VIENNA UNIVERSITY OF ECONOMICS AND BUSINESS

Institute for Management Information Systems

ao. Univ.Prof. Mag. Dr. Rony G. Flatscher

Bachelor Thesis

A Holistic View of Cloud Computing

submitted by:

Student:	Lukas Fischbach
Program:	Bachelor Program in Business, Economics and Social Sciences
Matriculation Number:	1054467
Date of Birth:	21.09.1991
E-Mail:	h1054467@wu.ac.at

Vienna, 3 February 2014

Erklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe. Alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten und nicht veröffentlichten Schriften entnommen wurden, sind als solche kenntlich gemacht. Die Arbeit ist in gleicher oder ähnlicher Form oder auszugsweise im Rahmen einer anderen Prüfung noch nicht vorgelegt worden.

Vienna, 3 February 2014

_____

Lukas Fischbach

Table of Contents

List of Figures

1 Introduction

This chapter is designed to give an overview of the background of this thesis. It states the motivation of the author and the structure of this thesis.

1.1 Motivation

During an internship at a large IT enterprise in summer 2012, the author of this thesis was confronted with the term “Cloud Computing” for the first time. He was responsible for creating a tool that collects information about customer needs and suggests an appropriate cloud offering. At first, he did not know much about cloud computing. The author had vague perceptions about this topic, but never engaged himself strongly with it. During his work, he had to strengthen his knowledge about cloud computing to be able to handle his job properly. Consequently, the author started to read books and articles about cloud computing. This learning process was the first spark that ignited his interest in this topic.

After the internship, the author wanted to further expand his knowledge about cloud computing. Within the scope of a course at his university, he could write a seminar paper about the practical application of cloud computing [48]. This paper describes how customers can access and use various cloud services from different cloud providers. The focus of that paper lies on the practical parts of cloud computing, which perfectly fitted the scope of the course. However, the author also wanted to describe his experiences and learn more about the theoretical parts of this topic. Therefore, he suggested to dedicate this bachelor thesis to the cloud computing backgrounds, models, advantages and also considerations for cloud computing.

1.2 Structure

This thesis is designed to give a general overview of important topics of cloud computing and is structured as follows.

Chapter two gives an overview of this thesis by presenting the research questions on which this bachelor thesis is based and the approach of the author to answer them. Chapter three deals with the specific description of cloud computing. It gives a definition and presents an architecture to illustrate the complex relationships in the context of cloud computing. It also mentions classifications with regard to service models and

deployment models. The last sub-chapter of chapter three describes platforms that can be used to manage cloud infrastructures. Chapter four deals with quality of service aspects of cloud computing. It illustrates criteria that are important and presents ways to assure that a certain level of quality of service is delivered by presenting an overview of service level agreements (SLAs). Chapter five is designed to discuss the advantages and disadvantages of cloud computing and also presents a conclusion. The last chapter gives an outlook by pointing out predictions and possible developments in the context of cloud computing.

2 Overview

This chapter lists the goal of this thesis by presenting the research questions that are answered and describes the chosen approach of the author.

2.1 Research Questions

Within the scope of this thesis the following research questions are answered:

- What is cloud computing?
  - Which technologies are used to enable cloud computing?
  - What are possible classifications of cloud computing …
    - … in terms of service models?
    - … in terms of deployment models?
  - How is it possible for cloud providers to guarantee certain quality criteria?
  - What are the advantages of cloud computing?
  - What are the disadvantages of cloud computing and which considerations do customers have to take into account?

2.2 Approach

This thesis is designed to give a holistic view of cloud computing. Therefore, it was necessary to greatly study recent scientific papers, news articles and also cloud offerings from different providers. It was important to build up knowledge of this topic to be able to understand different models in their respective context. This new knowledge was used to describe these models while trying to observe them from different point of views. It was essential to not only identify the advantages of cloud computing, but also the disadvantages. Consequently, a critical evaluation of the various approaches of

cloud computing was carried out while considering the current opinions of experts and the scientific community. In the end, this new knowledge and the respective conclusions were summarized in this thesis.

3 Cloud Computing – Definition and Description

This chapter deals with the description of cloud computing. It presents a definition of the term "cloud computing" and gives a classification of the various components of cloud computing. This is done by displaying an architecture that divides cloud computing into different layers. In addition, ways of deploying cloud environments are expressed and also means to control them by using software tools.

3.1 What is Cloud Computing?

The National Institute of Standards and Technology (NIST) [84] developed the following definition of 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. This cloud model is composed of five essential characteristics, three service models, and four deployment models.*" [80]

Essential for this definition are the adjectives ubiquitous, convenient and on-demand. That means, users do not have to know anything about the underlying infrastructure. Moreover, they can access desired resources at any time. What is more, the term "resources" is quite broadly defined. Resources that can be accessed are not limited to rare processing power. Storage, networks, applications and services can be also demanded via cloud computing. That turns cloud computing into a wide spread model that can find its use in many different situations.

The National Institute of Standards and Technology also states, that cloud computing is composed of five essential characteristics. These characteristics are the following [80]:

- On-demand self service

  As stated in the definition above, self-service is a key-criteria of cloud comput-

ing. Users must be able to access resources by themselves (e.g. via a webinterface) and without requiring additional human interaction.

- Broad network access

  The services can be consumed via a wide range of end-user platforms (e.g. workstation, smart phone, etc) over a network. Consequently, a fast and reliable network access must be present.

- Resource pooling

  Providers use physical machines which are pooled. These machines can be used to create virtualized resources that are dynamically assigned to customers on demand. Customers can use those resources, but may not be able to control exactly where those resources are geographically located. It is possible that resources assigned to a customer come from different data centers all over the world. However, service providers may offer customers to specify certain regions where the resources are hosted (e.g. Europe, USA, ...). This feature can be necessary when dealing with legal requirements.

- Rapid elasticity

  Cloud computing enables the rapid provision and removal of resources. This can be used to implement scaling features that allow users to quickly scale resources up and down (e.g. when dealing with peak loads) [22].

- Measured service

  The usage of resources is monitored and logged automatically by the provider for optimization purposes and to guarantee transparency.

The last sentence of the definition states that cloud computing also is composed of service and deployment models. Those models are described in chapters 3.3 and 3.4 on a large scale.

3.2 Architecture

There are various technologies used to enable this cloud computing model. Conse-

quently, it is important to understand the interaction of these technologies. In addition, there are interactions between the various services offered with cloud computing. It is possible that some services are based on, or make use of other services. To illustrate this interaction, the NIST also developed a model that is divided into three different layers. Each layer has a different functionality and some of them are again divided into smaller chunks. Figure 1 shows the complete model [77]. Other researchers also created models to illustrate the architecture of cloud computing (e.g. layered model of cloud computing by Zhang et al. [133]), but for this thesis the NIST model was chosen because it is the most appropriate and easy to understand.

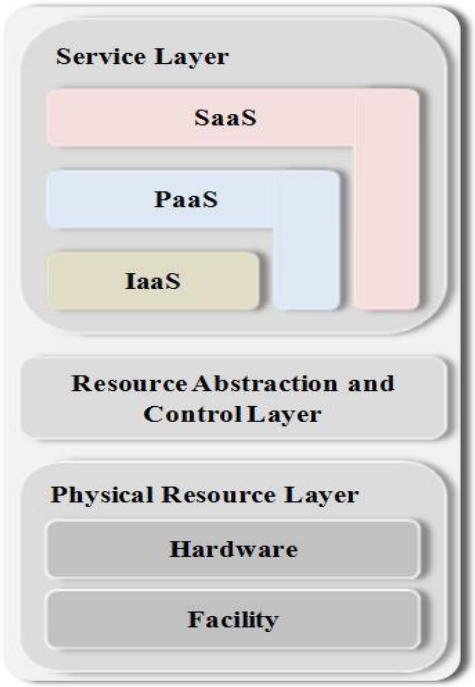

Figure 1: Cloud Computing Architecture [77]

3.2.1 Physical Resource Layer

The bottommost layer of the model is the "Physical Resource Layer". This layer represents the necessary hardware and facilities for cloud computing. This includes computational resources like CPUs and memory, but also network resources (router, switches, firewalls, etc), storage components and other physical devices needed for providing cloud services. In addition, facility resources like power supply, uninterpretable power supplies, cooling, etc are also required. All resources mentioned on this layer are most-

ly located in large data centers all over the world [77, 133].

3.2.2 Resource Abstraction Layer

The "Resource Abstraction Layer" contains the processes and tools cloud providers need to manage the underlying hardware and to supply users with services.

Virtualization is the key-technology on this layer. It is used to simulate the imagination of infinite resources because users are not directly confronted with the underlying infrastructure. Current technologies enable the creation and the deletion of virtual machines within seconds which enabled cloud computing. That allows fine-tuned adjustments of resources. Providers can offer their users to create customized virtual machines by themselves. In this way, users can easily scale their resources in use up and down by creating custom virtual machines [22, 77, 133].

There are various software products on the market that enable the virtualization of physical hardware. These products are highly leveraged in the context of cloud computing. Great examples for popular products are Xen [131], VMware vSphere Hypervisor [125], or KVM [75]. These products can be described as bare metal hypervisors. This means that they do not need an operating system to be able to run and can communicate directly with the physical hardware. Therefore, nearly all of the hardware resources can be used to run virtualized operating systems [74].

3.2.3 Service Layer

The service layer represents the connection to the user. On this layer, users can access various services offered with cloud computing. There are three "sub-layers" inside and every "sub-layer" represents a service model in the context of cloud computing. These three service models are: Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS). Every service offered with cloud computing can generally be allocated to exactly one of these service models. The allocation depends on the kind of the service and the features it offers. In addition, these service models build upon one another. That means, it is possible that SaaS services can be built on PaaS services and that PaaS services can use IaaS services [77, 80, 133]. Chapter 3.3.5 describes the relationships of these service models in greater detail.

The author of this thesis gives a short overview of each service layer in his seminar paper where he describes examples of how to use a specific service on each layer in

practice [48]. In this thesis, each service layer is described in much greater detail in the following chapters.

3.3 Service Models

Models with three layers are commonly used in the context of cloud computing to describe the kind of the service that is offered. A distinction between the models Infrastructure as a Service, Platform as a Service, and Software as a Service is widely used by various cloud providers and in science. What is more, the NIST has also adopted this classification [80]. However, there are different versions created by several scientists or cloud providers (e.g. with more layers, more differentiation) [76, 124, 128, 132, 133], but most of them are built upon this three layered model.

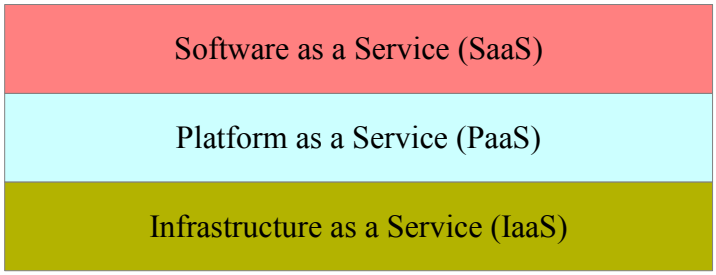

Figure 2: Cloud Computing Service Models

The following sub-chapters present each layer in greater detail. They describe how services can be allocated to a specific layer and what kind of services are offered on each layer.

Additionally, there is a description of a part of a more detailed model by Lenk et al. [76]. Lenk et al. created an architecture with a forth layer called “Human as a Service” that includes cloud-like work of humans instead of just raw computing resources.

3.3.1 Infrastructure as a Service (IaaS)

Services on the IaaS layer provide users with computing resources they can use to deploy and run arbitrary software. “Resources” in this context is a very broadly defined term. From this point of view, resources can be processing power, storage, memory, networks, etc. Within IaaS users do not have to handle the underlying infrastructure (i.e. the underlying hardware, cf. physical resource layer in chapter 3.2) but can use and configure the provided resources as they wish [80].

In practice, these resources are provided with the use of virtual machines. Users are able to request virtual machines with a desired configuration (processing power, storage, etc). The cloud provider creates these virtual machines and grants the users access to them (e.g. over the internet via SSH). Providers itself host these virtual machines within data centers. The computing resources in these data centers use hypervisors to manage the processes related to these virtual machines (cf. chapter 3.2.2). In addition, IaaS providers mostly use special software tools to offer their services on a convenient and on-demand basis. For example, customers can use a web-portal to request a desired virtual machine and it is created automatically without direct provider interaction [48]. What is more, most IaaS providers offer auto-scaling features that allow the automatic creation or deletion of virtual machines depending on certain criteria (e.g. CPU load) [7, 111].

Public IaaS providers commonly use pay-per-use pricing models to charge their customers. Within such a model, customers have to pay only for the resources they have used. In the practical context of such services, customers are generally charged for every hour a virtual machine is running, for the amount of data transferred, for the storage capacities used, and for other additional features [22].

Examples of IaaS products are Amazon Elastic Compute Cloud [6], Rackspace Cloud Servers [94], and GoGrid Cloud Servers [55].

3.3.2 Platform as a Service (PaaS)

PaaS services provide a platform that can be used to create, deploy, and run user-created applications. The platform provides necessary resources, but also required libraries, tools, compilers, etc. Users do not have to manage the underlying infrastructure by themselves. The provider is in charge of managing and maintaining it. Users only have to create their applications and deploy it onto the cloud-platform [80]. The applications deployed on the platform can be accessed via a wide range of end-user devices (e.g. web browser, smart phone). As only the providers have direct access to the underlying infrastructure, they generally provide an API (application programming interface) that allows programmers to interact with the underlying infrastructure.

Within the context of PaaS, the most common charging models are also pay-per-use models. However, they can be implemented in different ways. Some PaaS providers measure the used resources of applications that users have deployed onto their cloud.

Resources under this point of view are e.g. frontend instance hours, bandwidth, stored data, etc . The more resources an application has used during a certain billing period, the more the customer is charged. An example for a provider with such a charging model is Google with its Google App Engine [58]. Other providers charge their customers a monthly fixed sum per user (e.g. Salesforce1 Platform [107]). Another possible charging model is used by Windows Azure [83] where customers have to pay for every hour a virtual machine with their application is running (which is very similar to the charging models used within IaaS) [31].

3.3.3 Software as a Service (SaaS)

The SaaS service model deals with the provision of applications over a network (mostly the internet). Consequently, users do not have to install the software locally on their respective device. In this context, applications are completely created, tested, and maintained by the service provider. That means, users have less effort to use an application that is provided over a network, but in turn they depend on the efforts of the provider to keep the application running and up to date. In addition, users do not have to manage, control or maintain the underlying infrastructure because these applications run on the hardware of the provider. This leads to the fact that users do not own the software, but rather borrow or rent it from a provider.

The kind of the application that is provided over the network is not restricted to certain properties. Almost every application can be used over the network. The most common way to access an SaaS service is by using a web browser. In addition, SaaS services are generally not restricted to designated end-user devices. Many services can be consumed via a great variety of devices (e.g. personal computer, smart phone, tablet, …). There are SaaS offerings that are free to use, but SaaS products exist where users are required to pay fees to use them. Those fees are commonly calculated on a monthly basis and are based on the number of employees of a respective customer that use the respective application [80, 132].

Examples for SaaS services are mail applications (e.g. Google's Gmail [57]), social network applications (e.g. Facebook [47]) or complex business applications like Sales Cloud from Salesforce.com [105].

3.3.4 Human as a Service (HuaaS)

Human as a Service has a special status in this thesis since it is not mentioned in the cloud architecture model in chapter 3.2. Still, it is presented in this context because it is a great enrichment of the previous model. The term HuaaS goes beyond the physical hardware of various data centers or computers and illuminates humans as "tools" that offer workforce over a network.

Lenk et al. developed a model that is at its core related to the models presented in Figure 1 and in Figure 2. Their model illustrates also the three service model IaaS, PaaS, SaaS, but it adds the HuaaS layer on top [76].

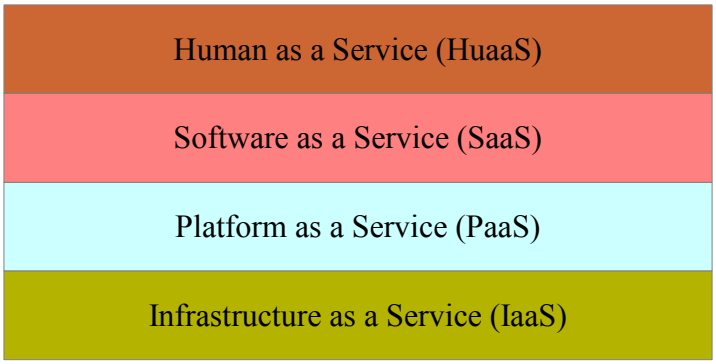

Figure 3: Service Models with HuaaS cf. [76]

Lenk et al. emphasize the need for this layer with the following statement.

