] [54] [81]. In general two pricing models *Pay-Per-Use* and *Charge-Per-Use* (i.e. subscription) can be distinguished. The former comprises Cloud resources like CPU, bandwidth, etc. that are billed based on their time of usage, e.g. using an Amazon EC2 virtual server instance costs $0.113 per hour [85]. Depending on the kind of Cloud resources a pricing can also be made *Pay-Per-Unit* where the amount of actual used resources is billed for a certain period of time, e.g. a dedicated Rackspace server for $499 a month [86]. Those two pricing models constitute the characteristic of "pay as you go" as they basically do not require upfront payments, long-term commitments, and can be requested easily by the Cloud consumer. On the contrary, if Cloud resources are consumed based on the Charge-Per-Use model those resources (e.g. dedicated servers) are subscribed in advanced for a period of time. Usually, payment is expected to be upfront and commitments are rather long-term (e.g. monthly or yearly). Beside those distinctions also a combined model of both previous pricing approaches might be possible. In addition to those pricing models, [81] considers the case that some Cloud services deliver resources for *free* but those are usually limited in time or amount of usage, like for example Amazon EC2 is free in terms of specific time and amount limitations [85].

The basis for monetary values charged by the service provider in their pricing models are several characteristics [6] [68]. This can be for instance the type of Cloud resource consumed (e.g. computing resources or software services), level of SLA assurance, support level, or the actual physical location of the computing resources (e.g. Amazon offers different Resource location at different prices) [85].

Because, service providers deliver their offerings in a variety of pricing models which influence cost analysis and overall cloud efficiency (conversion of the type of expenses) this decision *Select Pricing Model* has been identified in Decision Point 4 - Select Service Provider / Offering.

Possible outcomes
In case of this decision one of the following outcomes is possible to be selected:

- *Free*
- *Pay-Per-Use*
- *Pay-Per-Unit*
- *Charge-Per-Use (Subscription)*
- *Combined Pricing Model*

Relations with Tasks
As stated above the pricing models offered by the Cloud provider differ on several characteristics and depend on other decisions within this Decision Point 4 but is not affected by any existing task of the cDSF. But in turn the Task Cost analysis and this decision affect each other as depicted in Figure 3.25.

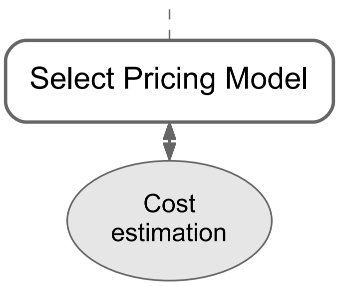


Figure 3.25.: Tasks Related to Decision 4.6 - Select Pricing Model

Decision 4.7 - Define Physical Cloud Resource Location

Description
Considerations about the actual physical location of the provided Cloud resources are a relevant aspect in selecting a service provider and/or offering for enterprises. Depending on location and their physical storage data has to be compliant to regulations and laws exposed by jurisdiction of countries or even single regions [80] [87]. A common example is that EU enterprises are required by legal constrains in terms of data privacy regulations

to store data within the European borders [56]. However, since data are obligated to conform to the law where they are physically stored even a US company transferring data to a service provider with its physical location in an EU country would face issues as the country's data protection restricts or even prohibits the data transferred back to the US [78].

This decision might be constrained by other decisions, for example in case of on premise hosting such concerns would be obsolete if not considering them on a company group level with companies in several countries. Issues surrounding the physical location of Cloud resources have also been raised by other decision support approaches for application migration to the Cloud [6] [80], and hence this decision *Define Physical Cloud Resource Location* has been identified as a part of Decision Point 4 - Select Service Provider / Offering.

Possible outcomes

The decision's outcome has to be evaluated based on various requirements individual for the application migrated to the Cloud comprising regulations associated with the company itself (e.g. compliance regulations) as well as respective legal regulations that possibly apply. Thus, similar to Decision 4.5 the outcome is an evaluated physical location for the Cloud resources appropriate and suitable based on individual stated requirements.

- *Evaluated Physical Cloud Resource Location*

Relations with Tasks

In case of this decision the same tasks as in Decision 4.3 are related with exception of effort estimation like depicted in Figure 3.26.

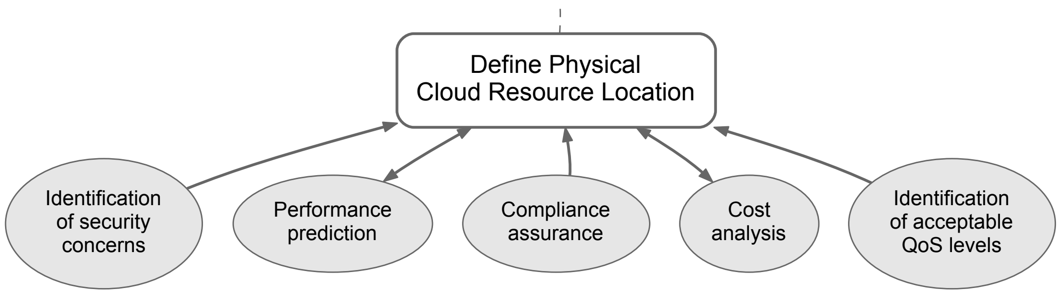


Figure 3.26.: Tasks Related to Decision 4.7 - Define Physical Cloud Resource Location

Decision Classification

In a sense, those decisions identified in this Decision Point 4 can all be classified specifically to Cloud migrations, however, considering just the bare decisions to be made some of them could also be necessary in application migrations not considering a Cloud Computing environment as shown in Table 3.4. Obviously, Decision 4.1 and Decision 4.2 are Cloud specific since they define both main distinctions in Cloud Computing as introduced in Section 2.1. Also, Decision 4.6 regarding pricing models is with respect to its outcomes highly related to Cloud services. In turn, Decision 4.3 and Decision 4.4 address outsourcing aspects, which are not Cloud migration specific. Similarly, considerations in terms of Decision 4.5 and Decision 4.7 apply in cases where the target environment is non-Cloud as well.

Decision	***Classification***
Decision 4	☁ ▷
Decision 4.1	☁
Decision 4.2	☁
Decision 4.3	☁ ▷
Decision 4.4	☁ ▷
Decision 4.5	☁ ▷
Decision 4.6	☁
Decision 4.7	☁ ▷

☁ : Cloud Migration Specific
▷ : Application Migration in General

Table 3.4.: Decision Classifications in Decision Point 4

3.2.5. Relationships between Decisions

This section refers to how the previously identified decisions within the four decision points are connected with each other. The cDSF (see Section 2.4) already shows that all decision points influence each other and thus the elaborated decision are assumed to influence each other as well. The focus is on if a certain decision either *influences* ($\succ$) or even *determines* ($\vdash$) another decision, however, only relationships on the level of decisions are considered.

In the following sections the relationships are discussed for each decision point and the associated tabular representations are append in Appendix A.1.

Influences of Decision in Decision Point 1

An overview of the relationships investigated in Decision Point 1 - Distribute Application is represented in Table A.1. All decisions are strongly related within their decision point itself as well as to other decisions in the eDSF and predominately other decisions are influenced.

Within the decision point itself all decisions are influenced by each other expect of Decision 1.3 that determines Decision 1.1 and Decision 1.2. It is assumed that based on the selection of certain application components both the layer(s) and tier(s) of the application where those components are arranged on can be identified. The selection of a migration type (Decision 1.4), however, is only influenced by Decision 1.3 since application components are considered in Type I as well as in Type II migrations. Due to the fact that application layers and tiers correspond to each other (e.g. functionality of the data layer is usually implemented on the data tier) they are also influence each other. Furthermore, either the selection of application layer(s) or tier(s) will impact which components and migration types can be selected. Finally, regarding Decision 1.4 the selection of a migration type influences all other three decisions as it constrains their possible selections (e.g. Type I and Type II migration constrain application parts considered of migration) or a selection gets obsolete, e.g. in case of Type III or IV migrations where the whole application is moved to the Cloud.

With regard to Decision Point 2 - Define Elasticity Strategy only to decisions are influences. All decisions of Decision Point 1 are related to the the scalability level selection (Decision 2.1), because the level depends on who the application is distributed (e.g. which components or application tier(s) are migrated) and if selection constraints are possibly raised. For example, if an application is entirely wrapped in a virtual machine and moved to the Cloud the lowest entity to scale is the VM and scaling is only possible on the virtualization level and the levels below (e.g. hardware level) but not on them above (e.g application level). Decision Point 1 also influences the type of scaling (Decision 2.2) as those are constrained by the selection of the scalability level as discussed in Section 3.2.2. Both Decision 2.3 and Decision 2.4 are not assumed to be related to decisions of Decision Point 1.

Decision 3.1 is assumed to be influenced by Decision 1.4, because in case the application is wrapped in a VM (Type III migration) native multi-tenancy characteristics are not possible to be utilized, however, this is more likely in case of application "cloudification" in Type IV migrations. Regarding the selection of a multi-tenancy architecture (Decision

3.2) all decision of Decision Point 1 influence, because they separately consider the application and database multi-tenancy with respect to several Cloud resource levels. By this means, the migration of an only database component for example influences the selection of an MTA by limiting possible selections.

The selection of the Cloud deployment model (Decision 4.1) is assumed to be unaffected by Decision Point 1 hence the application distribution does not concern if the Cloud resources are for instance private or public. In turn, the selection of the Cloud service model (Decision 4.2) is arguably influenced by all decisions concerning the application distribution. If for example single application components are replaced in terms of the Cloud migration likely SaaS or PaaS models are more suitable than IaaS. At a first glance Decision 1.4 could be assumed to influence Decision 4.3 regarding to the Cloud hosting location, but since the migration type distinguishes "traditional" computing resources and Cloud resources and not the actual location of those Cloud resources, a relationship can not be stated. The same applies to all other possible relationships between decision in Decision Point 1 and Decision Point 4.

Influences of Decision in Decision Point 2

Decisions in the Decision Point 2 - Define Elasticity Strategy are less intensively related to other decisions in the eDSF by comparison to those in Decision Point 1 as shown in Table A.2.

Interdependences within the decision point itself are weak. However, Decision 2.1 and Decision 2.2 are related to each other as the selected scalability level might constrain the selection of the scaling type (this holds vice versa) as already mentioned in the previous section and discussed in Section 3.2.2. Furthermore, Decision 2.3 and 2.4 are related and mutually influenced due to the fact how elasticity is automated constrains the possible scaling triggers as for example semi-automatic scaling (i.e. rule-based automatic) is obviously triggered by events, which are fired as soon as defined thresholds exceed. The same holds for the selection of scaling triggers vice versa, i.e. for instance proactive scaling is assumed to be possible only in case of real automatic scaling.

With regards to decisions in Decision Point 1 the first three can be indicated as influenced by Decision 2.1 and Decision 2.2. Because Decision 1.1 - Decision 1.3 identify certain application components and/or architectural layer(s) and tier(s) of the application, this primary impacts the level where scalability can possibly be applied and subsequently affects the selection of the scaling type.

In case multi-tenancy requirements are exposed, Decision 2.1 and Decision 2.2 influence

Decision 3.2 in Decision Point 3. This assumes the scalability level should be chosen appropriately to the level where multi-tenancy is applied. If for example multi-tenancy is applied on the level of VMs also scaling should be at least ensured on this level to manage the amount of resources appropriately in terms of varying workloads. Thus, this is again influenced by the selected type of scaling (Decision 2.2).

Finally, in Decision Point 4 only Decision 4.2 concerning the Cloud Computing service model is influenced by Decisions 2.1 and Decision 2.2. This assumes that the decision considering the scalability level influences the Cloud service model which is chosen best.

Influences of Decision in Decision Point 3

Principally, Decision Point 3 - Define Multi-Tenancy Requirements influences decisions related to the application distribution (Decision Point 1) as illustrated in the relationship overview for this decision point in Table A.3.

Within the decision point itself both decisions are strongly related due to the fact that if a selection in Decision 3.2 is made it determines Decision 3.1 to be either native multi-tenancy or multiple instance multi-tenancy. In turn, the selection of a multi-tenancy kind in Decision 3.1 influences Decision 3.2 as it restricts the number of possible outcomes.

With regards to decisions in Decision Point 1 the multi-tenancy kind (Decision 3.1) does not affect those decisions as for example the selection of which application layer(s) are migrated is not related to the multi-tenancy kind. However, a selection in Decision 3.2 is influenced, because it refers to application architecture characteristics as discussed in terms of this decision in Section 3.2.3. Furthermore, since this decision considers both application and database multi-tenancy requirements a selected multi-tenancy architecture influences and may even determine selections of decisions regarding the application distribution. For instance, if multi-tenancy is expected to be applied to the application and its data resources, this influences which components are possible to be selected as well as this consequently provides information about the application tier(s) and layer(s) that are likely affected. In Section 3.2.5 this relationship has already been pointed out to work vice verse.

Section 3.2.5 shows how Decision 2.1 and Decision 2.2 influence the selection of multi-tenancy architectures (Decision 3.2) and this works vice versa as well. If multi-tenancy is applied on a certain level of an application and/or its data components, the scalability level should be appropriate to not neglect performance goals or QoS. Other decisions in Decision Point 2 are not related.

With regard to the fourth decision point only Decision 4.2 is assumed to be influenced by Decision 3.2. Again, this is assumed due to the fact that a selected MTA indicates the level of stated application and/or data resources multi-tenancy requirements and Cloud resources provided by service models differ in how precise and flexible these services are and in turn which required multi-tenancy level can be supported or not. Apart from that, no other influences to Decision Point 4 are stated.

Influences of Decision in Decision Point 4

The overview in Table A.4 shows how decisions in Decision Point 4 - Select Service Provider / Offering are related within the eDSF. However, this decision point comprises the largest number of identified decisions and they are predominantly related within their own decision point itself and only merely related to others.

Decision 4.3 and Decision 4.4 influence Decision 4.1 as for example selections considering on premise hosting and holding the responsibility of ownership of Cloud resources might indicate the private Cloud deployment model. The same can be claimed vice versa, because a selected deployment model, the responsibility roles, and in addition an identified resource location impact the selected hosting solution in Decision 4.3 to be on premise, off premise or a hybrid approach. In turn, which responsibilities a Cloud consumer holds (Decision 3.4) is mainly influenced by the deployment model (e.g. public Clouds are supposed to be owned and operated by a service provider), the service model (e.g. consuming a IaaS allows to control some aspects of management like firewall settings), and the hosting solution (e.g. external hosted resources are at least operated by a third party). The selection of a pricing model (Decision 4.6) is influenced only by Decision 4.2, but in turn the kind of pricing model is not limited to certain deployment models or hosting solutions. On the other hand the selected pricing model possibly restricts selections in terms of the Cloud service model in turn. Decision 4.5 and 4.7, however, are mostly unrelated in the overall framework. Only within the own decision point Decision 4.7 is influenced by the selected Cloud vendor because not every vendor is feasible to provider resources at any identified location appropriate for requirements of the Cloud consumer. But the where Cloud resources can be hosted (Decision 4.3) is obviously influenced by the identified location.

According to Decision Point 1 only two relationships can be mentioned. First, a selected Cloud service model influences the selection of the migration type (Decision 1.4), e.g. SaaS can probably be used in terms of Type I but not for Type II and III migrations where PaaS or IaaS are likely needed. Second, since the migrations types consider local data center and Cloud resources Decision 4.3 can impact this decision.

In Decision Point 2 only Decision 4.2 is influencing the level of scalability due to the fact that IaaS resources are scaling on a lower level according to the scalability levels than for example Cloud resources provided by a SaaS model. More relationships could not be pointed out.

In the final Decision Point 3 again only Decision 4.2 influences Decision 3.2. Referring to the previous influence section, it is assumed that Cloud resources on an IaaS level are able to fulfill multi-tenancy requirements on another level than for example SaaS. A Cloud consumer migrating an application using IaaS might applied multi-tenancy from this level up to the top level if the application is adjusted accordingly. In terms of SaaS the service provider determines multi-tenancy.

3.3. Additional Tasks

This section refers to the identification of additional tasks within the conceptual Decision Support Framework. In order to perform appropriate decision-making some of the decisions elaborated in the previous section could not be addressed sufficiently by the existing tasks delivered by the cDSF. As a result, new tasks have been identified which will be introduced in this section.

3.3.1. Workforce Capabilities Identification

So far, the cDSF attended to the aspect of human labor merely in the Task Effort estimation. Adopting Cloud Computing certainly aims to provision Cloud resources in an easy and mostly automatic manner for the Cloud consumer according to its definition introduced in Section 2.1. However, certain scenarios using Cloud services make the Cloud consumer also act in additional roles rather then just using the service, e.g. in the case of on premise provisioning of Cloud resources. This likely requires human capacity with sufficient knowledge and capabilities regarding Cloud Computing to be in-house the Cloud consumer in order to cope with activities like manage, maintain, and/or operate the provisioned service.