"Some services rely on massive-scale aggregation and extraction of information from crowds of people. " [76]

This means that some services need the intelligence of humans to work properly. Thereby it is not necessary to predefine the tools humans have to use to solve certain tasks. Every individual in the crowd can decide for her- or himself which tool to use. Providers that utilize the HuaaS layer are for example Crowd Sourcing Platforms. A popular crowd sourcing platform is Amazon Mechanical Turk [2] where customers can define tasks that need to be solved. These tasks are generally hard to automate (e.g. analyzing pictures, writing reviews). Therefore, human intelligence is needed. These tasks are solved by other users of the platform who in turn receive adequate monetary remuneration. Another example of an HuaaS service is Iowa Electronic Markets [120] where the acting of humans is analyzed and aggregated to predict future events (mainly outcomes of political elections) [76].

3.3.5 Interaction of Service Models

As shortly mentioned in chapter 3.2.3, the arrangement of the service models in the cloud computing architecture is not arbitrary. The particular service models can built upon each other which leads to this specific layered architecture. Upper layers can utilize lower layers to offer respective services. The following paragraph illustrates an example of how a full interaction of all service models could work.

A company wants to host a crowd sourcing application called "**MyCrowdSourcingApplication**" on the internet. It decides that the application should be hosted in form of a website, but the company does not want to host this website in its own data center. The company rather wants to run it in a public cloud environment. Therefore, it rents virtual machines via **Amazon Elastic Compute Cloud** where it wants to host the website. Then the developers of the company create a platform which they call "**MyCompanyPlatform**" that they want to utilize when running the crowd sourcing website. They deploy this platform on the virtual machines rented from Amazon Elastic Compute Cloud and thereby built the company's own PaaS service. After that, the developers program the actual crowd sourcing website and deploy it via the previously created PaaS service. Then, the company makes the website available to the internet community. The website itself (i.e. the application) can now be considered as an SaaS service. **Users that utilize the website** (i.e. the crowd) and solve tasks constitute the top and last layer of the model, the HuaaS layer. This example is illustrated with the respective terms in Figure 4.

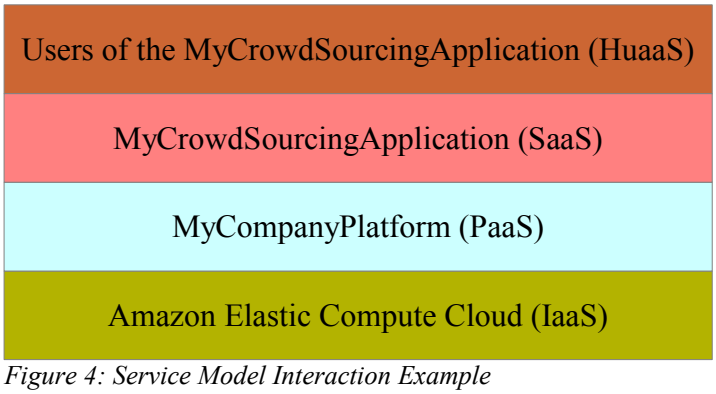

Figure 4: Service Model Interaction Example

3.4 Deployment Models

As stated in the definition of cloud computing in chapter 3.1, cloud computing as such is only a model. The term defines certain characteristics and key points, but does not cover actual implementation methods. However, specific deployment models of cloud environments have emerged. Four specific deployment models are considered in the NIST definition of cloud computing [80]. Moreover, they are also heavily used by other researchers and in practice [33, 66, 81, 133].

The following sub-chapters describe each of the four deployment models in detail.

3.4.1 Public Cloud

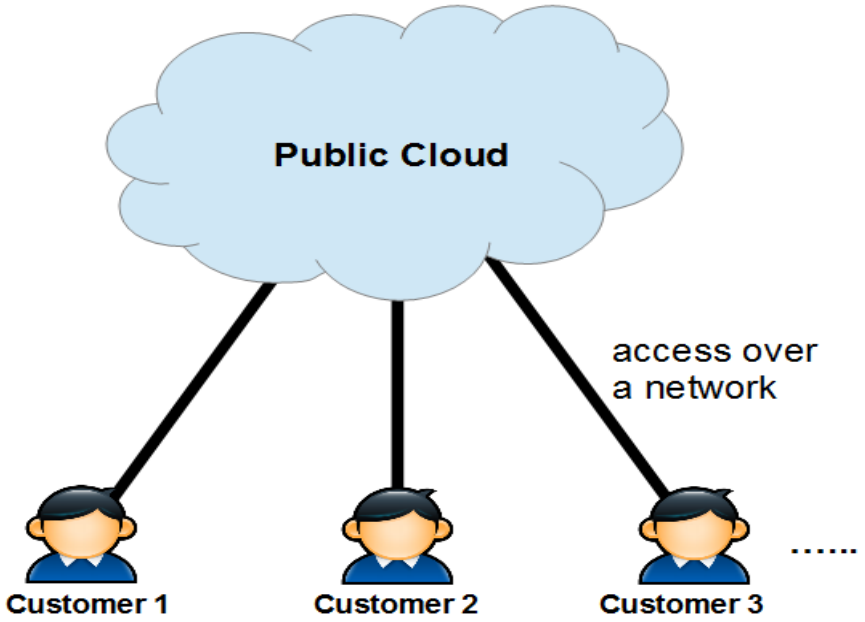

Figure 5: Public Cloud Deployment Model

A public cloud is hosted and operated by a cloud provider that offers cloud services to the general public. This means that customers do not have to purchase their own hardware or software. These services can be either offered free of charge or customers have to pay fees according to a respective billing model. Furthermore, customers consume these services over a network (e.g. the internet). This also leads to the fact that consumers do not have control over the underlying infrastructure and the specific processing of their data. Consequently, they have to rely on the cloud provider. That fact

makes it very difficult for organizations that deal with sensitive data to use such public cloud offerings. In the context of deployment models, the kinds of services a public cloud provider offers does not matter. It can be every service that fits into the model of cloud computing. The possible number of customers is specified by the public cloud provider. In general it is only limited by the amount of resources a service provider can offer. However, usually a virtually unlimited number of customers can access the services of a public cloud provider [66, 80, 81, 133].

All examples of cloud services mentioned in the "Service Models" section of this thesis are in fact offered by public cloud providers (e.g. Amazon Web Services that offers Amazon Elastic Compute Cloud [3], Salesforce.com that offers Sales Cloud [103]).

3.4.2 Private Cloud

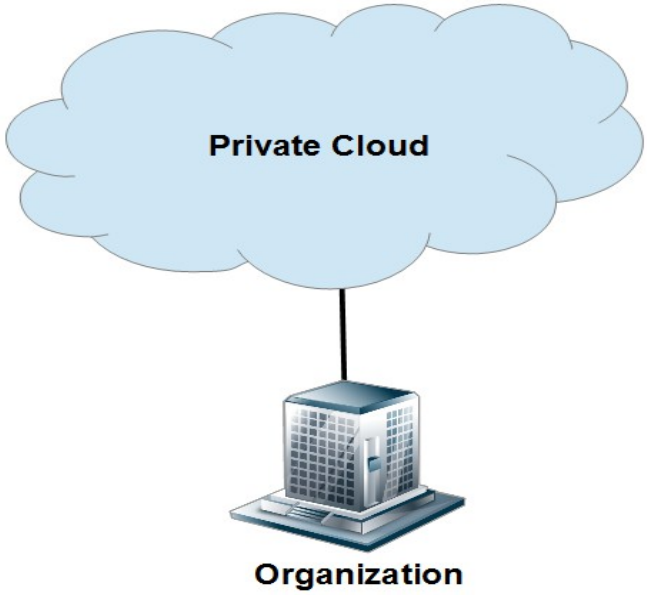

Figure 6: Private Cloud Deployment Model

A private cloud is a cloud infrastructure that is provisioned for exclusive use by a single organization. However, the cloud infrastructure itself can be run by the using organization, or by a third party provider. Therefore, it can be operated on or off premises. Nevertheless, it may be only used by one specific organization. Possible third party providers are just responsible for management and maintenance tasks [80].

A private cloud environment makes very fine grained configurations regarding performance, reliability and security possible. The owning organization has full control over the aspects of the cloud. It can solely decide which software to run inside the pri-

vate cloud and how to use it [133].

This private cloud topic raises the question which differences exist between such a private cloud environment and traditional data centers or server farms. One could argue that a private cloud is the same as data centers that leverage virtualization technologies (so called virtual data centers). The answer to this question is that although virtualization technologies are used in virtual data centers, support and intervention of the IT staff is still needed. On the contrary, a private cloud uses technologies that enable the essential cloud characteristics illustrated in chapter 3.1. Therefore, users can benefit from features like on-demand self-service and scalability. They can demand resources that are automatically provisioned to them by themselves without the interaction of the IT staff. In short, a private cloud is a virtual data center that enables the essential characteristics of cloud computing [27, 133].

3.4.3 Hybrid Cloud

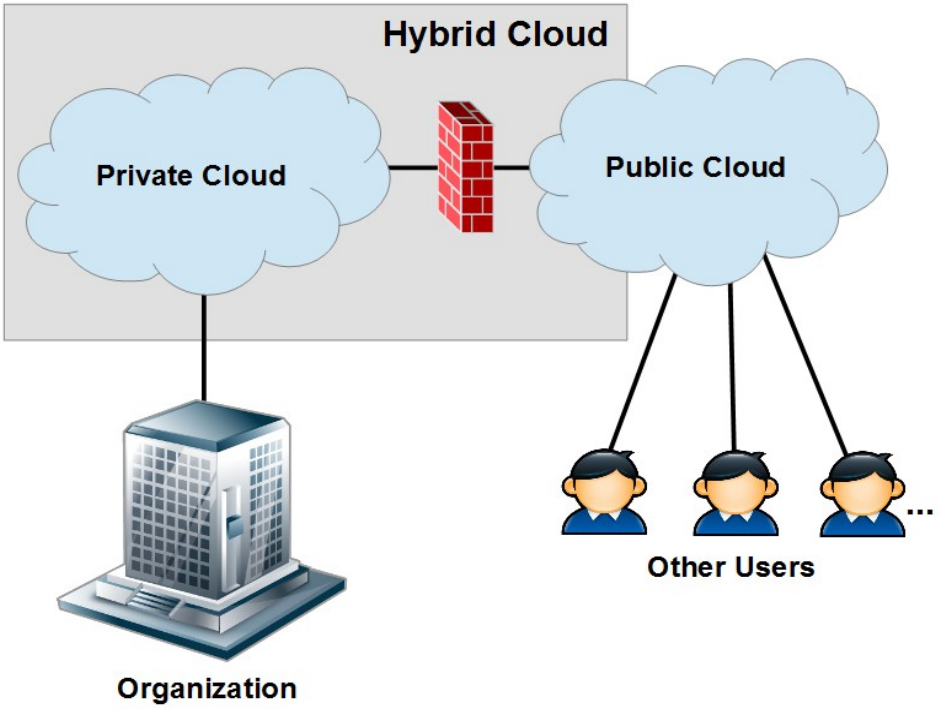

Figure 7: Hybrid Cloud Deployment Model cf. [86]

A hybrid cloud combines the advantages of a private and a public cloud. The hybrid cloud deployment model consists of two different clouds that still remain unique entities: a private cloud that is completely under the control of a certain organization and a public cloud that can be used to outsource heavy workloads. These clouds are connect-

ed via standardized or proprietary technologies. The hybrid cloud deployment model tries to alleviate the concerns of public clouds (e.g. processing of sensitive data) by being able to store and process such sensitive data in a private cloud and enabling it to outsource uncritical workloads to a public cloud. Nevertheless, in this model it is important to consider the best split between the public and private cloud components [80, 133].

Keeping that in mind, the possibility to "offload" uncritical workloads into a public cloud is a great advantage. This task is also called "cloud bursting" and describes the process of "bursting" data into a public cloud, e.g. when a company has to deal with peak loads. In fact, it can be used for load balancing purposes between the clouds. It outlines one major advantage of a hybrid cloud model because customers only have to pay for the extra resources (from the public cloud) when they are needed. However, cloud bursting is only recommended for non-critical applications that do not deal with sensitive data. In addition, it is important that applications that are burst into the public cloud are not strongly connected to other applications of the company so that no problems arise due to the bursting process. Consequently, cloud bursting is best used for applications that are mostly independent and are not integrated with other applications, components or systems [80, 85, 99].

Another great use case of hybrid clouds is hybrid cloud storage. In this scenario, the public part of the hybrid cloud is primarily used for archiving and backup purposes. Non-sensitive data can be easily outsourced into the public cloud to preserve the storage capacities of the private cloud. The data that is backed-up in the public cloud are typically file-level or image-based replication snapshots that only exist for a defined time period. After that period, the backups are deleted and replaced with the latest versions of the respective data. Additionally, the hybrid cloud storage approach can also be combined with cloud bursting by using the public cloud as fill area where storage capacity is borrowed on a temporary basis to deal with peak loads and short projects. After the peak load or the short project, the data that was burst into the cloud can be deleted in most cases [85].

3.4.4 Community Cloud

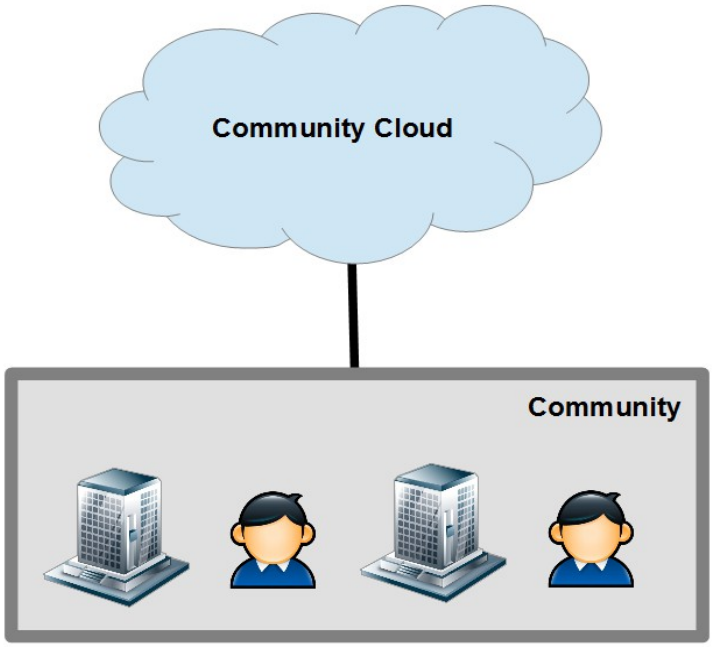

Figure 8: Community Cloud Deployment Model

A community cloud is provided for the use from a community. A community in this context are organizations or different entities that share certain concerns (e.g. missions, objectives, audit requirements, performance requirements, security requirements). The cloud infrastructure itself may be owned and managed by one of the members of the community, all of the members of the community, or a third party service provider. Furthermore, it is possible that a community cloud is operated on or off premises [70, 80, 101].

The aim of a community cloud is to enhance organizations that have joint goals that can be additionally supported by a common infrastructure. A community cloud delivers the benefits of a public cloud (e.g. rapid on-demand provisioning of resources, scalability) with an additional layer of security and privacy by having only certain members in a respective community. The members of the community only have to rely on the other members of it instead of relying on an external provider (which would be the case with public clouds). Thereby, the community itself can decide how to handle the various aspects of the cloud and can determine how to process sensitive data [101].

An example of a community cloud could be an infrastructure for state and local

government organizations within a country where all the organizations can access the common data from the same community cloud infrastructure. In turn, they are also able to share their own information via the community cloud. In this case, the members of the community are all the state and local government organizations. They all share the same infrastructure and the community itself can decide how to deal with the various aspects of the data processing procedure [65].

3.4.5 Choosing a Cloud Deployment Model

The previous four chapters are designed to give an overview about the different types of cloud deployment models. If a company wants to start using cloud services, it has to decide which cloud deployment model fits best for it. This decision generally depends on the business scenario of the company. It has to select the deployment model that is the most appropriate for a respective use case. Compute-intensive scientific applications that do not necessarily deal with sensitive data could fit extremely well inside a public cloud environment because of cost effectiveness. Applications that handle sensitive data that must not leave the premise of the company but still want to leverage the cloud model could fit best within a private cloud scenario. However, these two examples could also fit within a hybrid cloud scenario [133].

One sees that the decision for a cloud deployment model is not always clear and easy. One has to consider the up- and downsides of every cloud scenario and also evaluate the costs.

Practice shows that some deployment models are more popular than others. A survey by Taylor and Ericson [115] pointed out that the most common approach of cloud computing are internal private cloud deployments followed by managed vendor clouds (i.e. external private cloud deployments). A reasons for the result of this survey is that a private cloud approach deals best with the concerns of companies regarding security, stability and reliability. As owners of the cloud, companies are able to control the processing inside the cloud by themselves [36].

However, Gartner Inc. [50], a technology and research company, stated in a recent special report that there will be a shift from private clouds to hybrid clouds. They predict, that *"nearly half of large enterprises will have hybrid cloud deployments by the end of 2017"* [53]. Gartner argues that the private cloud deployment model reached maturity and became a tentative reality. Now the hybrid cloud model is in the same

spot as the private cloud model was at the beginning and the aspirations for the hybrid cloud model are high [53].

3.5 Cloud Management Platforms

A cloud is built by the interplay of various technologies and the underlying hardware. As mentioned in chapter 3.2, the architecture of the cloud model is rather complex and it is necessary to orchestrate the various elements of this architecture to build a working cloud environment. There has to be a control entity or platform that handles that interplay.

There are different tools used in practice that fill the role of such a control entity. However, the term for such a control entity is not standardized and different terminologies are used in this context. Such control entities are referred to as cloud management platforms [51], cloud orchestration platforms [18] cloud operation systems [88], or just as software frameworks [87]. The reason for the different terms is that there has not been a standardization of this control entity yet. In this thesis, the term "cloud management platform" is used to refer to this control entity.

3.5.1 What are Cloud Management Platforms?

Cloud management platforms are tools that provide a management layer for public, private and hybrid cloud environments. A cloud management platform manages the hardware and the virtualization hypervisors that run on top of it. In essence, a cloud management platform delivers an IaaS solution to its users. The cloud management platform itself is mostly an application with a web based user interface to deal with the various tasks of cloud computing (e.g. provision of virtual machines, metering, billing, provision of storage and network resources). Users can log into this platform and can request various resources (e.g. virtual machines) with a few clicks. The cloud management platform handles that request and sends appropriate requests to the respective hypervisors. That punctuates the on-demand self-service characteristic of cloud computing. The resources can be provided at any time a user requests it (provided that there are enough free resources within the underlying infrastructure). In fact, the user interfaces of various IaaS providers (e.g. Amazon Elastic Compute Cloud) are cloud management platforms [18, 51].

This taken into consideration, one sees that a cloud management platform is the

connection between the resource abstraction layer and the service layer within the cloud architecture model shown in chapter 3.2. More precisely, it even covers the IaaS layer since it delivers the respective services to its users. However, it is important to mention that the layers above the IaaS layer are not covered by cloud platforms. Nevertheless, users that want to provide services that correspond to layers above IaaS can use such cloud management platforms to deploy applications, e.g. by running their own SaaS service atop of it [87].

3.5.2 Usage of Cloud Management Platforms

As stated in the chapter before, cloud management platforms can be used to create public, private and hybrid cloud infrastructures. That means, such platforms are not only used by large public cloud providers that offer their services to various customers. There are also software products on the market that allow users to build their own cloud. Examples for such cloud management platform products are OpenStack [88], Apache CloudStack [17], and Eucalyptus [43, 87].

It is an interesting fact that all these mentioned cloud management platforms are under an open source and/or free software license. That means, users are allowed (amongst other things) to download and run the respective software free of charge. This fact shows how important the role of free and open source software nowadays is and that many great developments were made using open source licenses. OpenStack states on its website that the open source model is strongly needed in the context of cloud computing to foster cloud standards and to remove the fear of provider lock-ins [88]. There are big communities behind these software products that also include members of well-known companies. This way, the software can be steadily improved and enhanced [19, 45, 89]. In addition, companies that promote or distribute such software products usually also earn money by offering consulting, support or training services [44].

There are also many cloud platform products on the market that are based upon free or open source cloud platforms, but charge money for additional features or support. An example for such a software product is Red Hat Enterprise Linux OpenStack Platform [95]. The Red Hat Enterprise Linux OpenStack Platform is based on OpenStack but offers additional advanced features like improved virtualization and additional network features.

4 Quality of Service

In the context of cloud computing, quality of service represents how "well" a cloud provider delivers its services to customers. This taken into consideration, the main focus of this topic lies on public clouds (and to a certain degree hybrid clouds), because within these deployment models a provider takes charge of delivering services (at least partly in the case of hybrid clouds) and customers have to rely on them.

It is important for customers of cloud service providers that the services offered are delivered in a certain quality (e.g. a website must have a certain uptime and loading speed). For example, a web retail company could host its web shop within the cloud infrastructure of an IaaS provider. Therefore, the web retail company has to trust the provider that the servers have a maximum uptime and that customers of the web shop can access it at nearly any time. In addition, the retail company expects the provider to guarantee a certain bandwidth so customers can access the website fast and without variations in speed. The mentioned quality of service criteria in this example are availability and stability.

Quality of service criteria are of the highest importance for companies, especially when they are running their productive systems inside public cloud environments. Downtimes or outages can lead to massive costs for large enterprises [46].

There are certain quality of service criteria that are found frequently in the context of cloud computing. These are mentioned in the next chapter. Providers have to be able to guarantee these quality of service criteria if they want to acquire new customers. In addition, they have to provide a legal basis that defines remedies for customers if these criteria are not met. A way to guarantee certain quality of service criteria is that the cloud provider commits itself to a so called service level agreement. A service level agreement can either be provided solely by the provider or it can be negotiated between the provider and the customer. It states various quality of service criteria and other obligations of the provider and the customer. Moreover, the negotiation process between the service provider and the customer is also very important, because different customers can have different quality of service requirements. It is necessary that providers recognize this fact, and offer their services accordingly. Chapter 4.2 deals with the topic of service level agreements in greater detail [30].

4.1 Criteria

There are certain quality of service criteria that are virtually tailored to cloud computing. This chapter presents many criteria that are common and gives a short description of each one of them. The reason for the choice of the following criteria is that they can be found in various papers and articles. Additionally, they are also used in the ranking frameworks that were developed by Saravanan et al. and Garg et al. to rank different cloud services according to their ability to meet the user's quality of service requirements [49, 109].

These frameworks are a great basis for describing quality of service criteria because they deal exactly with the fact that quality of service requirements are different depending on the business case (as mentioned in chapter 4). They reflect many criteria that can be a factor within a customers decision for a respective cloud offering. Therefore, they also display different views of cloud computing services and illustrate several criteria from that different views. Garg et al. also empathize that it is difficult to evaluate cloud providers just by taking the obvious criteria like quality, reliability and security into consideration. They mention that customers see trade-offs when comparing different cloud providers regarding functional and non-functional requirements. Therefore, it is necessary to analyze also further criteria to evaluate which cloud offering fits best for a respective customer [49].

This short explanation of cloud ranking frameworks is to display why it is necessary to consider more then a few obvious criteria when looking at the quality of service standards of a cloud offering. It is important to empathize the relevance of the different criteria.

The following presents an extract of the quality of service criteria (or metrics/quality of service attributes, as they are called in the respective papers) presented in the works of Saravanan et al. [109] and Garg et al. [49] that fit best into the structure of this thesis.

- Costs

  Costs are a criterion that is usually not common in the context of quality of service. However, talking about public cloud services where many different pricing models are offered that mostly have a dynamical basis, it is a criterion that is crucial and has to be taken into account [49]. Moreover, one objective of

customers moving into the cloud is minimizing hardware and infrastructure costs [109].

In addition, the cost of a cloud service can be further divided into two separate parts [109]:

- On-Demand Costs

  These describe costs that are incurred when customers request resources without prior reservation. An example of this type of costs are on-demand instances of Amazon Elastic Compute Cloud where customers simply create and start virtual machines without long-term commitments or upfront payments. They only have to pay an hourly fee [4].

- Reservation Costs

  Cloud providers can offer customers to reserve resources by placing an upfront investment. In turn, further costs are reduced (e.g. the hourly fee). This has to be taken into consideration because it can greatly modify the overall costs. Reserved instances of Amazon Elastic Compute Cloud are an example of the utilization of such reservation costs. Customers can place an upfront payment to reserve an instance (i.e. virtual machines with custom configurations) for a specified time and have to pay significantly lower hourly fees [4].

- Service Response Time

  The service response time criterion is part of the performance of a certain cloud provider and deals with the speed of providing a certain service. This time should be at a minimum and can be measured by using the time between a user's request and the response of the service [109]. In the context of an IaaS service, the service response time is the time between the user requesting a virtual machine and the provision of a fully functional (i.e. booted and ready for use) virtual machine by the provider [49].

- Accuracy

  Accuracy deals with promised performance values of the provider. It measures

how often a cloud provider deviated from promised values. Performance values in the cloud computing context are values regarding compute units, network, and storage (e.g. CPU performance, network bandwidth, number of read/write operations per minute). The promised values of the provider are stated in the service level agreement that was either handed out by the provider or negotiated with the customer. Therefore, the accuracy can also be defined as frequency of failures regarding a provided service level agreement [49].

- Interoperability

  Interoperability is the possibility that services of the same provider or of different providers can interact with each other [49, 109]. This factor is crucial because a lack of interoperability can lead to lock-ins which restrict customers to designated services of certain providers.

- Portability

  Portability deals with the ability to move or migrate applications and data from one cloud provider to another, or between public and private cloud environments. As a quality of service criterion, this should be able without integration issues or disruptions. The need for portability can emerge because of various reasons, e.g. customers want to move to another provider because of a price increase or poorly offered service. Another possibility could be that a company wants to run its services in a location that is geographically closer to its customers and therefore wants to change providers. As with interoperability, without a certain degree of portability customers have to face possible cloud provider lock-ins [69, 100, 109, 127].

  An example of a use case where portability is important is the switching of a PaaS provider of a customer. The customer could have deployed various applications on the PaaS infrastructure of the provider. However, because of various reasons the customer wants to switch the PaaS provider. Therefore, the customer has to transfer the applications (or the source-code of the applications) to the new provider. The perfect situation would be that the customer does not have to do anything with the code and is able to deploy it without further ad-

justments on the PaaS environment of the new provider. However, this is not always possible in today's practice because of proprietary APIs and missing standards [116].

- Availability

  A customer should be able to access a service when needed [49, 109]. Therefore, availability deals with the uptime of a service. This criterion is highly important when hosting a productive system on the provider's infrastructure since failures of such systems can lead to massive losses as stated in chapter 4. Public cloud providers mostly try to guarantee an uptime of 99.95% during a monthly billing period [15, 61], but outages occur [129].

- Reliability

  Reliability is a criterion that indicates how well a service operates without failures. Failures in this context are not complete outages of a service, but deal with disruptions and hardware or software errors during operation (e.g. failure of a storage device) [49, 109].

- Scalability

  Scalability defines how well a system can scale up and down. Consequently, it is also a criterion if a service can handle a large number of simultaneous requests by scaling the necessary resources fast enough. This also reflects one essential characteristic of cloud computing mentioned in chapter 3.1: rapid elasticity [49].

  Moreover, there are two different forms of scalability:

  - Horizontal Scalability

    Horizontal scalability describes the act of increasing the performance of a system by launching new resources (e.g. virtual machines) and connecting these resources to the existing ones using technologies like load balancing or clustering [97].

  - Vertical Scalability

    Vertical Scalability deals with the ability to increase the performance of ex-

isting services (i.e. virtual machines in this context) by adding additional resources like processing units, memory, or network bandwidth [98].

Scalability is important when dealing with an unpredictable number of accesses and peak loads. As mentioned in chapter 3.1, providers often offer auto-scaling features that enables customers to define criteria (e.g. number of accesses per minute) to automatically scale the respective infrastructure up and down when certain thresholds are reached [7, 60]

- Security

  Security is an important issue for many companies, especially when dealing with sensitive data. When hosting data or applications within a public cloud environment, the customer has to trust the provider that it handles data safely and responsibly. Still, it is not possible for customers to control the actual data processing or look further into the provider's processes. As a result, security policies have to be in place and must be enforced. Therefore, security is a criterion that is highly important. Saravanan et al. divide security in the context of cloud computing further into three parts [109]:

  - Confidentiality

    Confidentiality deals with the protection of data stored within cloud environments against unauthorized or unintended access [109].

  - Data integrity

    Data integrity ensures that data stored within cloud environments can only be accessed and altered by authorized users [109].

  - Privacy

    Privacy describes that customers of cloud providers can (to a certain extent) act freely in a technical context and are not observed or disturbed by third parties [109].

4.2 Service Level Agreement (SLA)

The providers of cloud services should deliver their services with regard to the presented criteria. In addition, customers need a legal basis to make sure that the provider

meets the respective service criteria. To form a legal basis, a contract is necessary. This contract in the context of cloud computing is a service level agreement (SLA).

An SLA is a document that defines the obligations of the cloud customer and the cloud provider. By using an SLA, the provider can assure the customer that certain criteria will be met [72]. Therefore, it is also the basis of the trust-relationship between the provider and the customer, since customers are not able to directly control the processes of the provider [34].

4.2.1 Types of Service Level Agreements

The Cloud Computing Use Case Discussion Group mentions that there are in fact two types of SLAs used in practice: off-the-shelf agreements and negotiated agreements [34]:

- Off-the-shelf agreements

  These types of SLAs are offered by most public cloud providers. The agreements are fixed and supplied by the provider. They are non-negotiable, so the customers have to accept them if they want to use the service. Therefore, customers with critical applications or data will certainly not use these offerings [15, 34, 61].

- Negotiated agreements

  Negotiated agreements are created via a negotiation process between the customer and the provider. Hereby, the customer is able to present his specific needs and is able to propose necessary quality of service criteria. However, providers that accept the usage of negotiated agreements will certainly charge higher prices [34].

4.2.2 Contents of Service Level Agreements

There a typical contents which should be included in an SLA that are mentioned by Kandukuri et al. [72] and the Cloud Computing Use Case Discussion Group [34]. These are presented in the following sub-chapters. In addition, some of the mentioned contents are also taken from various news magazines on the internet because the respective topics fit well into the overall thesis.

4.2.2.1 Definition of Services

The SLA should state the exact services that are offered by the provider and should also describe the way they can be consumed [72]. In addition, it should include a description of each service delivered [34].

4.2.2.2 Ownership of Data

In many cases it is important to define which entity exactly is the owner of the data stored within a cloud computing environment of a public cloud provider. This is because customers could store data within this environment that is copyrighted, trademarked, patented, or where the ownership plays a critical role in any other way [119, 121].

4.2.2.3 Performance Management

This section deals with the quality of service criteria mentioned in the previous chapter. Customers often have certain needs regarding such criteria, e.g. they require a specific level of scalability or availability. An SLA has to state the respective criterion and should express target-values or target-metrics that the provider has to deliver. Therefore, there must be a mean for every service to monitor, measure, and analyze the respective criteria [34, 72].

4.2.2.4 Problem Management

A section about the management of occurring problems should be included in the SLA. It should also state arrangements of the provider to prevent problems [72].

4.2.2.5 Customer Duties and Responsibilities

The provider of a cloud service is commonly not the only entity that has obligations. The SLA can state that also the customer has certain duties and responsibilities to support the service delivery process [72]. This section mostly deals with arrangements against failure on the customer's side, e.g. the customer is responsible for created virtual machines or applications deployed within a cloud environment. For example, the public cloud provider Amazon Web Services states on its website for its IaaS service Amazon Elastic Compute Cloud that security obligations are shared with the customer. That means Amazon Web Services guarantees for the security of the underlying infrastructure but the customers are responsible to properly secure their created virtual ma-

chines [12].

4.2.2.6 Warranties and Remedies

The SLA should present remedies for customers if the provider could not meet the terms stated in the SLA. For example, it should state which compensation a customer receives if the provider was not able to meet the respective availability of a service defined in the agreement. In addition, it should also mention exclusions of such remedies, e.g. because of an act of nature [34, 72].