In the previous section this task has been identified to be related to *Decision 2.3 - Select Elasticity Automation Degree*, *Decision 4.3 - Define Cloud Hosting*, and *Decision 4.4 - Define Roles of Responsibility*. Determined by the outcome of those decisions an appropriate skill level (i.e. capabilities) regarding certain Cloud activities is required. In case of the first decision mentioned this could be knowledge about Cloud monitoring and management to decide the right resource scaling. The latter two decisions show activities in the area of Cloud service provisioning where operating and management of either single services or even the whole Cloud infrastructure has to be dealt with by the staff of the Cloud consumer.

With regards to enterprise constraints in human capacity and capabilities regarding Cloud Computing, which are possibly known in advanced or stated for the migration approach, this task also affects the previous mentioned decisions. Furthermore, additional labor acquired or educated to fulfill required skills has to be considered in terms of both one-off expenses (e.g. training and education costs) as well as running labor costs associated with the desired application solution migrated to the Cloud. Thus, cost analysis is affected by this task and in case cost limitations are defined it also affects in return.

3.3.2. Vendor Evaluation

During the elaboration of Decision 4.5 - Select Cloud Vendor this task has been identified. The initial cDSF does not provide a task supporting the identification of a Cloud vendor appropriate, hence, this task *Venodr benachmark* has been identified to decide on various vendors of Cloud services.

The authors in [81] present organization-specific aspects of participants in a Cloud environment, i.e. such as the vendor who provides the Cloud services to the consuming party. This regards aspects in the distinction of *reputation* (external view) incorporating e.g. reference projects, benchmarks, certificate, reports, etc. and *capability* (internal view) evaluating resources, knowledge, technical and business skills. Based on this data gathered for various candidate Cloud vendors, a proper evaluation can be performed to identify an appropriate vendor in terms of social or “soft” facts beside those technical requirements that are predominate in the tasks of the cDSF.

Another view regarding vendor comparison is shown by the authors in [82], which refer to a vendor characterization tool provided by Info-tech research based on results of a vendor survey [83]. This Excel-Tool considers attributes like Available features, Affordability, Usability, Vendor viability, and Support quality to point out an appropriate Cloud vendor (currently out of a list of eight vendors). But those attributes indicate that a vendor comparison in this case mingles hard technical and functional facts considered by tasks like Cost analysis and Identification of acceptable QoS levels in the cDSF with soft facts like Vendor viability or Usability.

In terms of the elaborated Decision Support Framework this task is expected to evaluate a Cloud vendor to the approach in [81], which is deemed to be more suitable for a discrete vendor evaluation rather than the latter approach that mingles up aspects already considered by other parts of the framework.

3.3.3. Application Analysis

The previous section, especially Section 3.2.5, indicates the major relevance of Decision Point 1 - Application Distribution by exposing its relationships to other decisions and tasks in framework. Based on tasks like Workload profiling and Performance prediction decisions concerning on how the application is distributed can be made [16] [82]. However, as a prerequisite the application desired to be migrated to the Cloud has to be analyzed first in order to perform such tasks or even make proper decisions in Decisions Point 1. The cDSF is currently missing a task managing this effort, which is why the Task *Application analysis* has been identified to provide characteristics about the existing application (e.g. architecture, programming language, etc.).

Related approaches in decision support for application migration to the Cloud (see Section 2.3.3) also mention the process of investigating the application and utilize findings for further decision-making. For example Cloudstep [6] creates an application profile considering on the one hand application usage characteristics (e.g. amount and origin of user requests to classify usage patterns), which can be used by tasks like Workload profiling. On the other hand a technical profile is used to investigate the application architecture and used technologies used. Such findings could be processed in software architecture diagrams, flowcharts, or class diagrams to support decision-making in Decision Point 1.

In this sense, this task aims to provide characteristics of the existing application for decisions and tasks within the eDSF. The decisions the task is related to are discussed in Section 3.2. Regarding existing tasks the application analysis is related to workload profiling, performance prediction, and identification of acceptable QoS levels.

3.4. The Elaborated Decision Support Framework

As a result of this chapter the conceptual Decision Support Framework has been elaborated and refined within its both domains decisions and tasks which results in the *elaborated Decision Support Framework* (eDSF). In each of the four decision points several decisions and their possible outcomes have been identified based on a literature investigation. Furthermore, additional tasks have been elaborated and the relationships between decisions and tasks as well as among decisions themselves have been discusses.

The elaborated Decision Support Framework consists of the decisions shown in Table 3.5, the tasks in Table 3.6, their relationships discussed for each decision in Decision Point 1 - Decision Point 4 (see Section 3.2), and the relationships between decisions that are presented in Section 3.2.5.

As part of the elaboration and refinement the framework gained in complexity and due to format limitations of this Master's thesis an illustration of the eDSF can not be provided (like the cDSF has been depicted in Figure 2.5). However, Chapter 5 presents a prototypical implementation to provide a visualization of the eDSF.

Decisions
Decision 1 - Application Distribution
Decision 1.1 - Select Application Layer
Decision 1.2 - Select Application Tier
Decision 1.3 - Select Application Components
Decision 1.4 - Select Migration Type
Decision 2 - Define Elasticity Strategy
Decision 2.1 - Define Scalability Level
Decision 2.2 - Select Scaling Type
Decision 2.3 - Select Elasticity Automation Degree
Decision 2.4 - Select Scaling Trigger
Decision 3 - Define Multi-Tenancy Requirements
Decision 3.1 - Select Kind of Multi-tenancy
Decision 3.2 - Select Multi-tenancy Architecture
Decision 4 - Select Service Provider / Offering
Decision 4.1 - Select Cloud Deployment Model
Decision 4.2 - Select Cloud Service Model
Decision 4.3 - Define Cloud Hosting
Decision 4.4 - Define Roles of Responsibility
Decision 4.5 - Select Cloud Vendor
Decision 4.6 - Select Pricing Model
Decision 4.7 - Define Resource Location

Table 3.5.: All Decisions of the Elaborated Decision Support Framework

Tasks
Workload profiling
Performance prediction
Effort estimation
Cost analysis
Identification of acceptable QoS levels
Compliance assurance
Identification of security concerns
Workforce capabilities identification
Vendor evaluation
Application analysis

Table 3.6.: All Tasks of the Elaborated Decision Support Framework

4. Evaluation of the Decision Support Framework

This chapter addresses the evaluation of the elaborated Decision Support Framework (eDSF) developed in the previous chapter. Insights into the evaluation process as well as a discussion of findings are provided in the following sections.

4.1. Evaluation Procedure

For evaluating the *elaborated Decision Support Framework* established in the previous chapter the research method of a survey is chosen because it is a systematic and standardizes procedure for data collection on individuals [88]. Its results should then provide a reasonable peer review of the eDSF to evaluate the elaborated state.

4.1.1. Objectives

The elaboration of the cDSF has been performed according to the elaboration process discussed in Section 3.1 based on a literature study, which results in several extensions in terms of decisions and tasks in comparison to the initial cDSF introduced in Section 2.4. Now, with respect to the given objectives of this Master's thesis in Section 1.1, the eDSF is evaluated to yield confirmation in terms of the following objectives:

EO. 1 *Evaluate the suitability of the eDSF Decision Points*

EO. 2 *Evaluate the completeness of elaborated decisions within the respective Decisions Point of the eDSF*

EO. 3	*Evaluate the suitability of the elaborated decisions within each Decision Point of the eDSF*

4.1.2. Instrument

The evaluation is performed by a web-based survey using a questionnaire attached in Appendix A.2. The questionnaire has been designed respecting conventional wisdom in terms of creating questionnaire like group questions on the same topic and process them from general to specific, etc. [89] to address the stated evaluation objectives.

A mixture of open (i.e. individual responses to answers in own words) and closed questions (i.e. responses are selected from a set of choices) has been applied according to the respective objectives. To realize the web-based survey an open source survey application called *LimeSurvey*[1] has been used, installed on an OpenShift[2] PHP PaaS Instance.

4.1.3. Participant Group

Decision support for application migration to the Cloud is a topic, which is not solely addressed by research but also Cloud service provider or consultancy enterprises are present in this field as stated in the introduction of this Master's thesis. In this sense, the participant group for this survey is expected to cover these different points of view.

The web-based survey link has been sent out to prospective participants in research as well as enterprise practitioners by the supervisor and the author of this thesis. All participants are assumed to have either theoretical or practical experience and knowledge in the field of application migration to the Cloud so that the evaluation is expected to be made by expert opinions.

4.1.4. Participation Overview

E-Mails with the survey link have been sent to prospective participants on 7th February 2014 and the survey was available for participation unit the 15th of February 2014. In this time a total of 14 participants started the web-based survey after reading the introduction and a total of six participants completed the questionnaire (a completion

[1] LimeSurvey: http://www.limesurvey.org
[2] OpenShift: https://www.openshift.com

rate of 42,9%). The further evaluation takes into account only the complete replies to the questionnaire.

Participants from the field of research as well as practitioners attended the survey as expected. With regard to information collected about the participants job role four can be assigned to the field of research (e.g. research associates) and two practitioners attended (e.g. IT project manager). The participant experience level in terms of IT in general has been rated in average with 7 and their specific Cloud Computing experience has been rated in average with 6, both on a scale from 0 to 9 (with 0 = no experience, 1 = very poor experience, [...], 9 = excellent experience). Hence, both rates are in the upper third range the overall results are arguably be evaluated through an expert's peer-review.

4.2. Findings

The findings in this section take into account only complete processed survey questionnaires to gain consistency in the number of responses through all survey questions. As expected and stated previously participants from research and practitioners took part in the survey. However, due to the small number of complete responses available for consideration a differentiation of findings regarding both participant groups separately is not made. Unless otherwise stated, all numeric results in this section respect the following scale:

0 = Not at all

1 = Slightly

2 = Moderately

3 = Very

4 = Absolutely

Each table with this abscissa axes scale shows the number of participants rated the respective value and in terms of evaluation a weighted average ($\bar{x}$) is calculated for each question.

Question	*Ratings*					
	0	*1*	*2*	*3*	*4*	$\bar{x}$
Is the differentiation between Decisions Points and Tasks understandable?	0	0	0	3	3	3.5
Is the meaning of the presented Decision Points understandable?	0	0	0	4	2	3.3
Is the meaning of the presented Tasks understandable?	0	0	1	3	2	3.2

Table 4.1.: Evaluation of the General Understandability of the Decision Support Approach

4.2.1. Understandability of the General Decision Support Approach

At first, the survey gives a brief introduction of the conceptual Decision Support Framework, its elements, and concept of considering decision points, tasks, and their relationships. To evaluate the general understandability of this decision support approach questions are formulated regarding the meaning of decision points and tasks as well as their differentiation is understandable. The results are shown in Table 4.1.

In consideration of the overall distinction of decision points and tasks made in this decision support approach is very close to absolutely understandable as rated with 3.5 in average. The individual meaning of each decision point and task within the cDSF is less but still very well understood by the participants. As a result, the decision support approach of the cDSF and its elements meanings is inferred to be understood and the following evaluation results of the eDSF can be assumed reasonable.

4.2.2. Relevance and Completeness of Elaborated Decisions

The survey questions have been grouped by decision point and to address *EO. 2* and *EO. 3* findings for each elaborated decision in terms of their relevance and if their meaning is understandable is presented in context of the respective decision point. In addition, decisions missed by the participants are briefly discussed.

Question	*Ratings*					
	0	***1***	***2***	***3***	***4***	$\bar{x}$
Is the meaning of Decision ... understandable?						
Decision 1.1	0	0	1	2	3	3.3
Decision 1.2	0	1	1	2	2	2.8
Decision 1.3	0	2	1	0	3	2.7
Decision 1.4	0	0	1	1	4	3.5
Is Decision ... relevant in Decision Point 1?						
Decision 1.1	0	1	1	2	2	2.8
Decision 1.2	0	1	0	2	3	3.2
Decision 1.3	0	2	1	1	2	2.5
Decision 1.4	0	0	0	2	4	3.7

Table 4.2.: Evaluation of Decisions in Decision Point 1

Decision Evaluation in Decision Point 1

Table 4.2 provides the evaluation results for decisions in Decision Point 1. Basically, all elaborated decisions are understood in their meaning as they are rated 2.7 or higher in average (i.e. at least very understandable). Although, Decision 1.2 (application tiers) and Decision 1.3 (application components) have been partially rated only slightly understandable, on the other hand they have been very clear for other participants. This could be due to the limited description (e.g. without examples, etc.). But in turn Decision 1.4 (migration type) also considering application components in migration type I is understood best of all elaborated decisions.

All elaborated decisions are relevant to Decision Point 1. In terms of considering an application from a physical or logical perspective the former is rated more relevant with regards to the application distribution. Migrating an application based on a physical distribution (e.g. a separation of application and/or database servers) can be performed more easily by utilizing migration type III and migrate the whole software stack of the server to a VM. As already mentioned this is assumed to be the most accessible Cloud migration approach [35]. To the contrary, Decision 1.3 shows the least relevant level of application distribution suggesting applications are less migrated on this fine granular level of components. Decision 1.4 is rated most relevant of all decisions when considering the application distribution, which has already been indicated in [16].

Question	***Ratings***					
	0	***1***	***2***	***3***	***4***	$\bar{x}$
Is the meaning of Decision ... understandable?						
Decision 2.1	0	0	0	3	3	3.5
Decision 2.2	0	1	0	2	3	3.2
Decision 2.3	0	0	2	0	4	3.3
Decision 2.4	0	0	2	0	4	3.3
Is Decision ... relevant in Decision Point 2?						
Decision 2.1	0	0	0	2	4	3.7
Decision 2.2	0	0	0	2	4	3.7
Decision 2.3	0	0	3	0	3	3.0
Decision 2.4	0	1	1	0	4	3.2

Table 4.3.: Evaluation of Decisions in Decision Point 2

Participants missed decisions regarding the specific distribution of the application data as well as to detect the application architecture paradigm (e.g. SOA) and if this has to or should change during the cloud migration. The former consideration appears reasonable as application data can be classified differently in terms of data private regulations. Within the eDSF this can be addressed by considering both Tasks Compliance assurance and Identification of security concerns, which deliver inputs for Decision Point 1 and thus the application distribution can be decided to follow the hybrid deployment model to keep classified data in-house. The latter noted missed decision considering the application architecture paradigm or style (e.g. like SOA is a known architectural style) has to be addressed more accurately as there are several areas of architecture styles regarding applications [90]. For detecting the current architecture style the new identified Task Application analysis can be referred. Whether the current style must or can change arguably depends for example on how it is desired to be distributed in terms of migration types (e.g. in a type III migration the whole application is wrapped in a VM whereas a type IV migration considers a re-construction where application changes are likely more extensive). Additionally, the meaning of an application architecture style has to be defined more accurately, but it is arguably possible to consider such a decision in terms of Decision Point 1.

Decision Evaluation in Decision Point 2

Findings in Decision Point 2 are depicted in Table 4.3 and show an overall higher rating in both decision understandability as well as relevance of the elaborated decisions in comparison to Decision Point 1. Especially Decision 2.1 (scalability level), which is composed out of various literature works in this thesis, has been understood best by the participants.

The same holds for the decision relevance perspective where overall each decision is rated at least very relevant in average, especially Decision 2.1 and Decision 2.2 (scaling type) have been rated very to absolutely relevant in consideration of the scalability strategy of an application migrated to the Cloud. Arguably, results show that these two decisions contemplating *where* scalability is applied in the application stack and *which* type of scaling is used cover the major thoughts in terms of defining the application scalability strategy. But also decisions on a more detailed level like Decision 2.3 (elasticity automation degree) and Decision 2.4 (scaling trigger) are very relevant rated in average, even though opinions are divided on both are either absolute or moderate to slightly relevant for the participants.

Participants additionally noted that decisions about the elasticity might have to be different regarding the application distribution for example in terms of layers or components. This is true and the eDSF addresses this issue indirectly through its relationships as Table A.1 shows influences of decision in Decision Point 1 on Decision 2.1 and Decision 2.2. Further notes for missing decisions mentioned by the participants consider costs, which are assessed by the Task Cost analysis and available evaluation solutions towards analyzing scaling techniques to select. This can rather be seen as an independent decision or task than tool or methodology utilized to actually make the decision of selecting an appropriate scaling type for example.

Decision Evaluation in Decision Point 3

The participants' results of evaluating multi-tenancy decision in Decision Point 3 are shown in Table 4.4. They indicate that the basic differentiation of native multi-tenancy and multiple instance multi-tenancy in Decision 3.1 is much better understandable as the successor Decision 3.2 establishing several multi-tenancy architectures in terms of considering application and data multi-tenancy separately. The latter does not indicate a trend as participants spread their rates between only slightly to absolutely understandable. This is probably due to the very brief introduction to this more comprehensive decision.

Question	*Ratings*					
	0	*1*	*2*	*3*	*4*	$\bar{x}$
Is the meaning of Decision ... understandable?						
Decision 3.1	0	0	0	3	3	3.5
Decision 3.2	0	1	2	1	2	2.7
Is Decision ... relevant in Decision Point 3?						
Decision 3.1	0	0	0	3	3	3.5
Decision 3.2	0	1	1	0	4	3.2

Table 4.4.: Evaluation of Decisions in Decision Point 3

Both elaborated decisions are also considered very relevant in average in terms of Decision Point 3. Comparable to their understandability Decision 3.1 is rated more relevant overall but Decision 3.2 tends to be absolute relevant despite that the results are spread over the range. This infers the relevance of considering multi-tenancy independently for application logic and application data as well as such multi-tenancy requirements can be applied on different levels of the respective (server) software stack.