4.2.2.7 Security

Security is especially important for cloud computing services, because the customer has to highly trust the provider. To ensure that the provider offers a secure environment for the customer, the SLA should state the specific infrastructure and the security practices of the provider. These statements can include the usage of encryption when storing data of a customer, a properly secured network by using up-to-date network resources (e.g. firewalls), physical security of the infrastructure by employing security personnel, and so on [121].

4.2.2.8 Location of Data

The location of the data, i.e. where the data is hosted, is very important in cloud computing environments because of legal aspects. When dealing with legal issues, the location of the data generally defines the laws that apply to the respective data. Usually, that are the laws of the country where the data resides. However, different countries have different requirements regarding the security and the processing of data [121]. In addition, if a customer stores data in more than one country, it is necessary that the customer is sure that the cloud provider also complies with foreign legal issues regarding the transfer of information [119].

Nevertheless, cloud computing is a service based on data and processing hardware. That means, provider often store redundant copies of data in more than one location to prevent the loss of data. That is also the reason why various cloud providers are not able to guarantee a definite location in their SLA. However, that act does certainly not always comply with the legal aspects described in the paragraph above. If data is stored in more than country, there are often different laws and requirements to consider

which can lead to legal issues [42].

4.2.2.9 Government Requests for Access to Data

An important topic in an SLA is also what happens, if there is a governmental request for a customer's data. It should specify when, and even if the customer is informed in the event of such an request, e.g. before or after the disclosure of the data. In addition, it should include the obligations of the provider if the customer decides to contest the governmental request [119, 121].

4.2.2.10 Disaster Recovery and Business Continuity

The SLA should include specific methods for cases of emergency on the providers side to recover lost data. This is important, because such cases can happen and the provider should be able to efficiently handle them. In addition, the customer has to be able to assume that the business of the provider and the services delivered by the provider will continue an acceptable amount of time [72].

Another form of continuity that could be mentioned in the SLA is the continuity of the SLA in its current form. It could be possible, that the SLA of the provider will be subject to change in future. Therefore, it is important to inform the customer if that can happen and to which extent [118].

4.2.2.11 Termination

Termination deals with the end of business between the provider and the customer. The SLA should state how a termination can occur and if there are certain obligations of the contracting parties if a termination happens (e.g. further payments) [72].

4.2.3 Service Level Objectives (SLOs)

Service level objectives are generally part of an SLA. They are commonly derived from the typical contents of a service level agreement mentioned in chapter 4.2, alongside with various quality of service criteria mentioned in chapter 4.1. Service level objectives define measurable performance indicators that have to reach a certain performance target. They define boundaries whether certain values of quality of service criteria are acceptable or not. Summarized, SLOs are the objectives that have to be fulfilled by the provider [34, 54, 130].

Examples of SLOs are [34]:

- the average availability of the virtual machines hosted by the provider must be at least 99.9999% per year
- the time it takes to create and run a new virtual machine must not exceed three minutes
- it must not take longer than 5 seconds to reply to a request from a client machine

There are also various factors that affect the SLOs that apply. Customer created applications that run within the cloud environment of the provider will require other handling than applications that are provided by the provider. There could also be differences between the interaction of applications and data that is hosted in cloud environments. An application that accesses data within the same cloud environment is likely to process this data faster than an application that requests data over the internet, because it can gather the data from the same infrastructure. In turn, an application that runs in one cloud environment, but accesses data that is stored in a cloud environment of another provider, will be slower because it has to request the data over a network. Such cases have to be considered when defining SLOs [34].

The SLOs are not just obligations for the provider that have to be fulfilled. They can also help the provider to plan the resources needed for a respective customer. Consequently, the provider can exactly adjust its resources according to its customers [130].

Nevertheless, the ratings of service providers are often based on their ability to perform according to such SLOs. Failing to perform according to SLOs can lead to consequences, including termination of the contract between customer and provider [130].

4.2.4 SLA Monitoring

The creation of an SLA is not the last step that has to be done to guarantee a certain level of quality of service for the customer. The criteria that have been defined in the SLA have to be monitored and measured so that violations of the SLA can be identified [34].

In addition, it is important to define if the customer automatically receives compensation if certain criteria of the SLA are not met, or if the customer has to report SLA violations to receive compensation. Indeed, this could be negotiated and defined in the SLA, but in today's practice customers generally have to report violations by them-

selves to receive remedies if the provider fails to fulfill the criteria of the SLA. Consequently, the customer also has to measure the respective criteria. Public cloud providers often offer interfaces to display some quality of service properties to their customers (e.g. via a website) in near-real-time [13, 106]. However, the customer has to trust these reports from the provider and has (without further interaction) no possibility to verify this information. In addition, it is possible that the measurements of the provider at its end and the events at the customer's end are different. Therefore, it is possible to involve a neutral third party to measure the performance and report the measurements to the customer and to the provider (e.g. Rackspace with its cloud monitoring product [91]) [35].

As mentioned in the paragraph above, monitoring the service performance is commonly a thing the customer should do too Therefore, there are mechanisms to monitor cloud services against SLA violations. These can be called SLA monitors and they are used to observe services at runtime to ensure, that the service fulfills the criteria defined in the SLA [21]. Such mechanisms are implemented within various software products that exactly monitor various performance aspects of a cloud service. Examples of such tools are Hazy Cloud [24], Cloud Status [64], and LogicMonitor [78].

4.2.5 Service Level Management

4.2.5.1 Service Level Management, Provider's View

The last chapter dealt greatly with the monitoring of cloud services and the focus lied mainly on the customer. However, cloud providers certainly have interest in properly fulfilling the SLA requirements. First, they want to have satisfied customers who want to further use the service offered by them. Second, they are probably reluctant to provide compensation for customers for failing to achieve the respective service level objectives stated in the SLA. Therefore, cloud providers of course also monitor and measure their service to see how well they comply with given SLAs.

Consequently, they gather and measure data about their performance to make decisions about their current infrastructure. This process is called “Service Level Management”. With service level management, providers can see how well they perform regarding the given SLA criteria and can make decisions based on the data gathered.

For example, while analyzing gathered data, a provider could realize that the time it

takes to supply a certain customer with a new virtual machine barely meets the customer's requirements stated in the SLA. The provider could now respond to this by assigning more physical hardware to the respective customer to speed up the process.

Consequently, the goal of service level management for providers is to take smart decisions regarding the SLAs of customers [34].

This service level management process can be further supported by so called “SLA management systems”. SLA management systems consist of various software products that provide features to enable or enhance service level management. These features deal with the processing of SLA information and include the collection, storage and management of such information. In addition, these systems commonly include monitoring mechanisms that can be used to monitor the performance of services for certain customers with regard to the defined SLA. Therefore, the system can acquire performance values of certain criteria and can check them against an SLA. If values do not or just barely meet the values stated in the SLA, the system can send notifications to the persons responsible [20].

4.2.5.2 Service Level Management, Customer's View

The term “Service Level Management” does not only exist at the provider’s side. Customers can also make use of service level management. However, the characteristic of the term is rather different from the customer's side. A customer uses service level management to make decisions about the best way to use cloud services offered by a provider. Criteria that play an important role in this decisions-making process are for example the performance of a service and the costs.

For example, a customer of an IaaS provider could realize that the CPU load of many created virtual machines is too high. Consequently, the customer could decide to increase the number of virtual machines and use load balancing technologies to evenly spread the CPU load. However, in turn the costs of the service could increase.

The process of analyzing the various possibilities to find out the best decision is also called service level management. It helps customers to make good decisions with regard to using a service [34].

4.2.6 SLA Frameworks

The SLA topic and the connected necessary steps are rather complex. The complete

SLA process ranges from the negotiation or provision of a respective SLA, over the monitoring and measurement of specific SLA criteria, to the reporting of violations of the SLA. Therefore, researchers tried to create models that illustrate this process and help to design and understand specific steps that can be realized in practice. These models are generally called SLA frameworks and have different approaches and goals.

Most research that has been done in this context does not include cloud computing, but greatly deals with the overall topic of SLAs. Two main specifications have emerged that deal with the SLA process of general web services. These are the Web Service Agreement (WS-Agreement) published by the Open Grid Forum (OGF) [16] and the Web Service Level Agreement language and framework (WSLA) from IBM [73]. In the context of cloud computing, researchers either decided to develop models that are based on these specifications or developed completely new models by arguing that models based on WS-Agreement or WSLA do not provide the appropriate structure that is needed for cloud services [1, 90].

The following two sub-chapters describe current SLA frameworks. The first presents a framework that is not based on a prior specification, the second model is based on WSLA.

4.2.6.1 SLA framework by Alhamad et al. [1]

Alhamad et al. [1] developed a conceptual framework that is not based on another specification. The authors decided that way because they mention that WS-Agreement and WLSA do not offer dynamic negotiation of criteria. They emphasize that such a framework has to have a specific structure, so that customers can design their own business rules with regard to the guarantees stated in the SLA.

The model is based on different metrics (i.e. the SLA criteria) that are proposed from the authors for each service model. These metrics define functions that describe how respective service criteria can be measured and state target-values that have to be achieved. Additionally, the authors also mention metrics for a model they call “Storage as a service” which describes the usage of storage capacity within a cloud and additional metrics for general SLA terms. Customers can use these metrics to negotiate appropriate SLAs with the provider. Examples of metrics which are suggested in the work of Alhamad et al. are [1]:

- SLA metrics for IaaS: CPU capacity, memory size, boot time, …

- SLA metrics for PaaS: integration, scalability, browsers, …
- SLA metrics for SaaS: reliability, usability, scalability, …
- SLA metrics for Storage as a service: geographical location, scalability, …
- SLA general terms: monitoring, billing, security, …

Additionally, the authors present different forms of the SLA negotiation process that can occur between the customer and the provider. They present three different scenarios of how the negotiation process could work out.

1. In the first scenario, the service provider offers a fixed SLA. Then the customer can accept it by signing it, the customer can start a re-negotiation process if there are some points that need to be improved, or the customer can terminate the whole process. This scenario is quite similar to the use of off-the-shelf agreements mentioned in chapter 4.2.1. The only difference is that there can be a re-negotiation step [1].

2. The second scenario includes a trusted agent that is consulted during the negotiation process. This agent should have experience in the fields of cloud computing and defining appropriate SLA criteria. In addition, a number of activities within the negotiation process can also be assigned to external agents. These activities include the collection of data of business processes and goals of the customer. This is done by monitoring necessary SLA criteria that are used in the negotiation process [1].

3. As mentioned above, the framework differentiates between the different cloud service models. The third scenario includes more than one agent. Every agent should be a specialist within at least one service model. The customer then can use consulting services from each agent for cloud services within the respective service model. The agents define the appropriate SLA criteria and complete the negotiation process. This scenario is most efficient if the customer needs services that belong to more than one service model [1].

4.2.6.2 SLA framework by Patel et al. [90]

The SLA framework by Patel et al. [90] is based on the WSLA specification. This specification consists of concepts and an XML language to create formal SLAs [73].

The authors emphasize the need of such a framework by highlighting that many present SLAs force the customer to report violations of the SLA by themselves. The customer has to recognize violations and have to notify the provider to enforce the SLA. The authors state that using a framework to create a formalized SLA allows the creation of an automated SLA violation reporting process [90].

The authors recognized that the quality of a service can change over time. Therefore, constant monitoring and measurement is necessary. However, they state that a simple "measure and trigger" operation is not suitable for cloud SLA enforcement because of the great variety in consumer demands. The monitoring and measurement process has to be customized.

In addition, the authors emphasize that the trust in the cloud service provider also has to be considered. As mentioned in the SLA monitoring section in chapter 4.2.4, customers may not completely trust the monitoring outcomes of the provider. Therefore, customers can make use of third party mediators. These mediators also monitor and measure various quality of service criteria and report the results and possible violations to the customer and the provider. This fact is also considered in the framework [90].

4.2.6.2.1 Background: WSLA [73]

To illustrate the complete framework it is necessary to describe some of the fundamental specifications of WSLA. As mentioned above, WSLA is a specification that consists primarily of formal XML documents and concepts [73]. It describes mainly three entities:

- Parties

  WSLA defines three different parties that are involved in the SLA process. These three parties are: service provider, service customer and third parties. Obviously the service provider is the provider of a respective service and the service customer is the customer of the respective service. A broad range of tasks can be assigned to third parties. Starting from the measurement of quality of service criteria, over the supervision of the guarantees stated in the SLA, to the reporting of possible violations of the SLA [73].

- SLA parameters

  In the context of WSLA, SLA parameters (i.e. quality of service criteria) are specified by metrics. Metrics define how to measure criteria (resource metric), or how to aggregate metrics to create more combined and meaningful information (e.g. transactions per hour, an aggregation of the transaction count and the uptime of a specific service). These aggregated metrics are called composite metrics. In addition, metrics also define which party has to do the measuring and aggregation of the respective SLA parameter and also defines how to retrieve it [73].

- Service Level Objectives (SLOs)

  The SLOs presented in the context of WSLA are very similar to the SLOs mentioned in chapter 4.2.3. However, in this context SLOs are enriched with a “if..then” structure and can therefore be defined as formal expressions. The “if” part contains conditions, e.g. “if the average availability of the virtual machines hosted by the provider is not at least 99.9999% per year, ...”. The “then” part defines an action guarantee that represents promises of parties to do something, e.g. “..., then the customer receives appropriate monetary compensation from the provider” [73].

It is important to remember that WSLA contains specifications and is build upon XML [73]. These specifications are used by Patel et al. to create a framework for SLA processes within the context of cloud computing [90].

4.2.6.2.2 Big picture of the framework [90]

Patel et al. [90] created an illustration of their framework. This illustration reflects the different aspects of the framework in a concise and clear way. The illustration is presented below, followed by descriptions of the different parts of the framework.

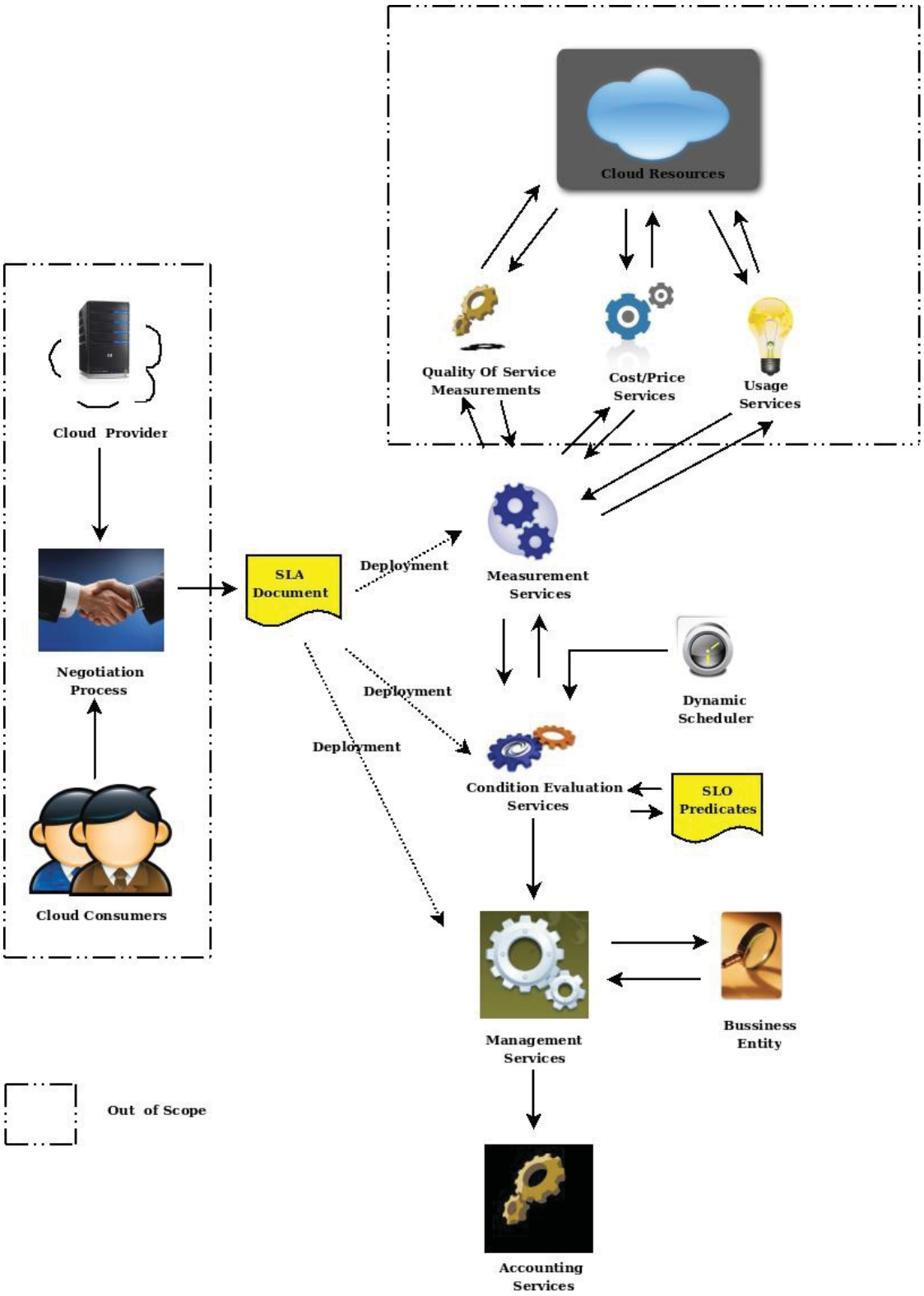

Figure 9: SLA Framework by Patel et al. [90]

It is important to mention that the authors assumed that the negotiation process has finished and that the SLA document has already been created when using this framework.

In addition, the SLA framework does not consider the way the respective cloud services are provisioned. Consequently, this framework deals only with monitoring and enforcement processes of SLAs. Therefore, the respective areas in the Figure 8 inside the dotted rectangles are considered out of scope of the framework [90].

The framework as such starts with the finished SLA document. This document has to be deployed to entities which offer certain services. Deployment in this context is a special term because it reflects a process stated in the WSLA specification. The deployment process includes the checking of the validity of the SLA and the distribution of the document fully or partly to the involved parties [73]. The reason why the SLA could also only be partly distributed to involved parties is because the provider and the customer may not want to hand the complete SLA to third parties. This could be because of security considerations for example [73, 90]. The receivers of the SLA then provide services to support the SLA process. These services are in fact typical WSLA processes but have been slightly adapted by the authors of the framework to fit into the cloud computing context [90]. The three mentioned services are:

- Measurement Services

  Measurement services measure the performance of cloud services at runtime. The authors also recognized that usage and costs of cloud services are also dynamic. Customers are easily able to request new resources and use them. In turn, the costs of the cloud resources are also increasing. Therefore, they included them inside the framework as usage and cost services and stated that they should be included in the set of measurement services [90].

- Condition Evaluation Services

  These services are used to check the data delivered by the measurement services against the service level objectives. If the condition evaluation services detect an SLA violation, they report it to the management services. The authors of the framework also realized that the evaluation action has to occur more frequently within cloud computing than with ordinary web services. They state that in cloud computing, SLA violations mostly occur when the load of the provider is high. Therefore, they added a dynamic scheduler that ensures that the evaluation checks are performed more frequently when the load of the

provider reaches specified thresholds [90].

- Management Services

  The management services are responsible for enforcing the SLA by receiving reports from the condition evaluation services. If they receive a report that states that an SLA violation occurred, the managements services are responsible for taking corrective actions. The specific execution of these corrective actions should be part of the SLA. The authors state that in the context of cloud computing these actions would commonly be financial compensations from the provider to the customer. However, practice showed that also the granting of credits which can be used to settle further payments is a way to reimburse the customers for not fulfilling the SLA [15, 61]. The process of handling the compensation-process is executed by accounting services that could also be delivered by a separate third party [90].

It is important to mention that the services described above can be provided by one or more different third parties (e.g. different companies). That means the customer could hire a company A that provides measurement services, a company B that provides condition evaluation services, and a company C that provides management and accounting services. It would also be possible that all these services are just provided by just one company. In short, the framework is just a formalized structure of how SLA processes can be handled in cloud environments and which "abstract" entities should take part in this process.

4.2.7 Current SLAs

This chapter is designed to give an overview of how SLAs are currently used in practice and how certain SLA processes are handled. It gives also a critical reflection of present SLAs and describes considerations in this context.
For this thesis, the SLAs of five different IaaS cloud products of well-known providers were analyzed and compared to give a concise overview of the current "SLA situation". The analyzed products are:

- Amazon Elastic Compute Cloud (EC2) by Amazon Web Services [15]
- Google App Engine by Google [61]

- Cloud Servers (Next Gen) by Rackspace [92]
- Cloud Services, Virtual Machines and Virtual Network of Windows Azure by Microsoft [82]
- vCloud Express by Terremark [117].

To enrich argumentation the work of Baset [108] was consulted. Baset developed an "anatomy" to compare SLAs and also presented present SLAs and made a conclusion [108].

All of the current SLAs from the mentioned cloud providers and cloud products are off-the-shelf SLAs (cf. chapter 4.2.1). That means, the SLA is presented by the respective service provider and the customers have to accept it if they intend to use the services of the provider. In essence, all of the mentioned SLAs are quite similar. This is reflected by four main points:

- Focus on availability

  All of the SLAs deal primarily only with the availability criterion. The service providers guarantee a certain uptime of the respective service per month (between 99.5% and 100%). Depending on the actual uptime percentage of the service, customers are eligible to receive compensation. However, the way the uptime percentage is calculated differs between the service providers. Within Amazon EC2, Google App Engine and the products of Windows Azure, the monthly uptime is simply calculated on a minute basis (total number of minutes of a month minus the number of minutes the service was not available divided by total number of minutes) [15, 61, 82]. The way the uptime of Cloud Servers (Next Gen) by Rackspace is calculated is more complex because it offers a much more itemized SLA. That means, there are separate categories for the different characteristics of the service, but in fact it comes down to a system that is based on the downtime of the different features of Cloud Servers (NextGen) [92]. Terremark indeed state that it guarantees a 100% uptime of vCloud Express, but it calculates on a basis of 15 minutes. That means as long as the service is not down for longer than 15 minutes a month, it is treated as if it had a monthly availability of 100% [117].

- Credits as compensation

  All of the mentioned providers grant credits as a form of compensation if they are not able to meet the uptime percentage stated in the respective SLA during a month. One credit has generally the value of one dollar. The amount of credits granted depends on the actual downtime of the respective service and differs among the providers. However, all providers (except Rackspace) state a certain cap that represents the maximum possible amount of credits granted per month. This cap is commonly defined as a percentage of the fees a customer has to pay in a given month (usually 10% to 50% of the service fees in the respective month) [15, 61, 82, 117]. As stated in the previous point, Cloud Servers (Next Gen) by Rackspace offers a much more itemized SLA that also allows the granting of credits up to 100% of the service fees in some cases [92]. The granted credits can be used to settle payments for the respective cloud service [15, 61, 82, 92, 117].

- Customer has to report SLA violations

  All SLAs of the mentioned cloud products leave the customer with the burden to report SLA violations (e.g. if the guaranteed uptime was not met). To claim credits, the customer has to inform the provider about the violation. The way the provider has to be informed depends on the respective provider. The SLA of Amazon EC2 states that customers have to contact Amazon Web Services by opening a “case” via a webform [15], customers of Google App Engine are just required to contact the technical support (an exact way is not stated) [61], Rackspace Cloud Servers (Next Gen) have to create a ticket and provide log-files of the downtime [92], the SLA of the products of Windows Azure states that customers have to contact the customer support [82], and customers using vCloud Express have to send an email to the vCloud Express support [117]. In addition, all SLAs state that the customer has to bring in such claims or requests within a certain time span (either within 30 days [61, 92, 117], until the end of the billing month after the incident occurred [82], or until the end of the second billing month after the incident occurred [15]). If a customer misses to submit a claim on time, the customer is not able to receive compensation at all.

What is more, all SLAs require the customer to provide evidence about the violation, e.g. by submitting log-files of the incident. That means the customer is in charge of exactly monitoring the uptime of the respective cloud service and has to create logfiles to report possible incidents.

- SLA Exclusions

  All of the SLAs present exclusions that define under which circumstances the SLA does not apply. The observation of every exclusion of each SLA would exceed the scope of this chapter. However, there is an important point that is mentioned in all of the SLAs. They all state that they do not apply when a downtime occurs with a cause that is outside of the reasonable control of the provider [15, 61, 82, 92, 117]. Examples of this statement mostly include force majeure and internet problems because of an error beyond the demarcation of the respective cloud service provider. However, it is hard to define the scope of "reasonable control" of the provider. The SLA of Cloud Servers (NextGen) by Rackspace mentions that denial of service attacks, virus activity and hacking attempts are also included in the "outside of reasonable control" term [92]. That means the risk of attacks by hackers is passed to the customers. In that case they would not receive compensation if the service fails because of such attacks. It is hard to tell if every service provider handles such incidents like that, but it shows that the term "outside of reasonable control" can be very broadly defined.

By comparing the different cloud SLAs, one recognizes that the providers only state availability guarantees in their respective SLAs. They do not provide statements and guarantees about the performance of the service. That means, customers do not get the assurance that (in case of IaaS services) custom-created virtual machines perform like defined during the creation process. Customers have to trust the provider that respective resources are assigned to their virtual machines. However, within business environments performance can be a very important criteria. As there are no guarantees about performance stated in the SLA, customers are not able to receive any compensation if the provider is not able to provide the needed performance [108].

Another critical point is the way customers receive compensation because of violations of the SLA. The SLAs of all the mentioned cloud products state that customers receive credits if certain guarantees are not met. The amount of credits granted generally depends on the actual uptime of a service in relation to a defined uptime percentage. As mentioned above, these credits can only be used to settle payments for the provider. That means, customers do not receive direct compensation for their losses because of a failure of the cloud service. They are not adequately reimbursed for financial damages they suffer because of downtimes. For example, if a customer runs productive services inside a public cloud environment of one of the providers mentioned above and the public cloud gets inaccessible because of errors of the provider, the customer has to accept the possible losses and only receives credits as compensation. Such a scenario is probably unacceptable for many enterprises.

It is also important to mention the representation of present SLAs. They are all in a textual form and differ in their structure. Consequently, the comparison of different SLAs is rather complex and takes much time. To achieve a better transparency between different SLAs, a standardization of the presentation of SLAs would be necessary. This standardization could lead to a structured representation of an SLA (e.g. with XML). A structured representation of an SLA would also enable it to automate the various SLA processes (cf. the cloud framework of Patel et al. in chapter 4.2.6.2) [90, 108].

5 Discussion

Cloud computing does not only have benefits. Therefore, it is necessary to discuss the advantages and disadvantages of cloud computing. Finally, this thesis is concluded by considering all the information of this thesis.

5.1 Advantages of Cloud Computing

This chapter deals with the various advantages of cloud computing. It presents key points that describe the possible potential of cloud services. In addition, there are examples for each point that illustrate the benefits in practice. Most of the presented advantages apply for all deployment and service models. However, some benefits can be rather matched to certain kinds of service or deployment models. For example, many benefits that deal with cost savings that result from renting resources from a provider

apply mainly to public cloud environments. In cases where such an association is necessary and informative it is pointed out.

5.1.1 Scalability

Besides that scalability is an essential characteristic in the context of cloud computing, it is also one of the major benefits of it. Resources can be delivered on-demand, but they can also be expanded on demand. Users of cloud computing can make use of this benefit if they want to deploy applications that will have to handle an unpredictable load. The provided resources are even able to scale according to pre-defined criteria (e.g. accesses, load). In addition, scalability can also support business growth. For example, even a small company can start with the usage of a cloud computing environment. As the company grows, the cloud environment can also scale with it [7, 28, 133].

Scalability can be implemented in many forms. As mentioned in chapter 4.1, there are two kinds of scalability: horizontal scalability and vertical scalability. The IaaS providers Amazon Web Services [3] and Rackspace [94] offer horizontal auto-scaling features for their respective IaaS products. That means, they enable the automatic launch of new virtual machines depending on pre-defined criteria or policies. However, it is always possible to launch new virtual machines and define the necessary performance criteria manually [7, 23].

The PaaS provider Google implemented automatic scalability for its product Google App Engine in a way that the customer does not have to specify anything to enable scaling. The automatic scaling is “built in” within Google App Engine. Therefore, the customer only has to create applications and deploy it on the Google App Engine cloud. Google App Engine automatically scales the required resources depending on the needs of the application. This process runs completely in the background and the customer does not notice it [60].

Customers of SaaS providers should not perceive the scaling processes either. As the SaaS model delivers the customer ready-to-use applications, she or he does not have to do anything regarding the underlying infrastructure. The provider is in charge of managing this infrastructure and has to implement models and processes that can deal with the needs of the customers. Therefore, the provider has to utilize scalable resources. This is generally done by using a cloud infrastructure in the background. However, it is also possible that an SaaS provider uses the services of other cloud

providers like Amazon Elastic Compute Cloud [110].

5.1.2 Dynamic Pricing Models

This chapter deals with the costs of cloud services and their pricing. However, it is necessary to mention that this chapter deals mainly with public cloud services because public cloud providers typically offer such pricing models. Within a private cloud environment the owner of it (i.e. the company who uses the environment) has to bear the entire costs for it.

Most public cloud providers (especially within IaaS and PaaS service models) offer highly dynamic pricing models. That means customers only have to pay for resources which they really use. These dynamic pricing models are also referred to as pay-as-you-go pricing models because that expression directly covers the economic benefit of the customer [22, 133].

For example, customers that use the IaaS service Amazon Elastic Compute Cloud [6] basically have to pay for each virtual machine they have running. The actual costs are calculated per hour a machine is running and they depend on the defined resources of the virtual machine. In addition, there are fees for data transfer and extra features [5].

The PaaS service Google App Engine has a similar pricing model. Users that deploy applications via Google App Engine have to pay for the resources these applications consume. There are different fees for various kinds of resources. For example, customers have to pay for every hour the front-end of an application is running, for the network traffic an application is generating (per GB transferred), and for certain numbers of API calls of the application (depending on the respective call, e.g. per 10k basic operations).[56].

Most SaaS providers also implement dynamic pricing models but in a slightly different way than IaaS or PaaS providers. To describe the way SaaS providers charge their customers the three SaaS providers Concur [38], Cornerstone OnDemand [40] and Salesforce.com [103] were observed. The SaaS products of the mentioned providers include a monthly fee for each user who accesses the SaaS product. In addition, the customer can select different editions of the software that differ in the features they offer. Editions that have more features come also with a higher fee. Considering these facts, it can be said that SaaS products also offer dynamic pricing models be-

cause the customers have to pay fees according to their selected feature set [39, 41, 104].

5.1.3 Prevention of Underutilization and Reduction of Operational Costs

Companies that offer services over the internet via their own infrastructure have to consider peak-loads. Such companies must have enough resources to deal with them, even if the average load is much lower. This is necessary because these companies do not want to loose customers because of unavailability or high response times. Therefore, many resources are in an idle state most of the time and this leads to high operational costs. With cloud computing (and especially with the usage of public or hybrid cloud services) companies do not have to provide the capacities to deal with peak-loads because the required resources to cope with them can be allocated and deallocated very quickly on demand. In such cases, companies only have to pay for the resources they really use, even during peak-loads. This benefit is accompanied by the benefit of scalability mentioned in chapter 5.1 because the rapid scalability enables this advantage. In practice, this benefit can be leveraged by completely running the service offered by the company within a public cloud or by running the service in a private cloud and utilizing additional resources from a public cloud for peak-loads (i.e. hybrid cloud model). In short, the possibility of requesting and renting resources from external cloud providers enables companies to reduce their operational costs by being able to efficiently use their own infrastructure and thereby preventing underutilization of it [22, 133].

5.1.4 Elimination of Upfront Investments

Companies that want to offer own web-based services do not necessarily have to invest in own infrastructure when they are using cloud services. They can rent resources from a cloud provider and have to pay for for their usage. This benefit also comes along with the dynamic pricing models mentioned in chapter 5.2. Therefore, customers do not have to buy own computing resources and other hardware. They do not have to maintain or manage the underlying infrastructure either [133].

This way of using resources in a cloud computing context is also often referred to as “CapEx to OpEx” which means that customers of cloud services convert capital expenditure (i.e. expenses to buy assets; in this context expenses to buy the respective in-

frastructure) into operational expenditure (i.e. ongoing costs for a running system, in this context the fees for the cloud service) [22].

5.1.5 Shifting the Risks

This benefit deals with the reduction of business risks by shifting them to the cloud provider (especially within public and hybrid cloud environments). In fact, there are two kinds of risks that can be mitigated with cloud computing.