In terms of missing decisions in this decision point participants noted that the case of an application consisting of both multi-tenant aware and non-multi-tenant aware parts should be considered. In addition, multi-tenancy requirements perhaps vary with regards to the application distribution and thus they should be stated separately for each application part. As this decision point comprises requirement definitions for multi-tenancy such requirements can obviously be stated for different application parts regarding application distribution, which actually depends on decisions in Decision Point 1. With regard to the first remark it can be stated that a previous separation of multi-tenant aware and unaware parts is not necessary since this decision point considers requirements definition for the target application. This means existing multi-tenancy requirements remain unaffected if they are not required to change and additional multi-tenancy requirements for certain application parts are included. The identification of existing multi-tenancy requirements is part of the Task Application analysis, which is related to decisions in Decision Point 3.

Decision Evaluation in Decision Point 4

The evaluations results for the elaborated decisions in Decision Point 4 are consolidated in Table 4.5. In average again all elaborated decisions are well understood by the par-

Question	***Ratings***					
	0	***1***	***2***	***3***	***4***	$\bar{x}$
Is the meaning of Decision ... understandable?						
Decision 4.1	0	0	0	0	6	4.0
Decision 4.2	0	0	2	0	4	3.3
Decision 4.3	0	0	2	0	4	3.3
Decision 4.4	0	0	2	2	2	3.0
Decision 4.5	0	1	0	1	4	3.3
Decision 4.6	0	0	0	1	5	3.8
Decision 4.7	0	0	1	2	3	3.3
Is Decision ... relevant in Decision Point 4?						
Decision 4.1	0	0	0	0	6	4.0
Decision 4.2	0	0	1	1	4	3.5
Decision 4.3	0	0	1	1	4	3.5
Decision 4.4	0	0	3	2	1	2.7
Decision 4.5	0	1	0	1	4	3.3
Decision 4.6	0	0	1	1	4	3.5
Decision 4.7	0	0	1	1	4	3.5

Table 4.5.: Evaluation of Decisions in Decision Point 4

ticipants and only Decision 4.4 (roles of responsibility) does not tend to be more than very understandable in its rating results.

With regard to the relevance of each decision in this decision point it is unexpected that Decision 4.2 (service model) is in some opinions not absolutely relevant in terms of the service provider and offering selection unlike Decision 4.1 (deployment model) although it is one of the major classifications in Cloud Computing. This could be possibly explained due to the various kinds and terminology of Cloud services beyond NIST as briefly mentioned in Section 2.1. All other decisions tend to be rated absolutely relevant except of Decision 4.4, which is predominately rated moderately relevant (in average only 2.7). This can possibly be put down to the fact that responsibility distribution is more related to the organizational perspective of Cloud Computing and organizational changes related to Cloud adoption or migration are not always taking into account [14].

As missing decisions in this decision point participants again pointed out that decision might have been considered multiple times with regards to the application distribution which is probably true. Other aspects mentioned are security with is covered by the eDSF Task Identification of security concerns and considering scaling solutions the Cloud vendor is providing in his offering. Hence, this decision support approach does not address the selection of a best fitting Cloud offering and rather point out a more general perspective of Cloud migration (see Section 2.4) this aspect is considered within Decision Point 2 - Define Scalability Strategy.

4.2.3. Relevance of Decision Points

The survey questionnaire is grouped regarding the four decision points. To address *EO. 1* each decision point is questioned in terms of its relevance for application migration to the Cloud. The results are shown in Table 4.6.

Question	***Ratings***					
	0	***1***	***2***	***3***	***4***	$\bar{x}$
Is Decision Point ... relevant in terms of application migration to the Cloud?						
Decision Point 1	0	0	1	1	4	3.5
Decision Point 2	0	0	1	3	2	3.2
Decision Point 3	0	0	2	0	4	3.3
Decision Point 4	0	0	0	0	6	4.0

Table 4.6.: Evaluation of Decision Points

Results show relevance of all decision points the elaborated Decision Support Framework comprises. Especially, Decision Point 1 and Decision Point 4 are rated highly relevant which is probably associated with the fact that both consider main elements of the Cloud migration (i.e. the application itself which is desired to be migrated and the service provider and offering characterizing the target environment). Decision Point 2 and Decision Point 3 are rated less but still in average are more than very relevant. These results can be interpreted as the basic Decision Points 1 and Decision Points 4 are necessary and without them an application migration to the Cloud cannot be discussed at all whereas Decision Point 2 and Decision Point 3 are optional considerations to exploit all capabilities of Cloud Computing.

5. Implementation of the Decision Support Framework

This chapter deals with the realization aspect of this Master's thesis with the focus on a prototypical implementation of the elaborated Decision Support Framework. As another maturing step on the way to a Decision Support System the implementation objective is to visualize the eDSF data in order to make them available and accessible for stakeholders to support their decision-making while considering the application migration to the Cloud. First, a brief listing of requirements regarding the prototypical implementation is given. Next, data visualization approaches are considered followed by an introduction of used technologies and frameworks and depicted system architecture. Finally, the prototypical implementation is presented by several screenshots showing how data of the eDSF has actually been visualized.

5.1. Requirements Analysis

Since the objective is a prototypical implementation, the requirements analysis in this section provides a short list of defined functional and non-functional requirement assumed relevant in case of the DSF. The following Table 5.1 lists the defined functional requirements and Table 5.2 shows identified non-functional requirements.

5.2. Specification and Design

5.2.1. Data Visualization

In terms of visualization the eDSF has to be considered on a more general level with regards to its data structures. The eDSF consists of *nodes*, elements like decision points, decisions, tasks, and outcomes and *links* that represent relationships between the node

No.	***Requirement***	***Description***
FR-1	Visualize the Decision Support Framework	Rendering data of the eDSF in a human perceptive manner to provide supporting information for decision-making in terms of application migration to the Cloud.
FR-2	Support defined Visualization Scenarios	For each Visualization Scenario defined in Section 5.2.1 at least one rendering solution has to be provided.
FR-3	Highlight relationships of single elements by selection	With selecting an element (e.g. a decision or task) highlight its relationships to other elements in the eDSF and provide further information.

Table 5.1.: Functional Requirements for the Prototypical Implementation

No.	***Requirement***	***Description***
NFR-1	Web-based implementation	In terms of portability, cross-platform support, and an easy-to-use common interface the prototypical implementation has to be created using web programming language and related technologies.
NFR-2	Application of state-of-the-art information technology	For realization current state-of-the-art technologies have to be used, e.g. HTML5[1], JavaScript[2], etc.
NFR-3	Usability	The implementation has to focus on visualizing the eDSF and the Visualizations Scenarios appropriately. Therefore, the user interface has to be well structured and controls have to be accessible intuitive and user-friendly.

Table 5.2.: Non-Functional Requirements for the Prototypical Implementation

elements. For instance, the relationships between a decision point and its decisions are such links or the relationships between decisions and tasks. A representation of elements and their relationships (i.e. nodes and links) is generally called a *graph* [39].

Considering the eDSF data two types of data structures can be detected. First, all node elements (i.e. decision point, task, etc.) can be represented in a *hierarchical graph* or *tree graph* with several subtrees (e.g. each decision point or decision) like depicted in Figure 5.1. On the other hand, when taking into account the affecting and influencing relationships between decisions and tasks this forms a more complex *network graph* structure like the cDSF already does (see Figure 2.5).

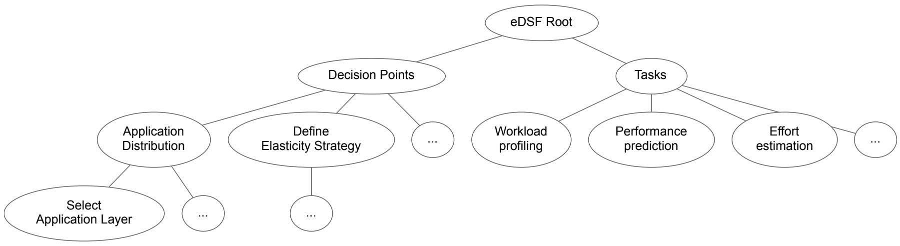


Figure 5.1.: Hierarchy of the Elaborated Decision Support Framework

The field of information visualization offers suitable representations for various kinds of data and according to [39], the following two sections introduce possible visualization solutions.

Visualization of Hierarchies and Trees

Hierarchically structured data visualizations are divided in *space-filling* and *non-space-filling* representations. Space-filling representations consider the maximal available space and display related element by using juxtapositioning. Figure 5.2 shows two common space-filling visualizations, a treemap display and a sunburst display.

Treemaps [91] map the hierarchy on a rectangular region that is recursively divided into slices based on the number of subtrees on each level. Sunbursts [92] are a radial space-filling visualization starting with a centered hierarchy root and appending a ring for each level, which is subdivided into arcs deeding on the subtrees.

In terms of non-space-filling representations a simple node-link relation is the most common representation for a tree or hierarchy diagram like also depicted in Figure 5.2 as the predecessor of the treemap and sunburst.

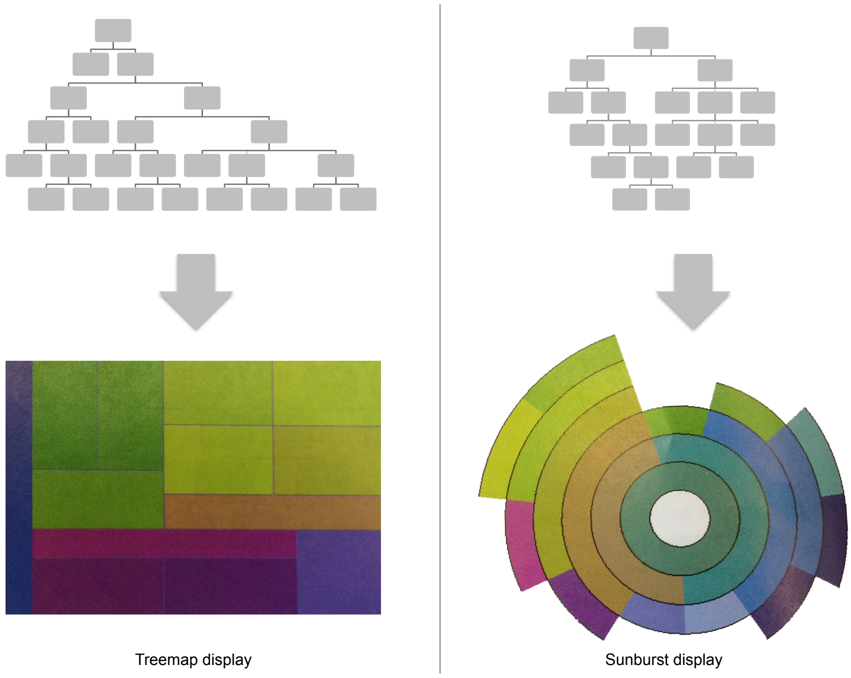


Figure 5.2.: Possible Hierarchy Visualizations [39]

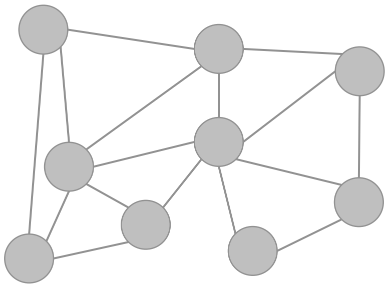
Visualization of Networks

Referring to [39], network graphs can be represented by using also a *node-link diagram* or a *matrix display*. Figure 5.3 shows examples of each representation kind. The former displays all elements as nodes and their relationship as links. A special kind of this representation is the force-directed network graph utilizing an algorithm to iteratively position nodes based on a spring analogy for the links. The latter, relies on an adjacency matrix with an n-by-n grid where n is the total number of nodes. A relationship (link) of node i and j is shown at the matrix position (i,j) whereas various kinds of values can be used to determine the relationship type.

5.2.2. Technologies and Frameworks

HTML5 and CSS

Both *HTML* (Hypertext Markup Language) and *CSS* (Cascading Style Sheets) represent two of the core technologies for constructing web pages [93]. HTML is the language



Node-link display

i/j	a	b	c	d	e	f	g	h
a		•	•			•		
b	•			•		•		
c	•				•		•	•
d	•	•				•		
e			•				•	•
f	•	•		•				
g			•		•			•
h			•		•		•	

Matrix display

Figure 5.3.: Possible Network Visualizations [39]

used to describes the structure of the web page by utilizing markup of elements like paragraphs, lists, tables, etc. and it is currently defined in its 5th major revision (HTML5) [94]. Besides, CSS is the language used to describe the presentation of a web page by defining with layout, colors, and fonts etc. of HTML elements [93]. Further, specified presentations for various device types are possible like for example small screens (i.e. mobile devices), large screens, or printers.

Scalable Vector Graphics

Scalable Vector Graphics (SVG) is comparable to HTML a markup language to describe two-dimensional vector graphics [95] [96]. Beyond defining just the geometry of shapes SVG also provides styling features like gradients, opacity, and filters. Furthermore, most modern browsers and mobile devices natively support SVG, which makes it useful in terms of using graphics within web pages.

JavaScript

JavaScript (JS) is a scripting language developed and defined by Ecma (with the name ECMAscript)[97], which is most commonly used in web browsers to dynamically manipulate web pages [98]. Such scripts are executed client-side in the users browser to shift functionality from the server to the client.

JSON

The *JavaScript Object Notation* (JSON) represents a programming language independent and lightweight text format for data interchange [99]. The format is defined as a subset of JavaScript in the Standard ECMA-262 [100]. The structure of a JSON object conforms to a collection of name/value pairs where a string specifies the name and the value can be a nested object, array, number, string, boolean value, or null. Due to its characteristics JSON is ideal to use in conjunction with web applications utilizing JavaSctipt-based frameworks.

jQuery

jQuery is a free JavaScript library under the terms of the MIT License[3] which provides simple and easy-to-use functions for HTML document access and manipulation, event handling, animation, and Ajax[4] [101]. The strength of jQuery are its fast, easy-to-use, and small-sized API with cross-browser support and compliance to latest web technologies (e.g. CSS3[5]), wherefore it is used by frameworks like Bootstrap d3.js, etc.

D3 - Data-Driven Documents

Data-Driven Documents (D3) is a free JavaScript library for data visualization making use of web standards like HTML, SVG, and CSS [102]. Based on visualization components (e.g. for chart, tree, or network representations) D3 allows an easy creation of interactive and dynamic representations of arbitrary data in modern browsers. D3 requires the jQuery library and is free to use under the terms of the BSD License[6].

Bootstrap

Bootstrap is a free front-end framework for creating state-of-the-are websites with meanwhile a special focus on responsive design and the mobile first approach [103]. The framework provides interface components and design templates build with HTML and CSS for navigation, buttons, forms, etc. with additional JavaScript support through

[3] Massachusetts Institute of Technology (MIT) License: http://opensource.org/licenses/MIT
[4] Asynchronous JavaScript + XML (Ajax): https://developer.mozilla.org/en-US/docs/AJAX
[5] Cascading Style Sheets 3: https://developer.mozilla.org/en-US/docs/Web/CSS/CSS3
[6] Berkeley Source Distribution (BSD) License:http://opensource.org/licenses/BSD-3-Clause

using the jQuery library. Bootstrap was created within the context of Twitter, open-sources in 2011 under the terms of the MIT License[3], and has received large popularity through its usage in projects like from NASA[7] or MSNBC[8] [104] [105].

5.2.3. System Architecture

The prototypical implementation developed is a single-page web application utilizing the previous mentioned technologies and frameworks. In Figure 5.4 the systems architecture is depicted showing on the one hand the logical architecture in terms of application layers with technologies and frameworks used for each layer and on the other the physical systems architecture with reference to application tiers.

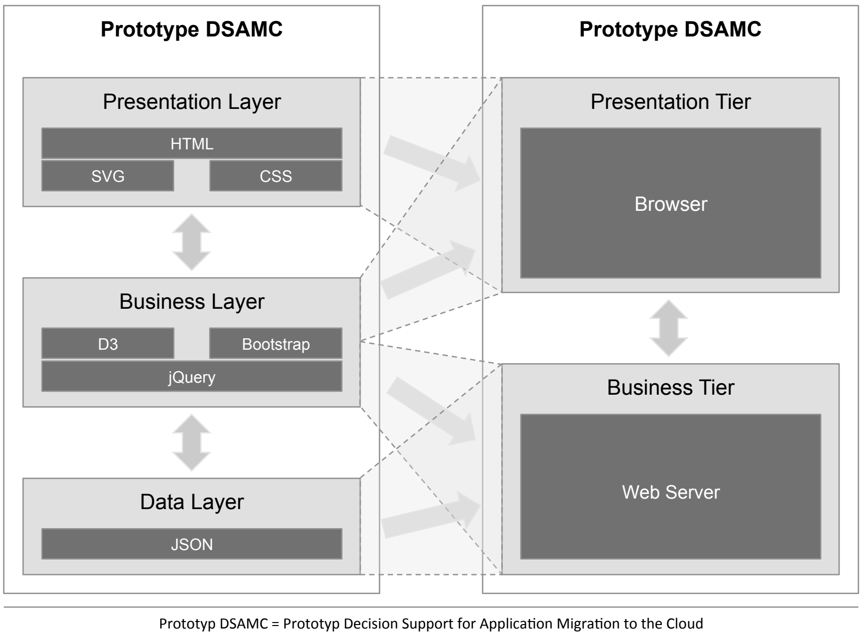


Figure 5.4.: System Architecture of the Prototypical Implementation

On the prototype's Presentation Layer HTML, SVG, and CSS are used for visualization and user interface of the web page. HTML describes the general structure with several elements, SVG is used for creating graphics which then visualizes the element of the eDSF, and CSS is used to style all various HTML and SVG element in an appropriate manner.