1. Shifting the risk of infrastructure errors

   By using the services of a cloud provider, companies do not have to deal with errors that occur within the cloud infrastructure. The cloud provider is in charge of dealing with such errors. That can be beneficial because cloud providers generally have more expertise and have more specialists to deal with such risks. However, it is necessary to mention that this benefit can only be considered advantageous if the cloud provider is really better in dealing with such errors than the respective customer and if the customer is properly safeguarded in case of errors (e.g. with an SLA) [133].

2. Shifting the risk of misestimating workload

   This point comes along with chapter 5.1.3 because it also deals with over- or underutilization of resources. However, it is also a point in this chapter since the over- or underutilization illustrates a risk companies usually have to deal with. By using cloud services, this risk is shifted to the cloud provider. Customers can easily and quickly request new resources if they are needed. Since cloud providers have to deliver the requested resources, the risk is shifted to them. It has to be mentioned that cloud providers may offer different hourly-fees depending on if the resources have been reserved upfront (cf. chapter 4.1). Customers usually have to pay higher fees if the resources have not been reserved, this higher fee can be seen as a premium charged by the provider for assuming the risk [22].

5.1.6 Easy Access

The benefit of easy access reflects the fact that cloud services are usually web based. Therefore, they can be accessed with all kinds of devices that have web-access. For ex-

ample, web services can be accessed via computers or laptops, but also via smart phones and tablets. Cloud providers commonly offer a website that can be accessed via a web browser, but some providers also offer other ways to access the service. Amazon Web Services offers a stand-alone app for iOS and Android that enables it to comfortably supervise and manage different aspects of its services [9]. Summarized, this means that users of cloud services can access and manage them on a great variety of devices with a connection to the internet and nearly everywhere [133].

5.2 Disadvantages of Cloud Computing and Considerations

This chapter deals with the disadvantages of cloud computing. As stated in the approach of this thesis (cf. chapter 2.2), it is also important to identify the disadvantages of cloud computing to make use of it appropriately. The following sub-chapters describe these disadvantages and present characteristics that have to be considered when using cloud computing services. They also describe possible solutions and developments regarding these issues. As in chapter 5.1, some of the stated disadvantages apply only for a selection of service and deployment models. Therefore, in cases where disadvantages can be matched only to some service or deployment models, it is indicated accordingly.

5.2.1 Availability

Availability is a critical factor for many enterprises. If services are offered for productive environments, it is necessary that they are available and accessible when they are needed. This taken into consideration, this sub-chapter greatly deals with the availability of cloud services and especially with the availability of public cloud services that are consumed over the internet.

Customers using services from a public cloud provider depend on that provider. They have to trust the provider that the service is available when needed. However, customers are not able to directly control the processes of the provider and therefore cannot make sure that they are executed properly. As mentioned in chapter 4.2.7, providers try to guarantee a certain uptime that is stated in their SLA. Nevertheless, there are two things to take into consideration. First, the guaranteed uptimes stated in the SLAs are just goals of the provider. Outages can and do occur [129]. Second, the compensation for customers if the provider is not meeting the requirements stated in the SLA are

commonly just credits that can be used to settle further payments (cf. chapter 4.2.7). Therefore, it can be highly critical for an enterprise to trust in only one cloud service provider.

Armbrust et al. [22] state that even despite the bad publicity of outages of cloud providers, the infrastructure of many enterprises is not as good as the infrastructure of cloud service providers. Therefore, they mention that the only way to guarantee a high reliability of cloud services is to make use of multiple cloud providers. This leads to a situation where there is no single point of failure.

There are in fact many providers that allow customers to select different regions (e.g. Europe, USA, …) where they want to host their data and applications (e.g. Amazon EC2 [6], Rackspace Cloud Servers [94]). When using services from the mentioned providers, customers can select that they want to host their data and applications redundantly in more than one region [14, 93]. However, it is still possible that the provider has a common software-infrastructure or the same accounting system across these regions. An outage of this common infrastructure or the common accounting system could lead to an outage of the whole service. Consequently, the usage of more than one cloud provider is preferable [22].

It is necessary to mention that using the cloud services of more than one cloud provider redundantly leads to higher costs. In addition, it is also questionable if the different cloud providers offer a certain level of interoperability and portability. This would be necessary to make use of more than one provider comfortably.

What is more, the availability of a certain cloud service is not only dependent on the cloud provider. Another factor that has to be considered when talking about availability is the internet connection of the cloud customer (e.g. an small business company). As cloud services are generally consumed over the internet, the internet connection of the customer also becomes a key point of vulnerability. If the internet connection of the customer breaks down, the employees of the company are not able to consume the service. In addition, it is also necessary that the internet connection of the customer provides enough speed so that the employees of the respective company can use the service appropriately [79, 126].

5.2.2 Interoperability and Portability

Interoperability is the possibility to run services across more than one cloud provider.

Portability deals with the way of transporting data and applications from one service provider to another. Both of them can become important if customers want to switch cloud providers. Additionally, they are also mentioned in chapter 4.1 as an important quality of service criteria.

A lack of interoperability and portability can lead to provider lock-ins. That is because the customers of a cloud provider depend on the cloud provider and do not control its infrastructure like their own IT. In addition, most major cloud providers use proprietary data storage and APIs. That means switching the provider is commonly not easy. [22, 63].

This is a fact that has been known for a long time, therefore solutions are in development. A possible way to deal with the topics of interoperability and portability is to make use of open-source software instead of proprietary. The cloud platform "OpenStack" [88] is heavily evolving and is already deployed by cloud providers [26]. For example, the HP Public Cloud [62] and IBM SmartCloud [122] are based on OpenStack.

5.2.3 Security

Security is a big issue within cloud environments. Customers that want to store their data and host their applications inside cloud environments have to be sure that they are properly secured. However, in the context of public clouds enterprises do not have full control of the underlying infrastructure. The data of the customers is hosted in data centers all over the world, so they have to trust the provider that their data is handled appropriately. In addition, there are companies which deal with sensitive customer data, trade secrets, classified information and so on. Such companies are often not able to move into the cloud because many public cloud providers are not able to guarantee proper protection [63].

Most major cloud providers try to build up trust by presenting assurance programs or certificates to guarantee that their infrastructure is designed and managed according to standards and best practices (e.g. HIPAA, SOC, ISO 27001). For example, Amazon Web Services presents a website called "AWS Compliance" [8] where it presents information about assurance programs, reports and certifications [8]. However, there is still uncertainty and there are considerations that have to be taken into account.

There are security risks that can be matched to certain service models as well as se-

curity issues that deal with the structure of the whole cloud computing model.
When considering the three service models of IaaS, PaaS, SaaS and their structure (i.e. IaaS at the bottom, PaaS in the middle, and SaaS at the top), it is noticeable that if a service provider takes care of the lower parts of this model, the customer has to take care of the upper parts. For example, an IaaS provider offers the infrastructure to enable customers to create and run virtual machines. That means, the provider is in charge of securing the underlying infrastructure and the customer has to take care of the created virtual machines. The same concept can be used when comparing PaaS and SaaS. Therefore, the lower the service model is allocated, the more responsible becomes the customer with regard to implementing and managing security capabilities. However, that also means customers have less control over their data when using "upper" service models. Customers of SaaS providers often do not have visibility into the way their data is stored and secured. This leads to the fact that a classification of security issues according to service models makes sense [114].

The following three sub-chapters describe security issues within the service models SaaS, PaaS, and IaaS. These chapters deal especially with public cloud implementations of these models. The fourth chapter describes issues that particularly deal with the architecture of the cloud computing model, i.e. physical resources that use virtualization technologies to offer certain characteristics which are essential for cloud computing (cf. chapter 3.2).

5.2.3.1 Security issues in SaaS

In the context of SaaS the customer has to fully trust the provider regarding security measures because customers only use the provided application and do not know what happens in the background. What is more, as SaaS provider often have more than one customer, they have to make sure that different users do not see the data of each other. It is also possible that SaaS providers host their offered applications inside the cloud infrastructure of other third-party cloud providers (e.g. Amazon Web Services) which leads to the fact that it could be possible that the applications of the SaaS provider could be hosted on the same physical infrastructure as applications of other SaaS providers. Moreover, these third-party cloud providers could replicate the data in various other data centers to maintain availability. Consequently, an overlapping infrastructure is created and the transparency of the the data is further constrained. This fact

indicates a lack of control of customers over their own data and increases concerns about data breaches and applications vulnerabilities [114].

5.2.3.2 Security issues in PaaS

PaaS services offer customers to use the underlying infrastructure to deploy own applications. That means the customer is in charge of properly securing the deployed application and the provider has to secure the environment below this application level. In addition, it is necessary that the provider ensures that data from a certain application is inaccessible by another application. In essence, PaaS is more flexible then SaaS because customers can create and run own applications. However, that means the built-in security capabilities of the provider are also more flexible or less complete. Customers therefore have to create their own layers of security inside their applications [114].

In addition to not being able to access data form other applications, it can also be critical if PaaS applications can directly access the underlying infrastructure. This could lead to attacks and misuse of it. Therefore, it is necessary that applications are isolated from each other and the underlying infrastructure. However, this fact depends on the way applications are executed by the cloud provider within a PaaS environment. Comparing Amazon Elastic Bean Stalk (the PaaS product of Amazon Web Services) [10] and Google App Engine (the PaaS product of Google) [58], there are two different ways of achieving the isolation of different applications.

Google App Engine uses sandbox environments to secure deployed applications. Every application that is deployed onto Google App Engine runs in its own sandbox environment. Within such environments the access to the underlying operating system is restricted. This leads to more security and enables Google App Engine to operate faster [59].

Amazon Elastic Beanstalk is built upon Amazon EC2 instances. Every application that is deployed via Amazon Elastic Beanstalk runs on a single Amazon EC2 virtual machine. The deployed application has complete control over this underlying virtual machine. However, this virtual machine has per default no access to other virtual machines created within Amazon EC2. Consequently, it can also be considered as a secure environment [10, 11].

Nevertheless, customers cannot directly check the implementation of these secure environments and therefore have to trust the provider that they are implemented prop-

erly and securely.

5.2.3.3 Security Issues in IaaS

Within IaaS service models the customer has the highest degree of control over security compared to the other service models. Customers can usually define by themselves how they want to implement security mechanisms on their created virtual machines. However, to create a secure environment it is required that the underlying infrastructure (i.e. the hardware and the virtualization technologies) does not have any security holes [114].

As mentioned above, the IaaS service model offers the customer much flexibility also regarding security. Therefore, IaaS providers want to make sure that they are not liable if their customers do not take care of the security of their virtual machines. For example, Amazon Web Services defines in a security statement for Amazon EC2 that the responsibility to provide security is shared with the customer. As shortly mentioned in chapter 4.2.2.5, Amazon Web Services guarantees that it secures the underlying infrastructure up to the hypervisor. That means it takes care of security issues on the physical and the virtualization level. Customers of the service have to secure their created virtual machines and anything that they install on it. They are required to properly create user-accounts that can access these virtual machines and need to configure proper firewall rules for network packets that are sent to and from the virtual machines. [12, 114].

This leads to two considerations. First, it is necessary that customers of the IaaS service have enough expertise in managing and securing their virtual machines. Second, they still have to trust the IaaS provider that the virtual machines are hosted within a secure environment.

5.2.3.4 Security Issues based on the Cloud Computing Architecture

This chapter deals with the security issues that result from the architecture of cloud computing. It presents especially issues that exist because of the usage of virtualization-technologies. Therefore, it describes threats on the “Resource Abstraction and Control Layer” (cf. chapter 3.2).

As mentioned in chapter 3.2, hypervisors are used to create and manage virtual machines in the context of cloud computing. This leads to the essential characteristics of

cloud computing presented in chapter 3.1. However, these hypervisors can be the target of attacks that result in severe consequences for the hardware and the hosted virtual machines. Especially in the context of IaaS services where users are able to create their own virtual machines on top of a hypervisor, the hypervisor is particularly vulnerable to attacks. Attack scenarios are possible because providers of IaaS services are not aware of the contents of virtual machines of their customers. The providers are not able to detect the exact software that is running inside a user-created virtual machine. For example, an attacker could be a customer of an IaaS provider. In that case, the attacker would be able to create virtual machines on the infrastructure of the provider and would be "close" to the hypervisor. This leads to the fact that the virtual machines created by customers are entities that cannot simply be trusted by the provider. Keeping all that in mind, the direct victim of possible attacks in this context is the IaaS provider. Its hypervisors and hardware are possible targets for attacks. However, since other customers are also hosting their virtual machines within the environment of the IaaS provider, they can be harmed and attacked indirectly [67].

Ibrahim et al. [67] presented an overview of possible vulnerabilities and security threats regarding the cloud computing architecture. They divided them into three different categories which are described below.

- Hypervisor Attacks

  Hypervisor attacks try to directly harm the hypervisor. The big upside attackers see in compromising the hypervisor is that it enables them to gain control over the host system. That includes the virtual machines that run on it and possible applications installed on it [67].

  Some hypervisors also run one special virtual machine that has privileged rights across all running virtual machines. This special virtual machine is used for administering the other hosted virtual machines. Such a special virtual machine could also be a promising target for attackers. A scenario could be that an attacker successfully exploits possible weak spots of such a special virtual machine and manages to gain control over the hypervisor or other virtual machines [67].

- vSwitch Attacks

  A vSwitch is a virtual switch that runs along with the hypervisor and manages the traffic between the virtual network interfaces of the virtual machines and the physical network interfaces of the physical hardware. In addition, it also controls the traffic between the virtual machines hosted on the same physical hardware. Keeping that all in mind the vSwitch basically acts like a physical switch in non-virtualized environments. The main task of the switch is to forward frames on networking layer two and therefore it is necessary for network communication. This all makes the vSwitch to a potential target of a range of layer 2 networking attacks, very similar to a physical layer two switch [67].

- Virtual Machine Attacks

  A typical physical machine in a cloud computing environment can host many different virtual machines. These virtual machines can be direct targets of various attacks. Nearly every attack that affects a physical server is also able to affect a running virtual machine. However, if a virtual machine on a specific physical server is compromised due to an attack, the other virtual machines on the same physical machine are also vulnerable, even if they are not running. This is because they share the same hardware and the same hypervisor [67].

Research regarding these security threats is ongoing. However, there are still challenges that have to be solved. Basic approaches that try to deal with these issues are presented under the following bullet-points.

- Usage of traditional security solutions for virtual machine environments, i.e. firewalls, intrusion detection systems and intrusion prevention stems [67].
- Implementation of a security virtual machine that monitors other virtual machines from “outside”. That means it creates a view of the virtual machines on a hypervisor level [67].
- Usage of micro hypervisors that use specialized micro-kernels which are more difficult to exploit [67].
- Usage of special processors that secure the hypervisor against hypervisor-attacks [67].

These approaches are steps toward solutions for the security issues. However, the challenges that have to be solved include performance problems that result from the necessary monitoring of the virtual machines and the lack of information acquired from it. For example, the use of a security virtual machine just enables the monitoring of the virtual machines from a hypervisor perspective which includes memory-pages, disc-blocks and low level instructions. However, important information like processes, files, events, and system calls are not monitored [67].

5.2.4 Privacy

Privacy is an important topic in the context of cloud comping. This is mainly because customers of public cloud providers entrust their data to them. Prominent examples for such cases are webmail services, which are in fact SaaS offers. The emails are stored on the infrastructure of the provider where customers can access them. This means customers have to trust the provider that their e-mails are treated confidentially and securely. However, the customers cannot look inside the borders of the provider and cannot check the actual processing of them. Therefore, accidental or even deliberate disclosure can be possible. This could of course have severe consequences for customers [102].

There are also other ways providers can use data collected from customers. An SaaS provider that offers a CRM (customer relationship management) service could log updates of contact data that is done by its customers and make this data available to other users with the same business connection. This could be a very useful service offered by the provider, but in fact it would be a violation of the customer's privacy [29].

Therefore it is necessary to provide policies and legislation that ensure the appropriate usage of customer's data. However, policies and legislation are insufficient to fully make sure that cloud providers do not abuse data of customers. Even if policies are in place, misuse is still possible. Consequently, another layer of security would be beneficial. Current research deals with algorithms that enable the processing of encrypted data. The goal of this research is that customers are able to upload encrypted data onto the cloud infrastructure of the provider and the provider would still be able to perform computations and searches on the encrypted data without having to decrypt it. This assures that the provider is not able to read the data. Such techniques have already been shown possible, however, they are very expensive and resource-intensive. Therefore,

they show little practical potential. Nevertheless, research is going on and there are regular improvements [102].