7 code.NASA: http://code.nasa.gov/project/
8 Breaking News: http://www.breakingnews.com

The Business Layer handles the logic for realizing the visualization scenarios and also the users' interactions with them. For data visualization of the eDSF the D3 library and its visualization components like force network, partition diagrams, tree diagrams, etc. are used. Most of these components utilize SVG elements (some also HTML elements) and append them to a selected element in the HTML document via JavaScript (for this purpose jQuery is required by D3). User interface components like dropdown lists or other form controls come from the Bootstrap framework, which provides those components with ready-to-use JavaScript (here also jQuery is required) functionality as well as browser independent layout styles to achieve easy cross-browser compatibility. The business logic for handling user interactions (e.g. changing the visualization from a tree to a partition layout or selecting just decision or task to be visualized) is implemented by using jQuery. Finally, on the Data Layer, the actual eDSF data (i.e. decision, tasks, relationships, etc.) is organized by using the JSON format.

With respect to the physical architecture the prototype consists of two tiers: the business tier represented by a web server storing the files and delivering them by request to the web browser on the client tier. As the prototype is a single-page web application the main file where all HTML and SVG elements as well as references to CSS and JavaScript files are defined is the *index.html*. General layout styles and styles specific for the eDSF visualization are defined separately in the two files *dsamc-layout.css* and *dsamc-svg.css*. The main business logic is implemented in the JavaScript file dsamc.js based on using the jQuery, D3, and Bootstrap JavaScript libraries. The actual eDSF data is stored in a single JSON file (*elaboratedDSF.json*).

5.3. Prototypical Implementation

This section provides insights into the prototypical implementation of a decision support system based on the elaborated Decision Support Framework information, which is required in *RO. 4* (see Section 1.1). The prototype takes into account requirements stated in Section 5.1, advised representations for data visualization pointed out in Section 5.2.1, and was developed considering the system architecture depicted in Figure 5.4 based on technology and frameworks introduced in Section 5.2.2. The prototype implements five types of visualizations grouped in either network or hierarchy layouts:

Network Layouts

- Network Layout
- Cluster Layout

Hierarchy Layouts

- Tree Layout
- Treemap Layout
- Partition Layout

As this is one smaller objective required in terms of this Master's thesis the following sections introduce the prototype and the different visualization types of eDSF data with the aid of screenshots. The actual implementation of this prototype is online available at http://www.cloudDSF.com.

5.3.1. User Interface

The prototype is implemented as a web-based application fulfilling non-functional requirements listed in Table 5.2 regarding general usability qualities. NFR-1 requires an easy-to-use common interface for the prototype and NFR-3 requires the user interface to be well structured, accessible intuitively, and focused on visualizing the eDSF. Figure 5.5 depicts the user interface of the prototype with its main areas highlighted.

Both Header and Footer area do not contain necessary elements for the main objective of visualizing the eDSF and therefore they are limited to general information like heading and author name. The horizontal navigation and label area first provides a drop-down list where the user can select between the visualization types (e.g. force layout, tree layout, treemap layout, etc.), which are grouped by either network or hierarchy related representational layouts. The label at the side dynamically changes based on the displayed eDSF information. Additionally, the bottom sidebar area (i.e. sidebar information) provides further information like element kind, its relationships, etc. when the user selects an element by clicking on it. The upper sidebar area (Settings) comprises additional preference attributes where the user is for example able to select only a subset of eDSF elements or possibly disables element labels in the visualization. To focus on information visualization of the eDSF this latter content area can be collapsed and expanded either automatically (e.g. when a new visualization type is loaded) or on demand by the user by clicking on the grey heading. Finally, the majority of the screen is reserved for the actual visualization of eDSF information in terms of several implemented visualization types.

Most of the CSS layout is based on procedural attributes (e.g. elements widths, margins, etc.) to individually scale the prototype on various devices but keep general proportions

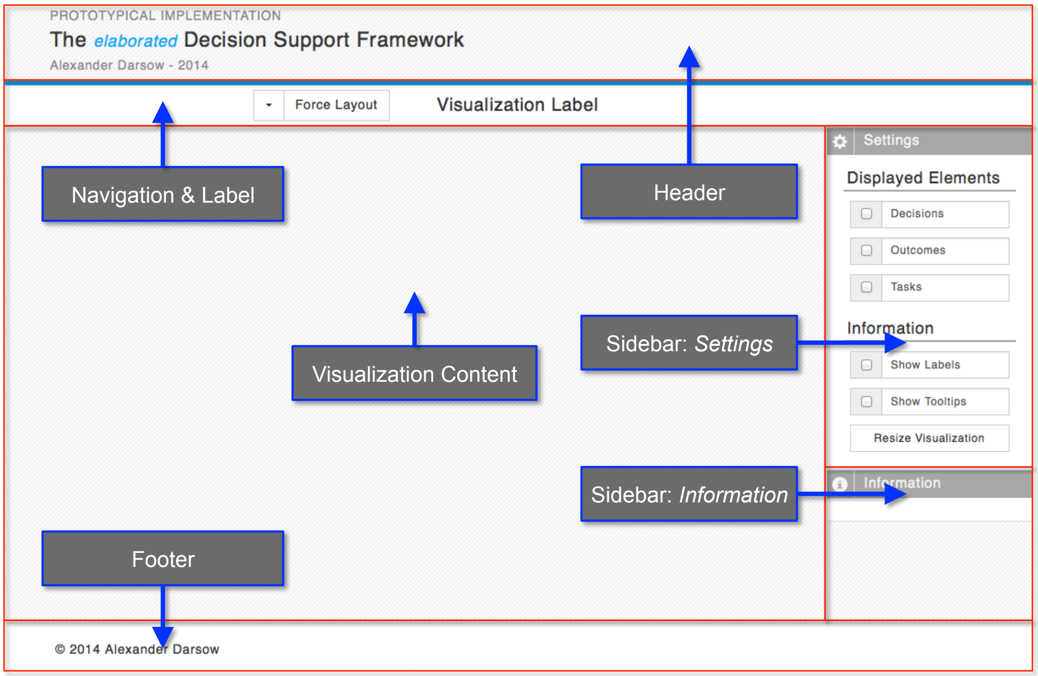


Figure 5.5.: User Interface of the Prototypical Implementation

and to maximize the visualization content area for example on large screens. Besides, an appropriate visualization of the eDSF and its relationships arguably struggles on small screen devices hence a minimal width of 1100 pixel is defined and an advice regarding this limitation is displayed to users with small screen devices.

5.3.2. Data Visualization Examples

Hierarchy Visualization Example

The prototype implements three visualizations for representing hierarchical structures of the eDSF, namely *tree layout*, *treemap layout*, and *partition layout*. By this, both approaches for hierarchical data visualizations mentioned in Section 5.2.1 (i.e. space-filling and non-space-filling) are addressed by the prototypical implementation, as the tree layout is a non-space-filling visualization and in turn the treemap and partition layout are examples for space-filling representations.

The following screenshots all visualize the hierarchical structure of the eDSF decisions i.e. four decision points with their subordinated and evaluated decisions, which again have their defined outcomes. Figure 5.6 shows the tree visualization, Figure 5.7 depicts the treemap representations, and Figure 5.8 illustrates the partition layout.

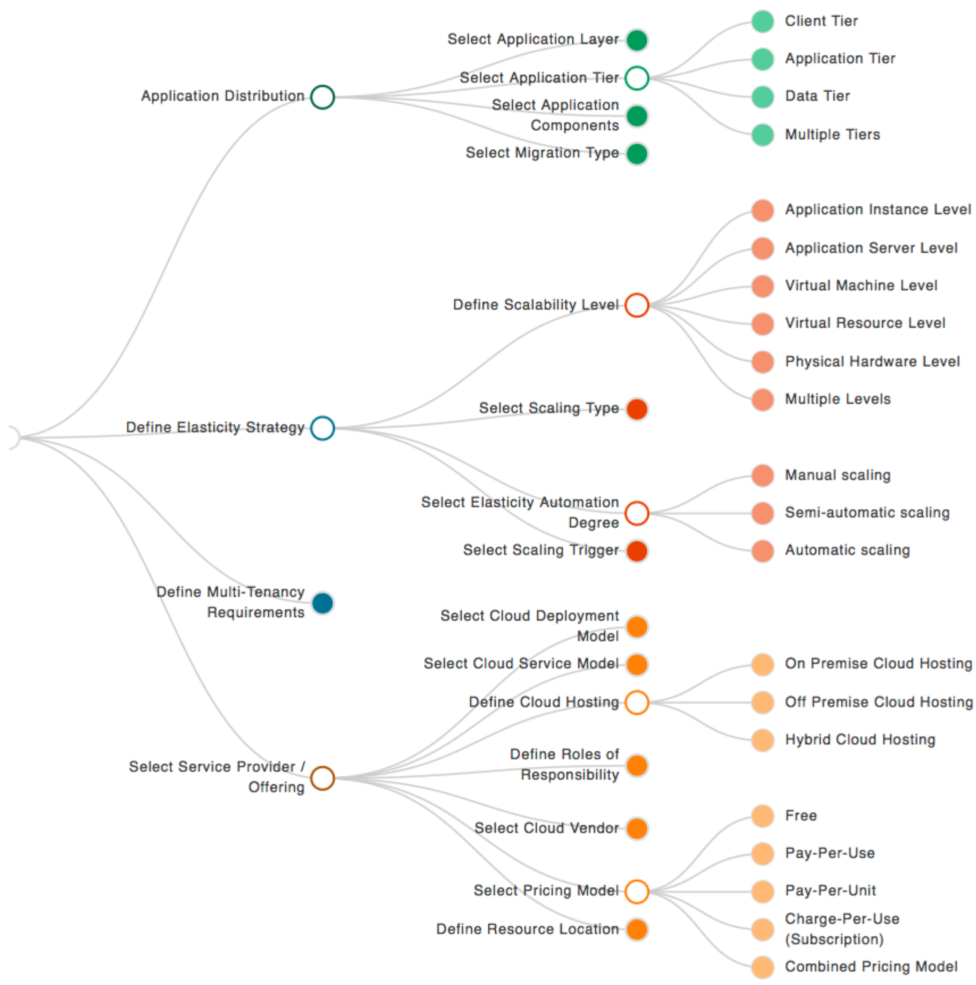


Figure 5.6.: Prototypical Implementation - Tree Layout

Network Visualization Example

Regarding the eDSF network structure the prototype implements two visualizations, named *force layout* and *cluster layout*. According to Section 5.2.1, both show a node-link-diagram in general but with different representation created by utilizing D3 template layouts.

Figure 5.7.: Prototypical Implementation - Treemap Layout

The force layout depicted in Figure 5.9 uses the force-directed graph layout implementation of D3 with some limitations made in terms of node positioning. Due to the number of eDSF elements and characteristics of the D3 layout (e.g. node individual property settings like minimum distance between nodes are not supported, definition of node sizes, etc.) the basic D3 force layout produced a very bad and confusing representation. Hence, elements link decision points have been fixed positioned, for others like decisions and task certain boundaries have been defined to impose some basic element organization and minimize element overlays.

The cluster layout in Figure 5.10 is another representation depicting the eDSF network structure but without the hierarchical connections between decision points and their

Figure 5.8.: Prototypical Implementation - Partition Layout

subordinated decisions. Two D3 layouts have been combined to establish the circular representation of decision groups and tasks and their chord connections representing either influencing or affecting relations.

Figure 5.9.: Prototypical Implementation - Force Layout

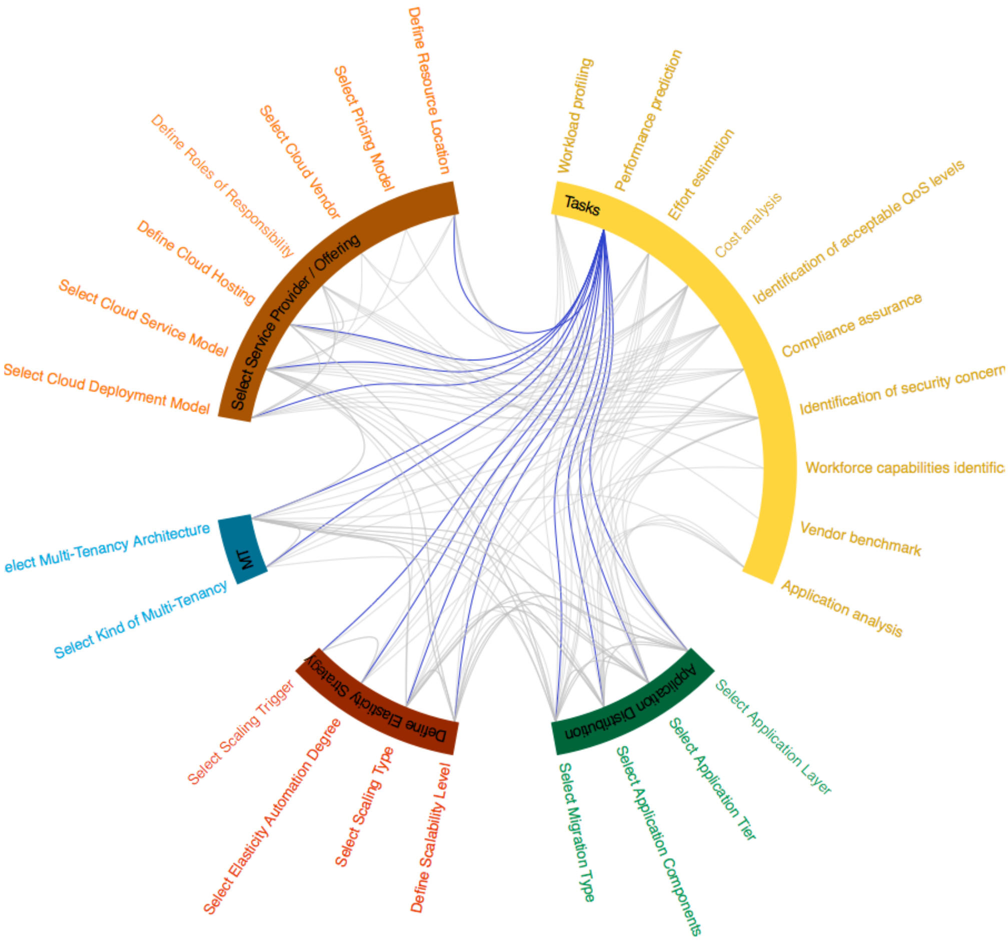


Figure 5.10.: Prototypical Implementation - Cluster Layout

6. Conclusion and Further Research

This Master's thesis refers to the elaboration of the conceptual Decision Support Framework developed by Andrikopoulos et al. [51] supporting decision-making in terms of application migration to the Cloud. The elaboration focus was set on the four decision points, their possible outcomes, and their relationships within the eDSF (i.e. relations to tasks and to other decisions).

6.1. Research Objectives

In this section, the research objectives stated in Section 1.1 are answered and notable findings are discussed.

RO. 1 *Organize the State of the Art on application migration support and Decision Support Systems in the context of Cloud Computing*

Chapter 2 addresses this research objective, since Section 2.2 introduces the field of application migration to the Cloud based on current literature and depicts migration solutions in terms of specific Cloud migration types. As the consideration of migrating existing applications to the Cloud receives increasing attention by enterprises, this field is meanwhile addressed by several research approaches. Section 2.3.3 presents promising current research approaches regarding decision support for the issue of application migration to the Cloud, which are also considered in a more comprehensive current review on Cloud migration research [5]. Based on this, the State of the Art on application migration support to the Cloud is organized.

Chapter 2 provides insights into decision dupport (Section 2.3.1) and decision support systems (Section 2.3.2). Different types of decision support systems (DSS) are depicted, whereas the DSF would enable a *knowledge-driven DSS* based on its data (e.g. decisions, tasks, relationships, etc.) representing the knowledge based. Furthermore, the architecture of an DSS is discussed with respect to the prototypical implementation.

RO. 2 *Elaborate, refine, and evaluate the conceptual Decision Support Framework*

The elaboration and refinement of the conceptual Decision Support Framework (cDSF) is the main objective of this Master's thesis and is addressed in Chapter 3. Chapter 4 provides an evaluation of the elaborated version of the framework (eDSF). Within the elaboration a total of 17 decisions with possible outcomes have been identified, three additional tasks have been defined, and relationships between both decisions within the eDSF as well as between decisions and tasks have been presented. An overview of the elaborated Decision Support Framework is given in Section 3.4. Chapter 4 considers the evaluation of the eDSF based on a web-based survey addressed to participants in research as well as practitioners. Despite the low response rate findings show the general decision support approach of the cDSF (i.e. the distinction of decision points and tasks) is understood, the elaborated decisions within this thesis are understood and relevant in their respective decision point, and all decision points are relevant to be considered in terms of application migration to the Cloud.

RO. 3 *Identify requirements for an implementation of the Decision Support Framework*

Requirements for the prototypical implementation mainly concern the visualization of the eDSF data since its elements and their relations create a network structure. Visualization approaches from the field of information visualization are appropriate regarding the eDSF data structures and are introduced in Section 5.2.1. Additionally, Section 5.1 gives a short list of functional and non-functional requirements for the desired web-based prototype and Section 2.3.2 has depicted architectural components of an decision support system in terms of information systems, which can also be referred as requirements for the implementation.

RO. 4 *Provide a prototypical implementation of the Decision Support Framework*

The prototype implemented based on the above mentioned requirements is depicted in Section 5.3 through several screenshots, which illustrate the different visual representations of the eDSF data. The web-based prototype utilizes State of the Art technologies and aids decision makers as a decision support system in terms of application migration to the Cloud.

6.2. Further Research

Based on this Master's thesis further research aspects regarding the discussed decision support approach of Andrikopoulos et al. can be noticed. Since this work focused on the elaboration and refinement of decision points in future work also tasks can arguably be refined to specify their activities and to point out which results they provide to the different decisions.

Furthermore, the process of this work showed that selections made in certain decisions may constrain selections of other decision, e.g. in case migration type III is selected a further selection of application layers or tiers is obsolete since the whole application is desired to be migrated. These constraints possibly effect decisions in the whole eDSF and they can be investigated by future work.

With regard to the low response rate on the eDSF evaluation in this work, a more comprehensive evaluation (e.g. after further elaborations of the framework) could be considered. Because the topic of application migration to the Cloud shows relevance in both research and enterprises, findings regarding the special demands of enterprise decision-makers or possible variety in prioritization of aspects (e.g. certain decision points, decisions, tasks, etc.) between both domains could lead further research topics.

In order to transform the eDSF into an actual decision support system future work and effort has to be spent. The prototype developed in this thesis is a first step towards this objective as it provides visual impressions on how the eDSF elements are related to each other. But referring to the generic architecture of a DSS depicted in Figure 2.3, the data and information of the eDSF, for example, have to be moved to a knowledge base component. Furthermore, the remaining components (problem processing system, presentation system, etc.) have to be defined and implemented to reach an actual decision support system for Cloud migration.

Bibliography

[1] Gartner, Inc.: Gartner says the road to increased enterprise cloud usage will largely run through tactical business solutions addressing specific issues. http://www.gartner.com/newsroom/id/2581315 (August 2013)

[2] Gartner, Inc.: Gartner says worldwide public cloud services market to total $131 billion. http://www.gartner.com/newsroom/id/2352816 (February 2013)

[3] Buyya, R., Broberg, J., Gosćinśki, A.: Cloud computing: principles and paradigms. Wiley, Hoboken, N.J. (2011)

[4] Golden, B.: The case against cloud computing, part one. http://www.cio.com/article/477473/The_Case_Against_Cloud_Computing_Part_One (January 2009)

[5] Jamshidi, P., Ahmad, A., Pahl, C.: Cloud migration research: A systematic review. IEEE Transactions on Cloud Computing (2013) 1–1

[6] Beserra, P.V., Camara, A., Ximenes, R., Albuquerque, A.B., Mendonca, N.C.: Cloudstep: A step-by-step decision process to support legacy application migration to the cloud. In: Maintenance and Evolution of Service-Oriented and Cloud-Based Systems (MESOCA), 2012 IEEE 6th International Workshop on the. (2012) 7–16

[7] Tran, V., Keung, J., Liu, A., Fekete, A.: Application migration to cloud: a taxonomy of critical factors. In: Proceedings of the 2nd International Workshop on Software Engineering for Cloud Computing. (2011) 22–28

[8] Badger, L., Grance, T., Patt-Corner, R., Voas, J.: Cloud computing synopsis and recommendations. http://www.nist.gov/customcf/get_pdf.cfm?pub_id=911075 (2012)

[9] IBM Deutschland GmbH: Cloud-plattform (iaas, paas). http://www.ibm.com/cloud-computing/de/de/iaas.html (January 2014)

[10] Hewlett-Packard Development Company, L.P.: Converged cloud services. http://www8.hp.com/us/en/business-services/it-services.html?compURI=1077445#.UukJaHkp8zQ (January 2014)

[11] Menzel, M., Ranjan, R.: Cloudgenius: decision support for web server cloud migration. In: Proceedings of the 21st international conference on World Wide Web. (2012) 979–988

[12] Hajjat, M., Sun, X., Sung, Y.W.E., Maltz, D., Rao, S., Sripanidkulchai, K., Tawarmalani, M.: Cloudward bound: planning for beneficial migration of enterprise applications to the cloud. ACM SIGCOMM Computer Communication Review **40**(4) (2010) 243–254

[13] Khajeh-Hosseini, A., Greenwood, D., Smith, J.W., Sommerville, I.: The cloud adoption toolkit: supporting cloud adoption decisions in the enterprise. Software: Practice and Experience **42**(4) (April 2012) 447–465

[14] Khajeh-Hosseini, A., Greenwood, D., Smith, J.W., Sommerville, I.: The cloud adoption toolkit: Supporting cloud adoption decisions in the enterprise. arXiv e-print 1008.1900 (August 2010)

[15] Andrikopoulos, V., Binz, T., Leymann, F., Strauch, S.: How to adapt applications for the cloud environment: Challenges and solutions in migrating applications to the cloud. Computing **95**(6) (June 2013) 493–535

[16] Andrikopoulos, V., Strauch, S., Leymann, F.: 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, 8-10 May 2013, Aachen, Germany, SciTePress (2013) 149–155

[17] Vaquero, L.M., Rodero-Merino, L., Caceres, J., Lindner, M.: A break in the clouds: towards a cloud definition. ACM SIGCOMM Computer Communication Review **39**(1) (2008) 50–55

[18] Armbrust, M., Fox, A., Griffith, R., Joseph, A.D., Katz, R.H., Konwinski, A., Lee, G., Patterson, D.A., Rabkin, A., Stoica, I., Zaharia, M.: Above the clouds: A berkeley view of cloud computing. Technical Report UCB/EECS-2009-28, EECS Department, University of California, Berkeley (February 2009)

[19] Parkhill, D.F.: The Challenge of the Computer Utility. Addison-Wesley Publishing Company (1966)

[20] Buyya, R., Yeo, C.S., Venugopal, S., Broberg, J., Brandic, I.: Cloud computing and emerging it platforms: Vision, hype, and reality for delivering computing as the 5th utility. Future Generation Computer Systems **25**(6) (June 2009) 599–616

[21] SYS-CON Media Inc.: Twenty-one experts define cloud computing. http://cloudcomputing.sys-con.com/node/612375/print (January 2009)

[22] Mell, P., Grance, T.: The nist definition of cloud computing. http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf (2011)

[23] Sriram, I., Khajeh-Hosseini, A.: Research agenda in cloud technologies. arXiv preprint arXiv:1001.3259 (January 2010)

[24] Bundesamt für Sicherheit in der Informationstechnik (BSI): Cloud computing grundlagen. https://www.bsi.bund.de/DE/Themen/CloudComputing/Grundlagen/Grundlagen_node.html;jsessionid=4262AD93020BA04DC983BDD60C8F06FD.2_cid359 (2013)

[25] Garcia, R.C., Chung, J.M.: Xaas for xaas: An evolving abstraction of web services for the entrepreneur, developer, and consumer. In: Circuits and Systems (MWSCAS), 2012 IEEE 55th International Midwest Symposium on, IEEE (August 2012) 853–855

[26] Lenk, A., Klems, M., Nimis, J., Tai, S., Sandholm, T.: What's inside the cloud? an architectural map of the cloud landscape. In: CLOUD '09 Proceedings of the 2009 ICSE Workshop on Software Engineering Challenges of Cloud Computing, IEEE (2009) 23–31

[27] Settu, R., Raj, P.: Cloud application modernization and migration methodology. In Mahmood, Z., ed.: Cloud Computing. Springer London, London (2013) 243–271

[28] Khajeh-Hosseini, A., Sommerville, I., Sriram, I.: Research challenges for enterprise cloud computing. arXiv preprint arXiv:1001.3257 (January 2010)

[29] Institute of Electrical and Electronics Engineers, International Organization for Standardization, International Electrotechnical Commission: Software engineering - Software Life Cycle Processes - Maintenance. Institute of Electrical and Electronics Engineers, New York, NY (2006)

[30] Gimnich, R., Winter, A.: Workflows der software-migration. Softwaretechnik-Trends (2005) 22–24

[31] Razavian, M., Lago, P.: A frame of reference for soa migration. In Nitto, E.D., Yahyapour, R., eds.: Towards a Service-Based Internet. Number 6481 in Lecture Notes in Computer Science. Springer Berlin Heidelberg (January 2010) 150–162

[32] Vu, Q.H., Asal, R.: Legacy application migration to the cloud: Practicability and methodology. In: Services (SERVICES), 2012 IEEE Eighth World Congress on, IEEE (June 2012) 270–277

[33] Cloud Computing Use Cases group: Moving to the cloud. http://cloudusecases.org/Moving_to_the_Cloud.pdf (February 2011)

[34] Babar, M.A., Chauhan, M.A.: A tale of migration to cloud computing for sharing experiences and observations. In: Proceedings of the 2nd International Workshop on Software Engineering for Cloud Computing. (2011) 50–56

[35] Khajeh-Hosseini, A., Greenwood, D., Sommerville, I.: Cloud migration: A case study of migrating an enterprise it system to iaas, IEEE (July 2010) 450–457

[36] Burstein, F.: Handbook on decision support systems 1: the basics. 1st ed edn. Number 3795 in International handbooks on information systems. Springer, New York (2008)

[37] Hwang, C., Paidy, S., Yoon, K., Masud, A.: Mathematical programming with multiple objectives: A tutorial. Computers & Operations Research **7**(1–2) (1980) 5–31

[38] Figueira, J., Mousseau, V., Roy, B.: Electre methods. In: Multiple Criteria Decision Analysis: State of the Art Surveys. Volume 78. Springer-Verlag, New York (2005) 133–153

[39] Ward, C., Aravamudan, N., Bhattacharya, K., Cheng, K., Filepp, R., Kearney, R., Peterson, B., Shwartz, L., Young, C.: Workload migration into clouds challenges, experiences, opportunities, IEEE (July 2010) 164–171

[40] Saaty, T.L.: How to make a decision: The analytic hierarchy process. European Journal of Operational Research **48**(1) (September 1990) 9–26

[41] L Saaty, T.: The analytic network process. Iranian Journal of Operations Research **1**(1) (2008) 1–27

[42] Saaty, T.L., Vargas, L.G.: Decision making with the analytic network process eco-

nomic, political, social and technological applications with benefits, opportunities, costs and risks. Springer, New York (2006)

[43] Saaty, T.L.: Fundamentals of the analytic network process. In: ISAHP 1999, Kobe (Japan) (1999)

[44] Figueira, J.R., Greco, S., Roy, B., Słowiński, R.: An overview of electre methods and their recent extensions: Electre methods. Journal of Multi-Criteria Decision Analysis **20**(1-2) (January 2013) 61–85

[45] Power, D.J.: Decision Support Systems: Concepts and Resources for Managers. Greenwood Publishing Group (January 2002)

[46] Keen, P.G.W.: Decision support systems: an organizational perspective. Addison-Wesley series on decision support. Addison-Wesley Pub. Co, Reading, Mass (1978)

[47] Holsapple, C.: Decisions and knowledge. In: Handbook on Decision Support Systems 1. Springer Berlin Heidelberg, Berlin, Heidelberg 21–53

[48] Holsapple, C.: Dss architecture and types. In: Handbook on Decision Support Systems 1. Springer Berlin Heidelberg, Berlin, Heidelberg 163–189

[49] Sprague, R.H., Carlson, E.D.: Building effective decision support systems. Prentice-Hall (1982)

[50] Fowler, M., Rice, D.: Patterns of enterprise application architecture. Addison-Wesley, Boston, MA (2003)

[51] Andrikopoulos, V., Song, Z., Leymann, F.: Supporting the migration of applications to the cloud through a. In: Proceedings of the 6th IEEE International Conference on Cloud. (2013) 565–572

[52] Menzel, M., Schönherr, M., Nimis, J., Tai, S.: (mc2) 2: A generic decision-making framework and its application to cloud computing. arXiv preprint arXiv:1112.1851 (2011)

[53] Chauhan, M.A., Babar, M.A.: Towards process support for migrating applications to cloud computing, IEEE (November 2012) 80–87

[54] Suleiman, B., Sakr, S., Jeffery, R., Liu, A.: On understanding the economics and elasticity challenges of deploying business applications on public cloud infrastructure. Journal of Internet Services and Applications **3**(2) (December 2011) 173–193

[55] Brebner, P.C.: Is your cloud elastic enough?: performance modelling the elasticity of infrastructure as a service (iaas) cloud applications, ACM Press (2012) 263

[56] Reese, G.: Cloud application architectures: building applications and infrastructure in the cloud. O'Reilly Media, Sebastopol, CA (2009)

[57] Sneed, H.M., Hasitschka, M., Teichmann, M.T.: Software-Migration. Dpunkt, Heidelberg, Neckar (2008)

[58] Amazon Web Services, Inc.: Aws | amazon rds - cloud relational database service. https://aws.amazon.com/rds/details/

[59] Strauch, S., Breitenbuecher, U., Kopp, O., Leymann, F., Unger, T.: Cloud data patterns for confidentiality. In: Proceedings of the 2nd International Conference on Cloud Computing and Service Science, CLOSER 2012, 18-21 April 2012, Porto, Portugal, SciTePress (2012) 387–394

[60] Strauch, S., Andrikopoulos, V., Breitenbuecher, U., Kopp, O., Leymann, F.: Non-functional data layer patterns for cloud applications. In: Proceedings of the 4th IEEE International Conference on Cloud Computing Technology and Science, CloudCom 2012, 3-6 December 2012, Taipei, Taiwan, IEEE Computer Society (2012) 601–605

[61] Tak, B.C., Urgaonkar, B., Sivasubramaniam, A.: To move or not to move: The economics of cloud computing. In: Proceedings of the 3rd USENIX Conference on Hot Topics in Cloud Computing, Portland, OR, USENIX Association (2011) 5–5

[62] Trowbridge, D., Mancini, D., Quick, D., Hohpe, G., Newkirk, J., Lavigne, D.: Tiered Distribution. http://msdn.microsoft.com/en-us/library/ff647195.aspx (June 2003) Microsoft Developer Network Library.

[63] Trowbridge, D., Mancini, D., Quick, D., Hohpe, G., Newkirk, J., Lavigne, D.: Three-Tiered Distribution. http://msdn.microsoft.com/en-us/library/ff647546.aspx (June 2003) Microsoft Developer Network Library.

[64] Herbst, N.R., Kounev, S., Reussner, R.: Elasticity in cloud computing: What it is, and what it is not. In: Presented as part of the 10th International Conference on Autonomic Computing, San Jose, CA, USENIX (2013) 23–27

[65] Vaquero, L.M., Rodero-Merino, L., Buyya, R.: Dynamically scaling applications in the cloud. ACM SIGCOMM Computer Communication Review **41**(1) (January 2011) 45

[66] Ardagna, C.A., Damiani, E., Frati, F., Rebeccani, D., Ughetti, M.: Scalability patterns for platform-as-a-service, IEEE (June 2012) 718–725

[67] Pors, M.: A Decision Support Model for Multi-Tenant Architectures. Master's thesis, Utrecht University, Utrecht (July 2013)

[68] Chauhan, M.A., Babar, M.A.: Migrating service-oriented system to cloud computing: An experience report, IEEE (July 2011) 404–411

[69] Amazon Web Services: Auto scaling. http://aws.amazon.com/en/autoscaling/ (2003)

[70] Sanjay Sohoni: Easily scale your cloud with rackspace auto scale. http://www.rackspace.com/blog/easily-scale-your-cloud-with-rackspace-auto-scale/ (November 2013)

[71] Sallam, A., Li, K.: Virtual machine proactive scaling in cloud systems. In: Cluster Computing Workshops (CLUSTER WORKSHOPS), 2012 IEEE International Conference on, IEEE (September 2012) 97–105

[72] Li, A., Yang, X., Kandula, S., Zhang, M.: Cloudcmp: comparing public cloud providers. In: Proceedings of the 10th ACM SIGCOMM conference on Internet measurement. (2010) 1–14

[73] Guo, C.J., Sun, W., Huang, Y., Wang, Z.H., Gao, B.: A framework for native multi-tenancy application development and management. In: E-Commerce Technology and the 4th IEEE International Conference on Enterprise Computing, E-Commerce, and E-Services, 2007. CEC/EEE 2007. The 9th IEEE International Conference on. (2007) 551–558

[74] Bezemer, C.P., Zaidman, A.: Multi-tenant saas applications: maintenance dream or nightmare?, ACM Press (2010) 88

[75] Chong, F., Carraro, G., Wolter, R.: Multi-Tenant Data Architecture. http://msdn.microsoft.com/en-us/library/Aa479086 (June 2006) Microsoft Developer Network Library.