5.2.5 Legal Aspects

Legal aspects are important with regard to cloud computing because of the structure of the cloud computing model. In the context of cloud computing, providers offer their resources to customers. These resources generally reside in data centers placed in different countries all over the world. Additionally, different countries also have different laws regarding information technologies. Consequently, the actual location of the data is important. However, cloud computing providers can keep redundant copies of data in more than one data center or can process the data in different locations. Therefore it can be difficult to determine the exact location of data (cf. chapter 4.2.2.8) [42].

IT laws state certain criteria that define how data may be processed and also how the transfer of data is handled. A specific topic in this context are the laws of the European Union (E.U.) and the USA. The E.U. established the *European Commission Directive on Data Protection* in 1998 that prohibits the transfer of data to non-European countries that do not meet certain very strict guidelines. In turn, the United States Department of Commerce and the E.U. established the "Safe Harbor" policy agreement in 2000 to harmonize the data protection laws. The "Safe Harbor" agreement regulates the way personal data of European citizens has to be processed to prevent information disclosure or loss. U.S. companies are not required to join the "Safe Harbor" program, but if they do they have to fulfill seven principles stated at the website of the U.S. Department of commerce that deal with the processing of data (e.g. allow access the personal information, implementation of adequate security mechanisms, etc) [123]. The U.S. companies that joined the "Safe Harbor" program are also listed on the website of the U.S. Department of commerce and E.U. organizations can check which companies fulfill the "Safe Harbor" privacy principles [96].

In the context of cloud computing, this is important because customers and also providers have to make sure that they comply with these regulations. This fact stresses why it is necessary to know where the data is hosted.

Another aspect that has to be taken into consideration is the ownership of data. Data in the context of cloud computing can also be seen as virtual machines with regard to IaaS, or applications with regard to PaaS. The definition of the ownership of data host-

ed within cloud environments is required to determine who is liable in case of illegal action. Cloud providers generally do not want to be liable for illegal actions that are caused by customers. For example, if a customer sends spam mails via virtual machines hosted on the environment of an IaaS provider, the customer should be held liable [22].

Therefore, the ownership of the data hosted in a cloud environment should be stated in the SLA between the cloud provider and the customer (cf. chapter 4.2.2.2).

5.3 Conclusion

This thesis was created to give an overview of the different aspects of cloud computing. It is important that it is possible to classify cloud computing into different parts. The architecture presented in chapter 3.2 is a good way to illustrate the whole cloud computing model [77].

In essence, cloud computing enables users to consume services that are hosted on servers. The term cloud computing simply describes the way the services are delivered and how the underlying infrastructure operates. However, the services differ among themselves in many aspects. Therefore, a breakdown is necessary to be able to correctly define these services. A breakdown results in four different service models: IaaS, PaaS, SaaS, and HuaaS. IaaS services enable users to create and run virtual machines, PaaS services support users in deploying and running applications, and SaaS services provide customers with different applications that they can directly consume via different end-user devices. HuaaS is currently still a special form of a service model because it has not been fully adopted across the IT-community. However, it is eligible to be presented among the other service models by delivering “human-capacity” over a network [76, 80].

This classification defines the services that are delivered. However, it is also necessary to structure the way how the actual hardware is used or who the owner of the underlying infrastructure is. Consequently, deployment models are presented which help to define who is the owner, the provider, and the customer of cloud infrastructures [80].

Moreover, it is also necessary to manage the underlying infrastructure and still following the essential characteristics of cloud computing presented in chapter 3.1. This can be done by using cloud management platforms. There are many open-source prod-

ucts on the market that offer this functionality. These can be used to implement any form of deployment model [26, 51].

By mentioning cloud computing, it is also necessary to keep the quality of service aspects in mind. Especially when customers make use of cloud services of a public cloud provider, it is important that the services are delivered in a certain quality. To be able to measure quality, it is necessary to define criteria that are measurable. Such criteria are mentioned in chapter 4.1. Providers of cloud computing should be able to guarantee performance with regard to these criteria and customers should receive a legal basis to insist on this performance. A good approach to solve this issues is the use of service level agreements (SLAs). SLAs enable providers and customers to negotiate target-values for criteria that are important for the customer [72]. Nevertheless, current SLAs of popular public cloud providers are not negotiated with the customer. They are so called off-the-shelf agreements and customers have to accept them if they want to use the services of the providers [34]. Moreover, these off-the-shelf SLAs commonly only deal with guaranteed uptimes and offer just credits for customers if the stated uptimes are not achieved [108]. Additionally, there is a maximum amount of credits a customer can receive and credits can only be used to settle further payments for the cloud service. Therefore, customers do not receive direct compensation for financial damages they suffer because of outages of the service. In addition, customers do not automatically receive credits if the service did not comply with the guaranteed uptime. Customers have to notify the provider and present evidence of the downtime to be able to receive them. This leads to the fact that the customers have to monitor the service by themselves and create log-files to be eligible to receive credits. These factors can be huge obstacles for users that want to move into the cloud. However, there is research going on regarding frameworks that help to improve SLA processes, but they are not widely implemented yet [1, 90].

With the quality of service aspects in mind, it is important to emphasize that these criteria and the SLA issues generally just come into effect when using services from a public cloud provider (or in hybrid cloud environments). If a company hosts its own private cloud, the company itself is in charge of delivering the cloud services in an appropriate quality and has to control the cloud environment.

This all raises the question why someone should use cloud computing. The answer

is that cloud computing has many advantages. Referring to the advantages mentioned in chapter 5.1, there are many fields of application in which cloud computing fits perfectly. Companies that use cloud services are able to demand resources fast and on-demand, which is useful when dealing with loads they cannot estimate. They can benefit from dynamic pricing models and are able to access cloud services from nearly everywhere.

However, companies also have to consider the various disadvantages. There are many issues that still have to be addressed in the context of cloud computing. Companies have to be aware that there can be outages in public (e.g. outage of Amazon EC2) and also private cloud environments (e.g. own data center outage) because cloud services are just based on ordinary processing hardware.

Companies also have to take into consideration that there can be problems regarding the interoperability and the portability of cloud services. Customers that want to switch from one cloud provider to another, or from a public cloud infrastructure to a private cloud infrastructure (or vice versa) have to consider that this may not easily be possible. That is because many cloud providers use proprietary platforms and APIs that are not compatible with others. Of course this can be a fact that is beneficial for the providers because it leads to customer lock-ins and makes it difficult for users to switch providers.

However, current organizations rely on ever increasing data volumes and increasingly complex applications. Therefore it is necessary that they are provided with a flexible cloud computing model where cloud components from various providers are compatible and can be used together. This can be achieved with more open cloud environments. Consequently, a solution to tackle the lock-in issue is the use of open source cloud platforms. The great benefit of open source solutions for cloud management platforms is that customers can easily switch providers or combine services that are based on the same open source cloud management platform [26].

The development towards the use of open source cloud platforms is necessary, because customers demand flexibility and will not accept provider lock-ins when open-source cloud platforms are widely implemented.

Another important topic is the security of cloud environments. As presented in chapter 5.2.3 there are attacks in various forms on cloud infrastructures. These deal

with the cloud service models or with the use of virtualization technologies to provide cloud services. Cloud providers and also cloud customers have to be aware of these security issues and have to take measures to properly secure their operations [67, 114].

The processing of sensitive data within cloud computing is also an issue. Especially when using services from a public cloud provider, customers cannot be sure how exactly the provider processes the data. The customer has to completely trust the provider that the provider does not abuse the stored data. However, in enterprise environments the storage of important data in "foreign" data centers is usually not an option. This is an issue that can either be addressed by encrypting data before storing it within a public cloud environment or by simply not using public cloud environments and relying on own infrastructure (e.g. by using a private cloud environment) [102].

The processing of customer's data can also lead to legal issues with regard to the actual ownership of data, the location of data, and the applicable laws. The *European Commission Directive on Data Protection* and the "Safe Harbor" program are important to harmonize data protection laws, but further developments are still necessary [22, 96].

6 Outlook

This chapter is designed to give an outlook of developments and trends with regard to cloud computing. Various research companies and magazines tried to predict how cloud computing will develop in 2014. An extract of the most informative predictions in the context of cloud computing is presented below.

- Spending on cloud computing will increase

  According to the American market intelligence, analysis and advisory company IDC (International Data Corporation) [68], the spending for cloud services and the technology to enable such services will increase by 25% in 2014 reaching over 100 billion dollars. This figure also includes the spending for software and cloud infrastructure. It also results from the predicted increasing number of data centers that are built by cloud service providers to achieve global scale [37].

- PaaS will grow
  IDC also predicts that more companies will start using PaaS solutions to develop and deploy applications more efficiently. IDC mentions that the PaaS market will reach 14 billion dollars by 2017 (growing from 3.8 billion dollars in 2013) [71, 112].

- Usage of SaaS services instead of running applications on-premise
  There are various enterprise applications on the market that are delivered via SaaS. Examples of such applications that are used by enterprises are HCM (human capital management) or CRM (customer relationship management) tools (e.g. Sales Cloud by Salesforce.com [105]). Many applications that have been available in multiple deployment modes will soon only be delivered via SaaS [113].

- Backup of SaaS data in the cloud
  Companies start to base their recovery strategy on cloud computing services by backing up their data stored within applications of public SaaS providers in cloud infrastructures of specialized SaaS backup vendors. An example of such a specialized SaaS backup vendor is Backupify [25]. This development deals with securing data from accidental or malicious deletion [113].

- Automated encryption of data before moving it into a cloud
  As mentioned in chapter 5.2.4, privacy is a big issue within cloud computing. It is difficult for customers of public cloud providers to ensure that their data is treated confidentially and securely. Therefore, the encryption of data before moving it into public cloud environments will become an essential feature. There are already companies that offer such encryption services, e.g. CipherCloud [32].

- Hybrid cloud deployments will increase
  More companies will start to implement hybrid cloud environments with regard to integration and interoperability. That enables them to combine the security

benefits of private clouds with public cloud advantages like cost-efficiency and easy scalability [52, 112].

As also mentioned in chapter 3.4.5, Garter Inc. predicts that nearly the half of all lager enterprises will use hybrid cloud environments by the end of 2017 [53].

- Number of cloud based web applications will increase

  Applications that can be accessed from everywhere (e.g. via mobile devices) will start to grow and will have a cloud/client architecture. This means that the cloud will be the control point and the application as such can span over multiple devices. By using such architectures, the applications can greatly benefit from scalability features of the cloud [52, 112].

7 References

[1] Alhamad, M., Dillon, T., and Chang, E. 2010. Conceptual SLA Framework for Cloud Computing.
[2] Amazon. Amazon Mechanical Turk. https://www.mturk.com/mturk/. Accessed 27 December 2013.
[3] Amazon Web Services. http://aws.amazon.com/. Accessed 28 December 2013.
[4] Amazon Web Services. Amazon EC2 Instance Purchasing Options. http://aws.amazon.com/ec2/purchasing-options/. Accessed 5 January 2014.
[5] Amazon Web Services. Amazon EC2 Pricing. http://aws.amazon.com/ec2/pricing/. Accessed 17 January 2014.
[6] Amazon Web Services. Amazon Elastic Compute Cloud. http://aws.amazon.com/ec2/. Accessed 26 December 2013.
[7] Amazon Web Services. Auto Scaling. http://aws.amazon.com/autoscaling/. Accessed 26 December 2013.
[8] Amazon Web Services. AWS Compliance. http://aws.amazon.com/compliance/. Accessed 21 January 2014.
[9] Amazon Web Services. AWS Console for iOS and Android. http://aws.amazon.com/console/mobile/. Accessed 18 January 2014.
[10] Amazon Web Services. AWS Elastic Beanstalk. http://aws.amazon.com/elasticbeanstalk/. Accessed 20 January 2014.
[11] Amazon Web Services. AWS Elastic Beanstalk Product Details. http://aws.amazon.com/elasticbeanstalk/details/. Accessed 21 January 2014.
[12] Amazon Web Services. AWS Security Center. http://aws.amazon.com/security/. Accessed 6 January 2014.
[13] Amazon Web Services. Current Status. http://status.aws.amazon.com/. Accessed 10 January 2014.
[14] Amazon Web Services. Regions and Availability Zones.

http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/using-regions-availability-zones.html. Accessed 20 January 2014.
[15] Amazon Web Services. Service Level Agreement EC2. http://aws.amazon.com/ec2-sla/. Accessed 5 January 2013.
[16] Andrieux, A., Czajkowski, K., Dan, A., Keahey, K., Ludwig, H., Nakata, T., Pruyne, J., Rofrano, J., Tuecke, S., and Xu, M. 2007. Web Services Agreement Specification (WS-Agreement). http://www.ogf.org/documents/GFD.107.pdf. Accessed 11 January 2014.
[17] Apache CloudStack. http://cloudstack.apache.org/. Accessed 3 January 2014.
[18] Apache CloudStack. Apache CloudStack: About. http://cloudstack.apache.org/about.html. Accessed 2 January 2014.
[19] Apache CloudStack. Apache CloudStack: Getting Involved. http://cloudstack.apache.org/contribute.html. Accessed 3 January 2014.
[20] Arcitura Education. SLA Management System. http://cloudpatterns.org/mechanisms/sla_management_system. Accessed 10 January 2014.
[21] Arcitura Education. SLA Monitor. http://cloudpatterns.org/mechanisms/sla_monitor. Accessed 10 January 2014.
[22] Armbrust, M., Fox, A., Griffith, R., Joseph, A. D., Katz, R., Konwinski, A., Lee, G., Patterson, D., Rabkin, A., Stoica, I., and Zaharia, M. 2009. Above the Clouds: A Berkeley View of Cloud Computing. Technical Report UCB/EECS-2009-28.
[23] Arredondo, J. R. Start Using Auto Scale Today. http://www.rackspace.com/blog/start-using-auto-scale-today/. Accessed 17 January 2014.
[24] Aternity. Hazy Cloud. The Impact of the Cloud End User Experience. http://www.aternity.com/products/end-user-experience/cloud/. Accessed 10 January 2014.
[25] Backupify. Protect and Save your SaaS Data. https://www.backupify.com/. Accessed 26 January 2014.
[26] Baker, M. 2013. Interoperability and Openness: Why an Open Cloud Is the Only Cloud for the Future. http://allthingsd.com/20131010/interoperability-and-openness-why-an-open-cloud-is-the-only-cloud-for-the-future/. Accessed 20 January 2014.
[27] Bigelow, S. J. What's the difference between a private cloud and a virtual data center? http://searchcloudcomputing.techtarget.com/answer/Whats-the-difference-between-a-private-cloud-and-a-virtual-data-center. Accessed 31 December 2013.
[28] Blaisdell, R. Cloud computing enables business scalability and flexibility. http://www.rickscloud.com/cloud-computing-enables-business-scalability-and-flexibility/. Accessed 16 January 2014.
[29] Burns, L. 2013. Business Cloud Computing: Privacy Is Just As Important As Security. http://readwrite.com/2013/05/28/business-cloud-computing-privacy-is-just-as-important-as-security#awesm=~otNJbO9DXLGERK. Accessed 23 January 2014.
[30] Buyya, R., Yeo, C. S., Venugopal, S., Broberg, J., and Brandic, I. 2009. Cloud computing and emerging IT platforms: Vision, hype, and reality for

delivering computing as the 5th utility. Future Generation Computer Systems 25, 6, 599–616.
[31] Chappell, D. What is PaaS? http://davidchappellopinari.blogspot.co.at/2011/01/what-is-paas.html.
[32] CipherCloud. http://www.ciphercloud.com/technologies/technology-overview/. Accessed 26 January 2014.
[33] Cloud Computing Competence Center For Security. What are Deployment Models in Cloud Computing? http://www.cloud-competence-center.com/understanding/cloud-computing-deployment-models/. Accessed 28 December 2013.
[34] Cloud Computing Use Case Discussion Group. Cloud Computing Use Cases White Paper. http://opencloudmanifesto.org/Cloud_Computing_Use_Cases_Whitepaper-4_0.pdf. Accessed 7 January 2014.
[35] cloudbook: The Cloud Computing & SaaS Information Resource. Accurately Monitoring Cloud SLAs. http://www.cloudbook.net/resources/stories/accurately-monitoring-cloud-slas. Accessed 10 January 2014.
[36] Columbus, L. Cloud Predictive Analytics Most Used To Gain Customer Insight. http://www.forbes.com/sites/louiscolumbus/2013/10/24/cloud-predictive-analytics-most-used-to-gain-customer-insight/. Accessed 2 January 2014.
[37] Columbus, L. IDC's Top Ten Technology Predictions For 2014: Spending on Cloud Computing Will Exceed $100B. http://www.forbes.com/sites/louiscolumbus/2013/12/03/idcs-top-ten-technology-predictions-for-2014-cloud-spending-will-exceed-100b/. Accessed 26 January 2014.
[38] Concur Technologies. https://www.concur.com/. Accessed 28 January 2014.
[39] Concur Technologies. Choose the Edition that is Right for You. https://www.concur.com/en-us/pricing-editions?icid=en_us_h-topnav_pricingeds-products. Accessed 17 January 2014.
[40] Cornerstone OnDemand. http://www.cornerstoneondemand.com/. Accessed 28 January 2014.
[41] Cornerstone OnDemand. Packages and Pricing. http://www.cornerstoneondemand.com/csb/how-csb-works/price. Accessed 17 January 2014.
[42] Cruz, X. 2013. Cloud Computing and its Legal Implications. http://cloudtimes.org/2012/12/03/cloud-computing-and-its-legal-implications/. Accessed 8 January 2014.
[43] Eucalyptus Systems. Eucalyptus. https://www.eucalyptus.com/. Accessed 3 January 2014.
[44] Eucalyptus Systems. Products & Services. https://www.eucalyptus.com/eucalyptus-cloud. Accessed 3 January 2014.
[45] Eucalyptus Systems. Welcome to Eucalytpus. https://www.eucalyptus.com/participate. Accessed 3 January 2014.
[46] Evolven. Downtime, Outages and Failures - Understanding Their True Costs. http://www.evolven.com/blog/downtime-outages-and-failures-understanding-their-true-costs.html. Accessed 4 January 2014.