[76] Rackspace, US Inc.: Rackspace: The leader in hybrid cloud. http://www.rackspace.com/ (2014)

[77] Amazon Web Services, Inc: Amazon web services, cloud computing: Compute, storage, database. http://aws.amazon.com/ (2013)

[78] Gagliardi, F., Muscella, S.: Cloud computing –data confidentiality and interoperability challenges. In Antonopoulos, N., Gillam, L., eds.: Cloud Computing. Springer London, London (2010) 257–270

[79] Brebner, P., Liu, A.: Performance and cost assessment of cloud services. In: Service-Oriented Computing. Springer (2011) 39–50

[80] Khajeh-Hosseini, A., Sommerville, I., Bogaerts, J., Teregowda, P.: Decision support tools for cloud migration in the enterprise, IEEE (July 2011) 541–548

[81] Gudenkauf, S., Josefiok, M., Goring, A., Norkus, O.: A reference architecture for cloud service offers. In: Enterprise Distributed Object Computing Conference (EDOC), 2013 17th IEEE International. (2013) 227–236

[82] Saripalli, P., Pingali, G.: Madmac: Multiple attribute decision methodology for adoption of clouds, IEEE (July 2011) 316–323

[83] Info-Tech Research Group: Cloud vendor shortlisting tool | info-tech research group. http://www.infotech.com/research/cloud-vendor-shortlisting-tool (June 2011)

[84] Armbrust, M., Fox, A., Griffith, R., Joseph, A.D., Katz, R., Konwinski, A., Lee, G., Patterson, D., Rabkin, A., Stoica, I.: A view of cloud computing. Communications of the ACM **53**(4) (2010) 50–58

[85] Amazon Web Services, Inc.: Amazon ec2 pricing, pay as you go for cloud computing services. http://aws.amazon.com/en/ec2/pricing/ (2013)

[86] Rackspace, US Inc.: Pricing for dedicated servers by rackspace. http://www.rackspace.com/managed-hosting/dedicated-servers/pricing/ (2014)

[87] Chaput, S.R., Ringwood, K.: Cloud compliance: A framework for using cloud computing in a regulated world. In Antonopoulos, N., Gillam, L., eds.: Cloud Computing. Springer London, London (2010) 241–255

[88] Marsden, P.V., Wright, J.D.: Handbook of survey research. Emerald, Bingley, UK (2010)

[89] Krosnick, J.A., Presser, S.: Question and questionnaire design. In: Handbook of survey research. Emerald, Bingley, UK (2010)

[90] Microsoft Developer Network: Microsoft Application Architecture

Guide. 2nd edn. Patterns & practices. Microsoft, Redmond (2009) http://msdn.microsoft.com/en-us/library/ee658117.aspx.

[91] Johnson, B., Shneiderman, B.: Tree-maps: a space-filling approach to the visualization of hierarchical information structures, IEEE Comput. Soc. Press (1991) 284–291

[92] Stasko, J., Zhang, E.: Focus+context display and navigation techniques for enhancing radial, space-filling hierarchy visualizations, IEEE Comput. Soc (2000) 57–65

[93] W3C: HTML & CSS. http://www.w3.org/standards/webdesign/htmlcss (2013)

[94] W3C: HTML5. http://www.w3.org/TR/html5/ (2013)

[95] W3C: Scalable Vector Graphics (SVG) 1.1 (Second Edition). http://www.w3.org/TR/SVG/ (2013)

[96] W3C: SVG Working Group. http://www.w3.org/Graphics/SVG/ (2013)

[97] Ecma International: Ecma-262 - ecmascript language specification. http://www.ecma-international.org/ecma-262/5.1/ (June 2011)

[98] W3C: JavaScript Web APIs. http://www.w3.org/standards/webdesign/script.html (2013)

[99] www.json.org: Introducing json. http://json.org/index.html (2014)

[100] Ecma International: Ecma-404 - the json data interchange format (October 2013)

[101] The jQuery Foundation: jquery. http://jquery.com/ (2014)

[102] Bostock, M.: D3 - data-driven documents. http://d3js.org/ (2013)

[103] Twitter, Inc: Bootstrap. http://getbootstrap.com/ (2014)

[104] Mark Otto: Bootstrap from twitter. https://blog.twitter.com/2011/bootstrap-twitter (August 2011)

[105] Mark Otto: Say hello to bootstrap 2.0. https://blog.twitter.com/2012/say-hello-to-bootstrap-2 (January 2012)

All links followed on March 4, 2014.

A. Appendix

A.1. Relationships between Decisions

This section of the appendix provides the tabular representation of the relationships between decision within the elaborated Decision Support Framework (eDSF) discussed in Section 3.2.5.

	Decision Point 1 - Application Distribution	Decision 1.1 - Select Application Layer	Decision 1.2 - Select Application Tier	Decision 1.3 - Select Application Components	Decision 1.4 - Select Migration Type	Decision Point 2 - Define Elasticity Strategy	Decision 2.1 - Define Scalability Level	Decision 2.2 - Select Scaling Type	Decision 2.3 - Select Elasticity Automation Degree	Decision 2.4 - Select Scaling Trigger	Decision Point 3 - Define Multi-Tenancy Requirements	Decision 3.1 - Select Kind of Multi-tenancy	Decision 3.2 - Select Multi-tenancy Architecture	Decision Point 4 - Select Service Provider / Offering	Decision 4.1 - Select Cloud Deployment Model	Decision 4.2 - Select Cloud Service Model	Decision 4.3 - Define Cloud Hosting	Decision 4.4 - Define Roles of Responsibility	Decision 4.5 - Select Cloud Vendor	Decision 4.6 - Select Pricing Model	Decision 4.7 - Define Resource Location
Decision Point 1 - Application Distribution																					
Decision 1.1 - Select Application Layer			≻	≻	≻		≻	≻					≻			≻					
Decision 1.2 - Select Application Tier		≻		≻	≻		≻	≻					≻			≻					
Decision 1.3 - Select Application Components		⊢	⊢		≻		≻	≻					≻			≻					
Decision 1.4 - Select Migration Type		≻	≻	≻			≻	≻				≻	≻			≻	≻				

≻ : influences ⊢ : determines

Table A.1.: Influences of Decision Point 1

	Decision Point 1 - Application Distribution	Decision 1.1 - Select Application Layer	Decision 1.2 - Select Application Tier	Decision 1.3 - Select Application Components	Decision 1.4 - Select Migration Type	Decision Point 2 - Define Elasticity Strategy	Decision 2.1 - Define Scalability Level	Decision 2.2 - Select Scaling Type	Decision 2.3 - Select Elasticity Automation Degree	Decision 2.4 - Select Scaling Trigger	Decision Point 3 - Define Multi-Tenancy Requirements	Decision 3.1 - Select Kind of Multi-tenancy	Decision 3.2 - Select Multi-tenancy Architecture	Decision Point 4 - Select Service Provider / Offering	Decision 4.1 - Select Cloud Deployment Model	Decision 4.2 - Select Cloud Service Model	Decision 4.3 - Define Cloud Hosting	Decision 4.4 - Define Roles of Responsibility	Decision 4.5 - Select Cloud Vendor	Decision 4.6 - Select Pricing Model	Decision 4.7 - Define Resource Location
Decision Point 2 - Define Elasticity Strategy																					
Decision 2.1 - Define Scalability Level		$\succ$	$\succ$	$\succ$				$\succ$					$\succ$			$\succ$					
Decision 2.2 - Select Scaling Type		$\succ$	$\succ$	$\succ$			$\succ$						$\succ$			$\succ$					
Decision 2.3 - Select Elasticity Automation Degree										$\succ$											
Decision 2.4 - Select Scaling Trigger									$\succ$												

$\succ$: influences $\vdash$: determines

Table A.2.: Influences of Decision Point 2

	Decision Point 3 - Define Multi-Tenancy Requirements	Decision 3.1 - Select Kind of Multi-tenancy	Decision 3.2 - Select Multi-tenancy Architecture
Decision 4.7 - Define Resource Location			
Decision 4.6 - Select Pricing Model			
Decision 4.5 - Select Cloud Vendor			
Decision 4.4 - Define Roles of Responsibility			
Decision 4.3 - Define Cloud Hosting			
Decision 4.2 - Select Cloud Service Model			⅄
Decision 4.1 - Select Cloud Deployment Model			
Decision Point 4 - Select Service Provider / Offering			
Decision 3.2 - Select Multi-tenancy Architecture		⅄	
Decision 3.1 - Select Kind of Multi-tenancy			⊢
Decision Point 3 - Define Multi-Tenancy Requirements			
Decision 2.4 - Select Scaling Trigger			
Decision 2.3 - Select Elasticity Automation Degree			
Decision 2.2 - Select Scaling Type			⅄
Decision 2.1 - Define Scalability Level			⅄
Decision Point 2 - Define Elasticity Strategy			
Decision 1.4 - Select Migration Type			⅄
Decision 1.3 - Select Application Components			⅄
Decision 1.2 - Select Application Tier			⅄
Decision 1.1 - Select Application Layer			⅄
Decision Point 1 - Application Distribution			

⅄ : influences ⊢ : determines

Table A.3.: Influences of Decision Point 3

	Decision Point 4 - Select Service Provider / Offering	Decision 4.1 - Select Cloud Deployment Model	Decision 4.2 - Select Cloud Service Model	Decision 4.3 - Define Cloud Hosting	Decision 4.4 - Define Roles of Responsibility	Decision 4.5 - Select Cloud Vendor	Decision 4.6 - Select Pricing Model	Decision 4.7 - Define Resource Location
Decision 4.7 - Define Resource Location						⅄		
Decision 4.6 - Select Pricing Model			⅄					
Decision 4.5 - Select Cloud Vendor								
Decision 4.4 - Define Roles of Responsibility		⅄	⅄	⅄				
Decision 4.3 - Define Cloud Hosting		⅄			⅄			⅄
Decision 4.2 - Select Cloud Service Model						⅄		
Decision 4.1 - Select Cloud Deployment Model				⅄	⅄			
Decision Point 4 - Select Service Provider / Offering								
Decision 3.2 - Select Multi-tenancy Architecture			⅄					
Decision 3.1 - Select Kind of Multi-tenancy								
Decision Point 3 - Define Multi-Tenancy Requirements								
Decision 2.4 - Select Scaling Trigger								
Decision 2.3 - Select Elasticity Automation Degree								
Decision 2.2 - Select Scaling Type								
Decision 2.1 - Define Scalability Level			⅄					
Decision Point 2 - Define Elasticity Strategy								
Decision 1.4 - Select Migration Type			⅄	⅄				
Decision 1.3 - Select Application Components								
Decision 1.2 - Select Application Tier								
Decision 1.1 - Select Application Layer								
Decision Point 1 - Application Distribution								

⅄ : influences ⊢ : determines

Table A.4.: Influences of Decision Point 4

A.2. Questionnaire

This section of the appendix shows the survey questionnaire was used for the evaluation of the elaborated Decision Support Framework (eDSF) in Chapter 4. It is a print version of the questionnaire exported from the open source survey application LimeSurvey, which was used to conduct the web-based survey.

Survey on Decision Support Framework for Application Migration to the Cloud - Master's Thesis Alexander Darsow

This survey in form of a questionnaire is a part of the thesis of Alexander Darsow in the Master degree course of Business Informatics (Wirtschaftsinformatik) at the University of Hohenheim in 2013 / 2014.

The expiration date of this survey is Saturday, 15th of February 2014

Any data of this survey will be used in terms of the evaluation of this Master’s thesis exclusively.
Results are used anonymously and no personal data will be passed for any other use!

- - - - - - - - - -

Introduction

Analysts like Gartner ensure ”Cloud computing is set to have a considerable impact on business in the future” which is also proven by continued growth of the Cloud service market today, since enterprises consider the Cloud also in terms of productive applications rather than just for development or testing scenarios. In this sense, one of the key interests of enterprises is to *migrate their existing applications* from in-house to the Cloud. But this Cloud adoption approach can be difficult especially in case of business-critical applications.
To aid enterprises in this challenging venture Cloud service provider (e.g. IBM or HP) started to offer consultancy in Cloud Computing adoption and application migration. But in addition, this field started to gain attention of research with meanwhile several approaches published to address the application migration to the Cloud issue.

Andrikopoulos et al. (PDF-Link) published one of the most recent approaches in this field of research. It considers a vision of a *Decision Support System for Cloud Migration* that supports decision makers by providing a conceptual view of which *Decision Points* have to be attended in the multi-dimension problem of application migration to the Cloud and which supporting *Tasks* are related to them.
The Master’s thesis is based on this work and is majorly focused to elaborate and refine the current conceptual vision of the Decision Support Framework in terms of its Decision Points to aid decision-making in terms of application migration to the Cloud.

Structure of the Questionnaire

The questionnaire is structured as followed:

- A few general questions related to the participant
- An introduction to the Decision Support Framework with follow-up questions
- Questions related to each elaborated Decision and Decision Point
- Closing questions for further information

Hint

*Questions marked with * are obligatory!*

General Questions to the Respondent

Name (Optional):

Please write your answer here:

Company / Organization: *

Please write your answer here:

Job role: *

Please write your answer here:

Please rate your experience level in terms of... *

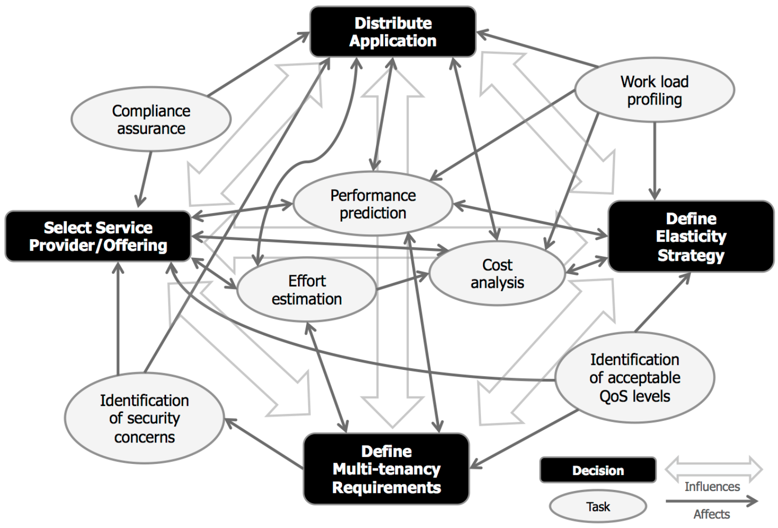
Please choose the appropriate response for each item:

	0	1	2	3	4	5	6	7	8	9
Cloud Computing:	○	○	○	○	○	○	○	○	○	○
IT in general:	○	○	○	○	○	○	○	○	○	○

[0 = no experience, 1 = very poor experience, 9 = excellent experience]

The Decision Support Framework for Application Migration to the Cloud

The *conceptual Decision Support Framework for Application Migration to the Cloud*



This conceptual view of the Decision Support Framework (DSF) stated a vision for what constitutes a complete solution for application developers and stakeholders alike that are considering whether and how to migrate their application to the Cloud.

Decision Point 1 - Distribute Application
How to distribute the application across service providers and between the Cloud and the local data center?

Decision Point 2 - Define Elasticity Strategy
Which is the elasticity strategy that the application needs to implement in order to cope with its demand in the face of SLAs and expectations of its users.

Decision Point 3 - Define Multi-Tenancy Requirements
What are the requirements of the application in terms of multi-tenancy, i.e. to what extent the existing application is required to support multi- tenancy, to what degree it is designed for this purpose, and how it should be (re-)engineered to support multi-tenancy.

Decision Point 4 - Select Service Provider / Offering
How to select a (Cloud) service provider and offering that fits the application needs in terms of cost, expected performance, compliance requirements, etc.

Task - Work Load Profiling
Defining or estimating the expected work load profile of the application.

Task - Compliance Assurance
Ensuring the compliance to regulations regarding, e.g., privacy of personal data.

Task - Identification of Security Concerns
Defining which data and communications are critical to be protected.

Task - Identification of Acceptable QoS Levels
Based on existing and planned SLAs, acceptable levels for QoS characteristics like availability of the service provider can be inferred.

Task - Performance Prediction
This task creates performance goals and the non-functional behavior of the application after it is migrated to the Cloud.

Task - Cost Analysis
Based on made decisions costs of the application migration and its future running cost are estimated and may be considered with respect to given thresholds.

Task - Effort Estimation
This task focuses on providing an estimate related to the amount of work required to adapt the application depending on how it's migrated.

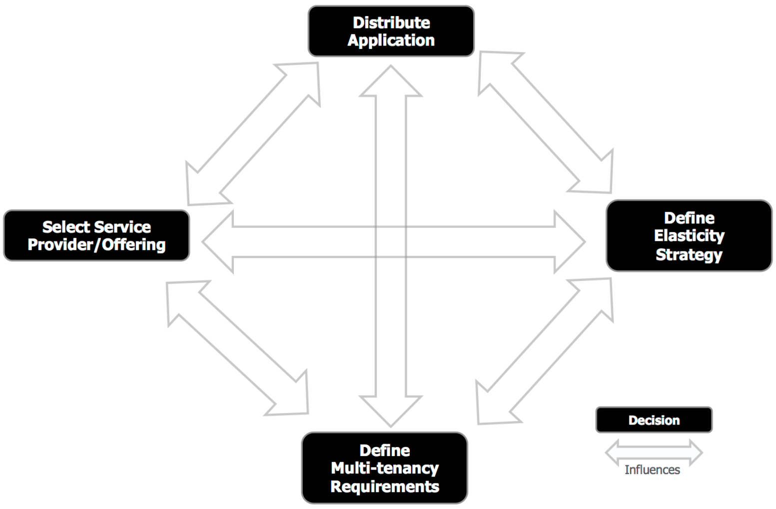
With regards to the Decision Support Framework... *

Please choose the appropriate response for each item:

	Not at all understandable	Slightly understandable	Moderately understandable	Very understandable	Absolutely understandable
Is the differentiation between Decisions Points and Tasks understandable?	○	○	○	○	○
Is the meaning of the presented Decision Points understandable?	○	○	○	○	○
Is the meaning of the presented Tasks understandable?	○	○	○	○	○

Overview Decision Points

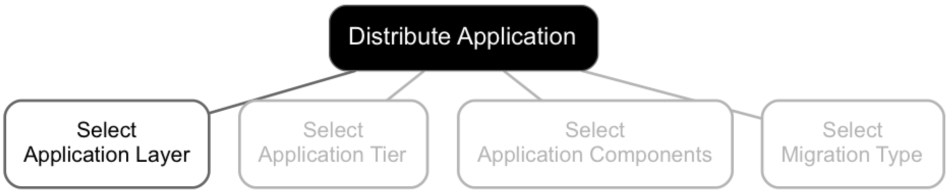
Overview Decision Points



In the following Decision Points will be passed through one by one and each concrete Decision elaborated will be addressed with questions.

Decision 1.1 - Select Application Layer

Decision 1.1 - Select Application Layer



Description

This decision considers encapsulating the functionality of an application migrated to the Cloud in terms of logical separation into *layers* according to the three layers pattern of Fowler.

By this approach the following three layers are distinguished:

- Presentation layer, with functionality regarding the interaction and requests between user and application.
- Business layer, which contains the actual application logic.
- Data layer, where databases and functionality related to database communication and interaction are arranged

Possible Outcomes of this Decision

- Presentation Layer
- Business Layer
- Data Layer
- Multiple Layers

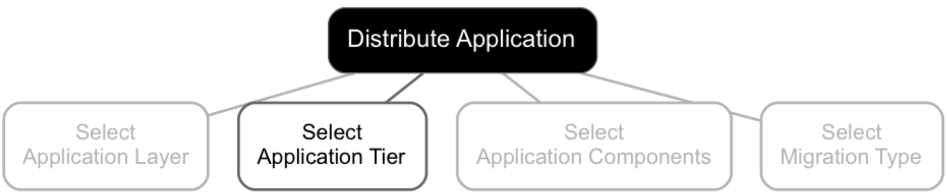
With regards to this Decision... *

Please choose the appropriate response for each item:

	Not at all	Slightly	Moderately	Very	Absolutely
Is the meaning of this Decision understandable?	○	○	○	○	○
Is this Decision relevant to be considered in this Decision Point?	○	○	○	○	○

Decision 1.2 - Select Application Tier

Decision 1.2 - Select Application Tier



Description

This decision pays attention at the physical architecture of an application migrated to the Cloud and distinguishes several *tiers* of an application. A tier comprises physical infrastructure (e.g. application and/or database server) that host implemented application functionality.

Usually logical functionality arranged on a layer (Decision 1.1) is hosted on physical components on the corresponding tier (i.e. functionality of the data layer is implemented in components on the data tier).

This approach distinguishes the following tiers:

- Client tier, e.g. the users web browser to access an web application.
- Application tier, e.g. application server running the actual application.
- Data tier, e.g. databases and/or database server of the application.

Possible Outcomes of this Decision

- Client Tier
- Application Tier
- Data Tier
- Multiple Tiers

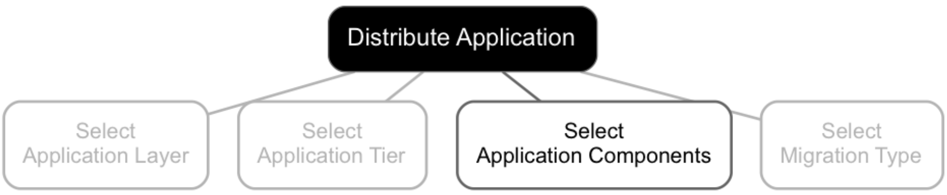
With regards to this Decision... *

Please choose the appropriate response for each item:

	Not at all	Slightly	Moderately	Very	Absolutely
Is the meaning of this Decision understandable?	○	○	○	○	○
Is this Decision relevant to be considered in this Decision Point?	○	○	○	○	○

Decision 1.3 - Select Application Components

Decision 1.3 - Select Application Commponents



Description

This decision refers to considering the granularity of the application architecture if the layers and tiers views are not appropriate or sufficient.

Such a distinction has to be made individually by investigation the application migrated to the Cloud, thus a general listing of components cannot be provided.

As a result if a selection of components is considered it can be chosen either to select a single component or multiple components.

Possible Outcomes of this Decision

- Single Component
- Multiple Components

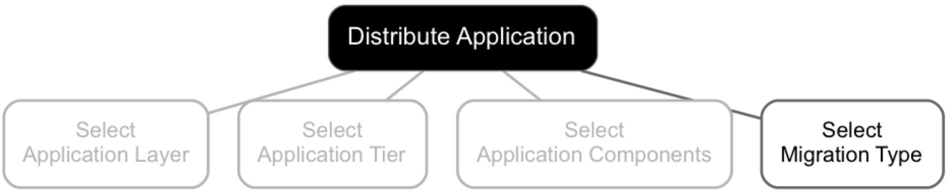
With regards to this Decision... *

Please choose the appropriate response for each item:

	Not at all	Slightly	Moderately	Very	Absolutely
Is the meaning of this Decision understandable?	○	○	○	○	○
Is this Decision relevant to be considered in this Decision Point?	○	○	○	○	○

Decision 1.4 - Select Migration Type

Decision 1.4 - Select Migration Type



Description

Considering the migration of an application to the Cloud from an application point of view, there are different scenarios regarding its architecture and distribution how a migration can be achieved. Based on this consideration Andrikopoulos et al. (2013) (PDF-Link) have identified four application migration types.

- Type I - Replacement of component(s) with Cloud offerings, i.e. one or more components of an application are substituted by cloud services (e.g. a MySQL database by a Google App Engine Datastore).
- Type II - Partially migration of application functionality, i.e. whole application functionalities are moved to the Cloud, which may involve several architectural components on one or more application layers.
- Type III - Whole software stack migration, i.e. a migration of a whole software stack where an existing application is encapsulated in a virtual machine (VM) and deployed on an IaaS.
- Type IV - Application cloudification, i.e. the application is completely re-constructed by implementing the applications functionalities out of a combination of Cloud services.

Possible Outcomes of this Decision

- Type I
- Type II
- Type III
- Type IV

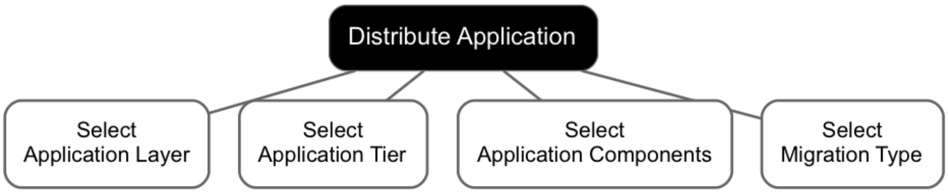
With regards to this Decision... *

Please choose the appropriate response for each item:

	Not at all	Slightly	Moderately	Very	Absolutely
Is the meaning of this Decision understandable?	○	○	○	○	○
Is this Decision relevant to be considered in this Decision Point?	○	○	○	○	○

Decision Point 1 - Distribute Application

Decision Point 1 - Distribute Application



With regards to this Decision Point... *

Please choose the appropriate response for each item:

	Not at all	Slightly	Moderately	Very	Absolutely
Is this Decision Point relevant to be considered in terms of Application Migration to the Cloud?	○	○	○	○	○
Are the Decisions appropriate to be considered within this Decision Point?	○	○	○	○	○

Do you miss any Decision(s)? *

Please choose **only one** of the following:

○ Yes

○ No

What Decision(s) do you miss in this Decision Point?

Only answer this question if the following conditions are met:
Answer was 'Yes' at question '18 [QG10Q2]' (Do you miss any Decision(s)?)

Please write your answer(s) here:

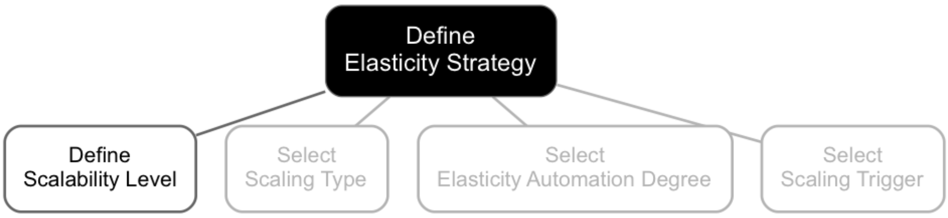
#1

#2

#3

Decision 2.1 - Define Scalability Level

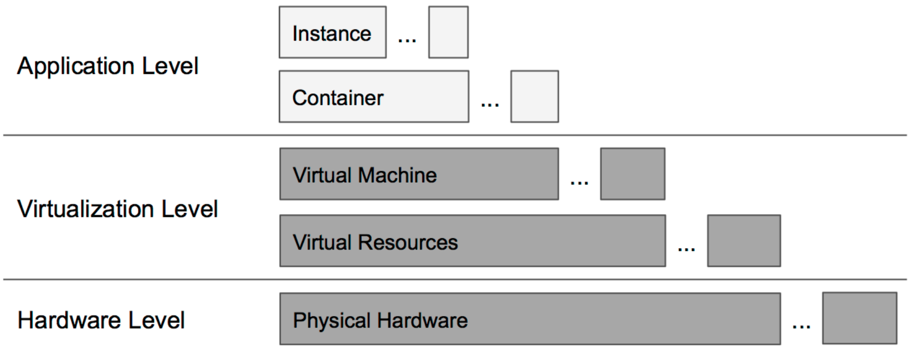
Decision 2.1 - Define Scalability Level



Description

In terms of elasticity the characteristic of scalability can be seen as a prerequisite for resources comprising all hardware, virtualization and software. In this sense, scalability can be performed on different levels (e.g. hardware, virtualization, etc.).

Different scalability levels are enabled by Cloud Computing. A classification of theses levels is shown in the following figure.



Possible Outcomes of this Decision

- Instance Level (e.g. application or database instance)
- Container Level (e.g. application or database server)
- Virtual Machine Level (i.e. Virtual Machine)
- Virtual Resource Level (i.e. virtual recourses like CPU, RAM, network, etc.)
- Physical Hardware Level (i.e. physical resources like servers)
- Multiple Levels

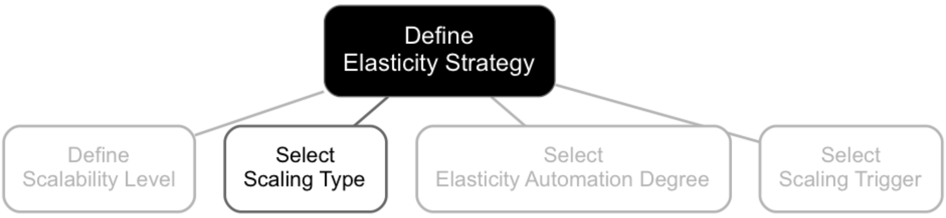
With regards to this Decision... *

Please choose the appropriate response for each item:

	Not at all	Slightly	Moderately	Very	Absolutely
Is the meaning of this Decision understandable?	○	○	○	○	○
Is this Decision relevant to be considered in this Decision Point?	○	○	○	○	○

Decision 2.2 - Select Scaling Type

Decision 2.2 - Select Scaling Type



Description

With regards to the previous introduced scalability levels (Decision 2.1) where scalability can be applied, another consideration on how scaling is performed can be made. In this sense the following types of scaling are distinguished:

- <u>Vertical scaling (aka scaling up-down)</u>, i.e. assigning or removing additional underlying resources to the own level
- <u>Horizontal scaling (aka scaling out-in)</u>, i.e. replicating or removing instances on the same level.
- <u>Hybrid scaling</u>, i.e. a scaling in both directions vertical as well as horizontal.

Possible Outcomes of this Decision

- Vertical scaling
- Horizontal scaling
- Hybrid scaling

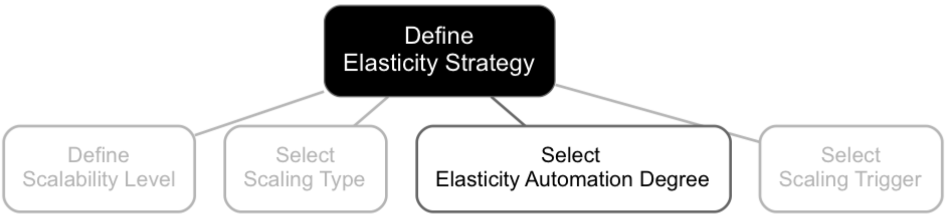
With regards to this Decision... *

Please choose the appropriate response for each item:

	Not at all	Slightly	Moderately	Very	Absolutely
Is the meaning of this Decision understandable?	○	○	○	○	○
Is this Decision relevant to be considered in this Decision Point?	○	○	○	○	○

Decision 2.3 - Define Elasticity Automation Degree

Decision 2.3 - Define Elasticity Automation Degree



Description

This decision considers which degree of automation in scaling is applied.

For example, Cloud Offerings often promote their services with the term *automatic scaling* (auto-scaling), but in reality user-defined rules are taken into account to determine when (e.g. exceeding what threshold), what (e.g. VM instance), and how much (e.g. one, two, etc.) resources are scaled. Such a case of scaling is more likely semi-automatic scaling rather then automatic scaling.

In terms of scaling resources in the Cloud the following degrees of automation are possible to be considered:

- Manual scaling, e.g. manual apply a hardware server.
- Semi-automatic scaling, e.g. scaling IaaS based on defined rules.
- Automatic scaling, e.g. anticipate workloads based on log files, past workload, etc.

Possible Outcomes of this Decision

- Manual scaling
- Semi-automatic scaling
- Automatic scaling

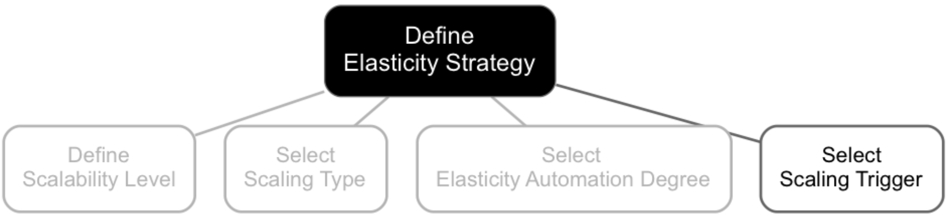
With regards to this Decision... *

Please choose the appropriate response for each item:

	Not at all	Slightly	Moderately	Very	Absolutely
Is the meaning of this Decision understandable?	○	○	○	○	○
Is this Decision relevant to be considered in this Decision Point?	○	○	○	○	○

Decision 2.4 - Select Scaling Trigger

Decision 2.4 - Select Scaling Trigger



Description

This decision refers to the trigger used to initiate scaling. The amount of time a scaled resource needs to be actually ready for consumption (e.g. time for starting a VM) is referred as scaling latency or spin-up time. As this amount of time affects performance and costs related to the application two approaches are mentioned to trigger scaling:

- Event-driven, e.g. react based on monitoring scaling rules.
- Proactive, e.g. predict scaling action based on log files and/or real-time measures.