[47] Facebook. http://www.facebook.com. Accessed 27 December 2013.
[48] Fischbach, L. 2013. A Practical View of Cloud Computing. Seminar Paper, Vienna University of Economics and Business.
[49] Garg, S. K., Versteeg, S., and Buyya, R. 2013. A framework for ranking of cloud computing services. Special Section: Utility and Cloud Computing 29, 4, 1012–1023.
[50] Gartner. http://www.gartner.com/technology/home.jsp. Accessed 26 January 2014.
[51] Gartner. Cloud Management Platforms. http://www.gartner.com/it-glossary/cloud-management-platforms. Accessed 2 January 2014.
[52] Gartner. Gartner Identifies the Top 10 Strategic Technology Trends for 2014. Analysts Examine Top Industry Trends at Gartner Symposium/ITxpo 2013 October 6-10 in Orlando. http://www.gartner.com/newsroom/id/2603623. Accessed 26 January 2014.
[53] Gartner. Gartner Says Nearly Half of Large Enterprises Will Have Hybrid Cloud Deployments by the End of 2017. http://www.gartner.com/newsroom/id/2599315. Accessed 2 January 2014.
[54] Gartner Inc. SLO (service-level objective). http://www.gartner.com/it-glossary/slo-service-level-objective. Accessed 9 January 2014.
[55] GoGrid. GoGrid Cloud Servers. http://www.gogrid.com/products/cloud-servers. Accessed 26 December 2013.
[56] Google. App Engine Pricing. https://cloud.google.com/products/app-engine/. Accessed 17 January 2014.
[57] Google. Gmail. https://mail.google.com/intl/en/mail/help/about.html. Accessed 27 December 2013.
[58] Google. Google App Engine. http://appengine.google.com/. Accessed 26 December 2013.
[59] Google. What Is Google App Engine? https://developers.google.com/appengine/docs/whatisgoogleappengine. Accessed 21 January 2014.
[60] Google. Why App Engine? https://developers.google.com/appengine/whyappengine. Accessed 17 January 2014.
[61] Google. 2013. App Engine Service Level Agreement. https://developers.google.com/appengine/sla. Accessed 5 January 2014.
[62] Hewlett-Packard Development Company. Get to know HP Public Cloud. Open, enterprise-grade public cloud based on OpenStack® technology. http://www.hpcloud.com/. Accessed 20 January 2014.
[63] Hofmann, P. and Woods, D. 2010. Cloud Computing: The Limits of Public Clouds for Business Applications. Internet Computing, IEEE 14, 6, 90–93.
[64] Hyperic. Cloud Status. http://www.hyperic.com/products/cloud-status-monitoring. Accessed 10 January 2014.
[65] IBM. Private and community clouds deliver speed and efficiency for the public sector while maintaining security and control. http://www-07.ibm.com/systems/au/cloud/pdf/GOV_02.pdf. Accessed 1 January 2014.
[66] IBM. What is cloud? Computing as a service over the Internet. http://www.ibm.com/cloud-computing/us/en/what-is-cloud-computing.html. Accessed 28 December 2013.

[67] Ibrahim, A. S., Hamlyn-harris, J. H., and Grundy, J. 2010. Emerging Security Challenges of Cloud Virtual Infrastructure.
[68] IDC. http://www.idc.com/. Accessed 26 January 2014.
[69] Janssen, C. Cloud Application Portability. http://www.techopedia.com/definition/26432/cloud-application-portability. Accessed 6 January 2014.
[70] Janssen, C. Community Cloud. http://www.techopedia.com/definition/26559/community-cloud. Accessed 1 January 2014.
[71] Kanaracus, C. PaaS market to reach $14B by 2017, IDC says. http://www.computerworld.com/s/article/9243900/PaaS_market_to_reach_14B_by_2017_IDC_says. Accessed 26 January 2014.
[72] Kandukuri, B. R., Paturi, V. R., and Rakshit, A. 2009. Cloud Security Issues.
[73] Keller, A. and Heiko, L. 2002. The WSLA Framework: Specifying and Monitoring Service Level Agreements for Web Services. http://domino.watson.ibm.com/library/cyberdig.nsf/papers/CDEDB79080F59EE285256C5900654839/$File/RC22456.pdf. Accessed 11 January 2014.
[74] Kleyman, B. 2012. Hypervisor 101: Understanding the Virtualization Market. http://www.datacenterknowledge.com/archives/2012/08/01/hypervisor-101-a-look-hypervisor-market/. Accessed 11 December 2013.
[75] KVM. Kernel Based Virtual Machine. http://www.linux-kvm.org/page/Main_Page. Accessed 11 December 2013.
[76] Lenk, A., Klems, M., Nimis, J., Tai, S., and Sandholm, T. 2009. What's inside the Cloud? An architectural map of the Cloud landscape. In Proceedings of the 2009 ICSE Workshop on Software Engineering Challenges of Cloud Computing. IEEE Computer Society, 23–31.
[77] Liu, F., Tong, J., Mao, J., Bohn, R., Messina, J., Badger, L., and Leaf, D. 2011. NIST cloud computing reference architecture. NIST special publication 500, 292.
[78] LogicMonitor Inc. LogicMonitor. http://www.logicmonitor.com/. Accessed 10 January 2014.
[79] Lowe, D. Disadvantages of Cloud Computing for Networks. http://www.dummies.com/how-to/content/disadvantages-of-cloud-computing-for-networks.html. Accessed 20 January 2014.
[80] Mell, P. and Grance, T. 2011. The NIST Definition of Cloud Computing. Recommendations of the National Institute of Standards and Technology. NIST Special Publication 800, 145.
[81] Metha, N. The 4 Primary Cloud Deployment Models. http://www.cloudtweaks.com/2012/07/the-4-primary-cloud-deployment-models/. Accessed 28 December 2013.
[82] Microsoft. Service Level Agreements. http://www.windowsazure.com/en-us/support/legal/sla/. Accessed 14 January 2014.
[83] Microsoft. Windows Azure. http://www.windowsazure.com/en-us/. Accessed 27 December 2013.
[84] National Institute of Standards and Technology. http://www.nist.gov/. Accessed 26 January 2014.
[85] Neistat, M. 2013. 5 Use Cases for Hybrid Cloud Storage.

http://ussignalcom.com/blog/5-use-cases-for-hybrid-cloud-storage. Accessed 1 January 2014.
[86] NSK. Hybrid Clouds. The best of both Worlds. http://blog.nskinc.com/Default.aspx?app=LeadgenDownload&shortpath=docs%2fHybrid+Clouds+The+Best+of+Both+Worlds.pdf. Accessed 31 December 2013.
[87] Nurmi, D., Wolski, R., Grzegorczyk, C., Obertelli, G., Soman, S., Youseff, L., and Zagorodnov, D. 2009. The Eucalyptus Open-Source Cloud-Computing System.
[88] OpenStack. http://www.openstack.org/. Accessed 2 January 2014.
[89] OpenStack. OpenStack Community. http://www.openstack.org/community/. Accessed 3 January 2014.
[90] Patel, P., Ranabahu, A., and Sheth, A. 2009. Service level agreement in Cloud Computing.
[91] Rackspace. Cloud Monitoring. Monitor everything, and stop problems before they happen. http://www.rackspace.co.uk/cloud/monitoring. Accessed 10 January 2014.
[92] Rackspace. Cloud Servers (Next Gen) SLA,. http://www.rackspace.com/information/legal/cloud/sla. Accessed 14 January 2014.
[93] Rackspace. Global infrastructure. http://www.rackspace.com/about/datacenters/. Accessed 20 January 2014.
[94] Rackspace. Rackspace Cloud Servers. http://www.rackspace.com/cloud/servers/. Accessed 26 December 2013.
[95] Red Hat, Inc. Red Hat Enterprise Linux OpenStack Platform. Build public and private clouds on a secure, scalable foundation. http://www.redhat.com/products/enterprise-linux/openstack-platform/. Accessed 3 January 2014.
[96] Rouse, M. 2005. Safe Harbor. http://searchcio.techtarget.com/definition/Safe-Harbor. Accessed 24 January 2014.
[97] Rouse, M. 2007. horizontal scalability. http://searchcio.techtarget.com/definition/horizontal-scalability. Accessed 6 January 2014.
[98] Rouse, M. 2007. vertical scalability. http://searchcio.techtarget.com/definition/vertical-scalability. Accessed 6 January 2014.
[99] Rouse, M. 2011. cloud bursting. http://searchcloudcomputing.techtarget.com/definition/cloud-bursting. Accessed 1 January 2014.
[100] Rouse, M. 2011. cloud portability. http://searchcloudprovider.techtarget.com/definition/Cloud-portability. Accessed 6 January 2014.
[101] Rouse, M. 2012. community cloud. http://searchcloudstorage.techtarget.com/definition/community-cloud. Accessed 1 January 2014.
[102] Ryan, M. D. 2011. Cloud Computing Privacy Concerns on Our Doorstep. Communications of the ACM 54, 1.

[103] Salesforce.com. http://www.salesforce.com/. Accessed 28 December 2013.
[104] Salesforce.com. Editions and Pricing. http://www.salesforce.com/crm/editions-pricing.jsp. Accessed 17 January 2014.
[105] Salesforce.com. Sales Cloud. http://www.salesforce.com/de/sales-cloud/overview/. Accessed 27 December 2013.
[106] Salesforce.com. Salesforce System Status. http://trust.salesforce.com/trust/status/. Accessed 10 January 2014.
[107] Salesforce.com. Salesforce1 Platform. http://www.salesforce.com/platform/overview/.
[108] Salman, A. B. 2012. Cloud SLAs: present and future. SIGOPS Oper. Syst. Rev. 46, 2, 57–66.
[109] Saravanan, K. and Kantham, M. L. 2013. An enhanced QoS Architecture based Framework for Ranking of Cloud Services. International Journal of Engineering Trends and Technology 4, 4.
[110] Schuller, S. 2010. So You Wanna Be a SaaS Provider? http://gigaom.com/2010/05/23/so-you-wanna-be-a-saas-provider/. Accessed 17 January 2014.
[111] Sohoni, S. Easily Scale Your Cloud With Rackspace Auto Scale. http://www.rackspace.com/blog/easily-scale-your-cloud-with-rackspace-auto-scale/. Accessed 26 January 2013.
[112] Sreedhar, S. Seven Cloud Computing Trends in 2014. http://www.forbes.com/sites/sungardas/2013/12/10/seven-cloud-computing-trends-in-2014/. Accessed 26 January 2014.
[113] Staten, J. Cloud Computing Predictions for 2014: Cloud Joins the Formal IT Portfolio. http://blogs.forrester.com/james_staten/13-12-04-cloud_computing_predictions_for_2014_cloud_joins_the_formal_it_portfolio. Accessed 26 January 2014.
[114] Subashini, S. and Kavitha, V. 2011. A survey on security issues in service delivery models of cloud computing. Journal of Network and Computer Applications 34, 1, 1–11.
[115] Taylor, J. and Ericson, J. Predictive Analytics in the Cloud 2013 - Opportunities, Trends and the Impact of Big Data. http://www.information-management.com/web_seminars/-10024894-1.html. Accessed 2 February 2014.
[116] TechTarget. Using portable applications in cloud computing still difficult. http://searchcloudapplications.techtarget.com/feature/Using-portable-applications-in-cloud-computing-still-difficult. Accessed 6 January 2014.
[117] Terremark North America Inc. Service Level Agreement. Terremark Cloud Service Level Agreement. https://community.vcloudexpress.terremark.com/en-us/product_docs/w/wiki/service-level-agreement.aspx. Accessed 14 January 2014.
[118] The Open Group. Cloud Computing Portability and Interoperability : Cloud Portability and Interoperability. http://www.opengroup.org/cloud/cloud_iop/cloud_port.htm. Accessed 5 January 2014.
[119] The Security Advocate. 2013. Cloud Service Contracts: Breaking Down the All Important Service Level Agreement (SLA). http://www.thesecurityadvocate.com/2013/03/20/cloud-service-contracts-

breaking-down-the-all-important-service-level-agreement-sla/. Accessed 8 January 2014.
[120] The University of Iowa. Iowa Electronic Markets. http://tippie.uiowa.edu/iem/. Accessed 27 December 2013.
[121] Trappler, T. 2010. If It's in the Cloud, Get It on Paper: Cloud Computing Contract Issues. http://www.educause.edu/ero/article/if-its-cloud-get-it-paper-cloud-computing-contract-issues. Accessed 8 January 2014.
[122] Trossman, A., Pang, J., and Peay, A. SmartCloud Foundations: Inside IBM's OpenStack-based products. https://www.openstack.org/summit/portland-2013/session-videos/presentation/smartcloud-foundations-inside-ibm-s-openstack-based-products/#video. Accessed 20 January 2014.
[123] U.S. Department of Commerce. Safe Harbor Privacy Principles. http://export.gov/safeharbor/eu/eg_main_018475.asp. Accessed 24 January 2014.
[124] Vecchiola, C., Pandey, S., and Buyya, R. 2009. High-Performance Cloud Computing: A View of Scientific Applications.
[125] VMware. http://www.vmware.com/products/vsphere-hypervisor/. Accessed 11 December 2013.
[126] Ward, S. 5 Disadvantages of Cloud Computing. Consider These Before You Put Your Small Business In the Cloud. http://sbinfocanada.about.com/od/itmanagement/a/Cloud-Computing-Disadvantages.htm. Accessed 20 January 2014.
[127] Webopedia. cloud portability. http://www.webopedia.com/TERM/C/cloud_portability.html. Accessed 6 January 2014.
[128] Weinhardt, C., Anandasivam, A., Blau, B., Borissov, N., Meinl, T., Michalk, W., and Stößer, J. 2009. Cloud Computing – A Classification, Business Models, and Research Directions. Business & Information Systems Engineering 1, 5, 391-399.
[129] Whittaker, Z. Amazon Web Services suffers outage, takes down Vine, Instagram, others with it. http://www.zdnet.com/amazon-web-services-suffers-outage-takes-down-vine-instagram-flipboard-with-it-7000019842/. Accessed 5 January 2014.
[130] wiseGEEK. What Are Service Level Objectives? http://www.wisegeek.com/what-are-service-level-objectives.htm. Accessed 9 January 2014.
[131] Xen Project. http://www.xenproject.org/users/cloud.html. Accessed 11 December 2013.
[132] Youseff, L., Butrico, M., and Da Silva, D. 2008. Toward a Unified Ontology of Cloud Computing. In Grid Computing Environments Workshop, 2008. GCE '08, 1–10.
[133] Zhang, Q., Cheng, L., and Boutaba, R. 2010. Cloud computing: state-of-the-art and research challenges. Journal of Internet Services and Applications 1, 1, 7–18.