Possible Outcomes of this Decision

- Event-driven
- Proactive

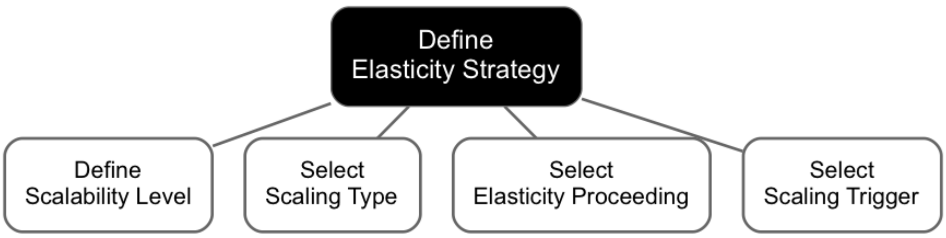
With regards to this Decision... *

Please choose the appropriate response for each item:

	Not at all	Slightly	Moderately	Very	Absolutely
Is the meaning of this Decision understandable?	○	○	○	○	○
Is this Decision relevant to be considered in this Decision Point?	○	○	○	○	○

Decision Point 2 - Define Elasticity Strategy

Decision Point 2 - Define Elasticity Strategy



With regards to this Decision Point... *

Please choose the appropriate response for each item:

	Not at all	Slightly	Moderately	Very	Absolutely
Is this Decision Point relevant to be considered in terms of Application Migration to the Cloud?	○	○	○	○	○
Are the Decisions appropriate to be considered within this Decision Point?	○	○	○	○	○

Do you miss any Decision(s)? *

Please choose **only one** of the following:

- ○ Yes
- ○ No

What Decision(s) do you miss in this Decision Point?

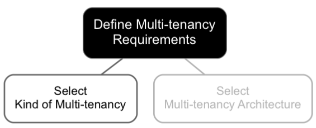
Only answer this question if the following conditions are met:
Answer was 'Yes' at question '30 [QG20Q2]' (Do you miss any Decision(s)?)

Please write your answer(s) here:

#1
#2
#3

Decision 3.1 - Select Kind of Multi-Tenancy

Decision 3.1 - Select Kind of Multi-Tenancy



Description

This decision addresses what kind of multi-tenancy is chosen for an application migrated to the Cloud if such requirements are exposed.

In literature the characteristic of multi-tenancy is basically distinguished into two kinds regarding isolation vs. sharing of the consumed Cloud resources of multiple tenants.

- Multiple instances multi-tenancy, e.g. tenants are supported with dedicated resource instances in some kind (e.g. dedicated application instance, dedicated VM, dedicated hardware resources, etc.).
- Native multi-tenancy, e.g. tenants share a single application based on shared resources below.

Possible Outcomes of this Decision

- Multiple instance multi-tenancy
- Native multi-tenancy

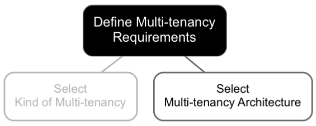
With regards to this Decision... *

Please choose the appropriate response for each item:

	Not at all	Slightly	Moderately	Very	Absolutely
Is the meaning of this Decision understandable?	○	○	○	○	○
Is this Decision relevant to be considered in this Decision Point?	○	○	○	○	○

Decision 3.2 - Select Multi-Tenancy Architecture

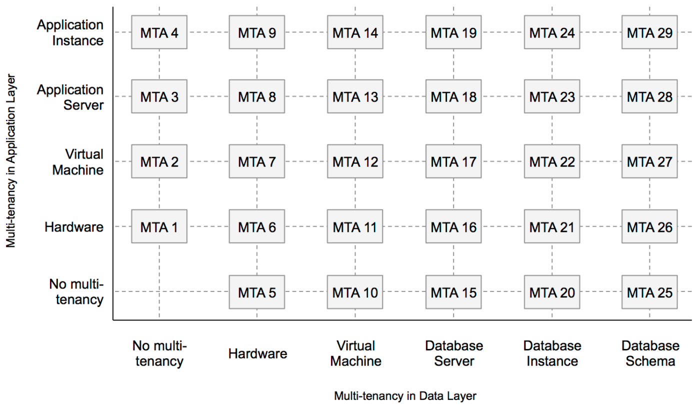
Decision 3.2 - Select Multi-Tenancy Architecture



Description

This decision considers possible multi-tenancy architectures how this characteristic can be applied on different resource level and at different parts of the application.

With respect to the former, several resource levels like hardware, virtual machine, instance, etc. can be differentiated. The latter considers application part in terms of application tiers (Decision 1.2) whereby it's assumed only the application and data tier is relevant to be considered due to the fact the accessing web browser is assigned to the client tier which is negligible. The approach is borrowed from Pors (2013) (Link) and adapted in terms of this Master's thesis as shown in the following figure:



Possible Outcomes of this Decision

- One of the Multi-Tenancy Architectures depicted above

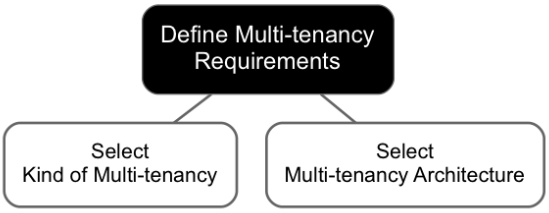
With regards to this Decision... *

Please choose the appropriate response for each item:

	Not at all	Slightly	Moderately	Very	Absolutely
Is the meaning of this Decision understandable?	○	○	○	○	○
Is this Decision relevant to be considered in this Decision Point?	○	○	○	○	○

Decision Point 3 - Define Multi-Tenancy Requirements

Decision Point 3 - Define Multi-Tenancy Requirements



With regards to this Decision Point... *

Please choose the appropriate response for each item:

	Not at all	Slightly	Moderately	Very	Absolutely
Is this Decision Point relevant to be considered in terms of Application Migration to the Cloud?	○	○	○	○	○
Are the Decisions appropriate to be considered within this Decision Point?	○	○	○	○	○

Do you miss any Decision(s)? *

Please choose **only one** of the following:

○ Yes

○ No

What Decision(s) do you miss in this Decision Point?

Only answer this question if the following conditions are met:
Answer was 'Yes' at question '38 [QG30Q2]' (Do you miss any Decision(s)?)

Please write your answer(s) here:

#1

#2

#3

Decision 4.1 - Select Cloud Deployment Model

Decision 4.1 Select Cloud Deployment Model

Description

This decision attends to one of the key differentiations in Cloud Computing, namely the Deployment Model that considers with whom are Cloud resources shared. With reference to the commonly used definition of Cloud Computing provided by the National Institute of Standards and Technology (NIST) four deployment models are known:

- Private Cloud, where Cloud resources are exclusive for the consumer.
- Community Cloud, where Cloud recourses are shared among a certain number of consumers.
- Public Cloud, where Cloud resources are shared among an unknown number of consumers.
- Hybrid Cloud, where Cloud resources are shared based on a combination of two or more of the former models

Possible Outcomes of this Decision

- Private Cloud
- Community Cloud
- Public Cloud
- Hybrid Cloud

With regards to this Decision... *

Please choose the appropriate response for each item:

	Not at all	Slightly	Moderately	Very	Absolutely
Is the meaning of this Decision understandable?	○	○	○	○	○
Is this Decision relevant to be considered in this Decision Point?	○	○	○	○	○

Decision 4.2 - Select Cloud Service Model

Decision 4.2 - Select Cloud Service Model

Description

This decision addresses the other major differentiation in Cloud Computing, namely Service Model, which considers what kind of resource the Cloud provides as a Service. Again, with regards to the Cloud Computing definition of NIST the following three service models are defined:

- Software as a Service (SaaS), provides on-demand software application to the consumer.
- Platform as a Service (PaaS), provides resources to deploy consumer-created or acquired application.
- Infrastructure as a Service (IaaS), provides IT infrastructure to the consumer.

Possible Outcomes of this Decision

- Software as a Service (SaaS)
- Platform as a Service (PaaS)
- Infrastructure as a Service (IaaS)

With regards to this Decision... *

Please choose the appropriate response for each item:

	Not at all	Slightly	Moderately	Very	Absolutely
Is the meaning of this Decision understandable?	○	○	○	○	○
Is this Decision relevant to be considered in this Decision Point?	○	○	○	○	○

Decision 4.3 - Define Cloud Hosting

Decision 4.3 - Define Cloud Hosting

Description

This decision refers to the where the services and resources are hosted. Regarding the perspective of the Cloud consumer hosting can either be in-house (aka. on-premise), when the resources are for instance located at an own datacenter or external (aka. off-premise) when they are host by an external 3rd party. In addition, a combination on in-house and external hosting might be possible in terms of for example extending in-house resource by additional external resource if they are needed.

Possible Outcomes of this Decision

- On-Premise Cloud Hosting

- Off-Premise Cloud Hosting
- Hybrid Cloud Hosting

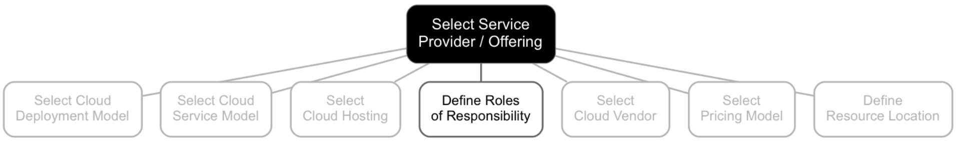
With regards to this Decision... *

Please choose the appropriate response for each item:

	Not at all	Slightly	Moderately	Very	Absolutely
Is the meaning of this Decision understandable?	○	○	○	○	○
Is this Decision relevant to be considered in this Decision Point?	○	○	○	○	○

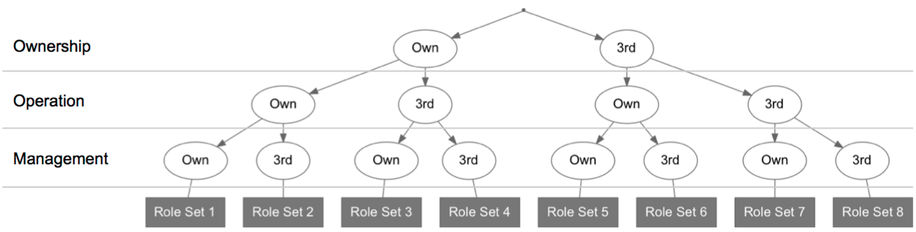
Decision 4.4 - Define Roles of Responsibility

Decision 4.4 - Define Roles of Responsibility



Description

This decision considers several roles that can be hold in terms of the Cloud service, mostly regarding the infrastructure. According to the NIST definition these roles are ownership, operation, and management. Picturing this from the Cloud consumer point of view a role can either be hold by the consumer himself or be delegated to a 3rd party. Regarding three roles where each can either be hold by the consumer or a 3rd party, this results in a total of 8 possible responsibility sets that are illustrated in the following figure.



Possible Outcomes of this Decision

- Either one of the previous depicted Responsibility Roles Sets

With regards to this Decision... *

Please choose the appropriate response for each item:

	Not at all	Slightly	Moderately	Very	Absolutely
Is the meaning of this Decision understandable?	○	○	○	○	○
Is this Decision					

relevant to be considered in this Decision Point?	○	○	○	○	○

Decision 4.5 - Select Cloud Vendor

Decision 4.5 - Select Cloud Vendor

Description

This decision refers to the selection of a Cloud Vendor providing services. Whereas the majority of decision considers “hard facts” in term of e.g. function requirements of the application, this one is desired to select a Cloud Vendor suitable for the application migration approach in terms of “soft facts” like for instance resources, knowledge, skills, etc. evaluated based on reference projects, benchmarks, reports etc.

Such an evaluation is specific for each case of application migration and thus a general statement of certain Cloud Vendors as possible outcomes is not provided.

Possible Outcomes of this Decision

- Evaluated Cloud Vendor

With regards to this Decision... *

Please choose the appropriate response for each item:

	Not at all	Slightly	Moderately	Very	Absolutely
Is the meaning of this Decision understandable?	○	○	○	○	○
Is this Decision relevant to be considered in this Decision Point?	○	○	○	○	○

Decision 4.6 - Select Pricing Model

Decision 4.6 - Select Pricing Model

Description

This decision pays attention at the pricing models possible in Cloud Computing. As the Cloud Computing paradigm is associated with the "pay as you go" pricing approach, over the years several pricing models evolved for varying offerings.

- Free, where the service or Cloud resources are provided free of charge.
- Pay-Per-Use, where Cloud resources are charged based on their actual usage time.
- Pay-Per-Unit, where Cloud resources are charged based on unit per time usage.
- Charge-Per-Use (Subscription), where a defined amount of Cloud resources is provided to the consumer and charged upfront.
- Combined Pricing Model, a combination of two or more of the stated pricing models.

Possible Outcomes of this Decision

- Free
- Pay-Per-Use
- Pay-Per-Unit
- Charge-Per-Use (Subscription)
- Combined Pricing Model

With regards to this Decision... *

Please choose the appropriate response for each item:

	Not at all	Slightly	Moderately	Very	Absolutely
Is the meaning of this Decision understandable?	○	○	○	○	○
Is this Decision relevant to be considered in this Decision Point?	○	○	○	○	○

Decision 4.7 - Define Physical Cloud Resource Location

Decision 4.7 - Define Physical Cloud Resource Location

Description

This decision considers the actual physical location of the resources provided by the Cloud service. While Decision 4.3 selects the resources to be in-house or external this decision considers the actual location due to the fact where data is actually stored and processed might be relevant in terms of compliant regulations of the enterprise or law.

Again, this decision has to be made based on an evaluation with respect to the individual application that is migrated, the Cloud service and its vendor's location, organizational limitation, etc.

Possible Outcomes of this Decision

- Evaluated Physical Cloud Resource Location

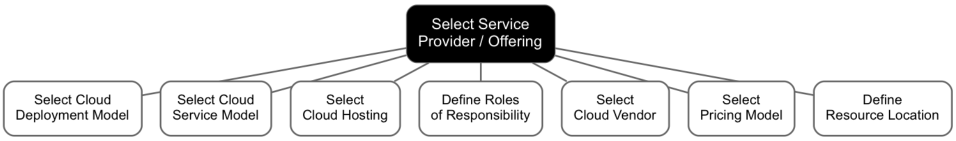
With regards to this Decision... *

Please choose the appropriate response for each item:

	Not at all	Slightly	Moderately	Very	Absolutely
Is the meaning of this Decision understandable?	○	○	○	○	○
Is this Decision relevant to be considered in this Decision Point?	○	○	○	○	○

Decision Point 4 - Select Service Provider / Offering

Decision Point 4 - Select Service Provider / Offering



With regards to this Decision Point... *

Please choose the appropriate response for each item:

	Not at all	Slightly	Moderately	Very	Absolutely
Is this Decision Point relevant to be considered in terms of Application Migration to the Cloud?	○	○	○	○	○
Are the Decisions appropriate to be considered within this Decision Point?	○	○	○	○	○

Do you miss any Decision(s)? *

Please choose **only one** of the following:

○ Yes

○ No

What Decision(s) do you miss in this Decision Point?

Only answer this question if the following conditions are met:
Answer was 'Yes' at question '56 [QG40Q2]' (Do you miss any Decision(s)?)

Please write your answer(s) here:

#1

#2

#3

Closure Questions

Would you like to receive a link to the web-based implementation of the *Decision Support Framework or Application Migration to the Cloud* as soon as the Master's thesis is completed? *

Please choose **only one** of the following:

- ◯ Yes
- ◯ No

Would you like to receive an electronic copy of the Master's thesis *Decision Support for Application Migration to the Cloud* as soon as it is completed? *

Please choose **only one** of the following:

- ◯ Yes
- ◯ No

Please enter an E-Mail address:

Only answer this question if the following conditions are met:

-------- Scenario 1 --------

Answer was 'Yes' at question '58 [QG99Q1]' (Would you like to receive a link to the web-based implementation of the Decision Support Framework or Application Migration to the Cloud as soon as the Master's thesis is completed?)

-------- or Scenario 2 --------

Answer was 'Yes' at question '59 [QG99Q2]' (Would you like to receive an electronic copy of the Master's thesis Decision Support for Application Migration to the Cloud as soon as it is completed?)

Please write your answer here:

[ ]

This E-Mail address will only be used to provide you further information according to your selection above!

Thank you for participating in this survey and supporting my Master's thesis!

If you have any questions regarding this survey or the topic of my Master's thesis feel free to contact me via alexander.darsow@uni-hohenheim.de.

16.02.2014 – 19:33

Submit your survey.
Thank you for completing this survey.

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.

Ort, Datum, Unterschift

Erklärung gemäß § 14 Abs. 5 und Abs. 6 der Prüfungsordnung der Universitäten Hohenheim und Stuttgart für den Masterstudiengang Wirtschaftsinformatik

Hiermit erkläre ich, dass ich die Masterarbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Alle Stellen der Arbeit, die wörtlich oder sinngemäß aus Veröffentlichungen oder aus anderweitigen fremden Äußerungen entnommen wurden, sind als solche einzeln kenntlich gemacht.

Die Masterarbeit habe ich noch nicht in einem anderen Studiengang als Prüfungsleistung verwendet.

Des Weiteren erkläre ich, dass mir weder an den Universitäten Hohenheim und Stuttgart noch an einer anderen wissenschaftlichen Hochschule bereits ein Thema zur Bearbeitung als Masterarbeit oder als vergleichbare Arbeit in einem gleichwertigen Studiengang vergeben worden ist.

Stuttgart-Hohenheim, den ____________________________________

Unterschrift: ____
(als Originalunterschrift in beiden Exemplaren der Masterarbeit; **nicht** als Kopie)