Towards an Open Service Compendium: Evaluation and Extension of Brokering Technologies

by

Fabian Knaack

Matriculation Number 339002

An exposé submitted to

Technische Universität Berlin
Faculty IV - Electrical Engineering and Computer Science
Department of Telecommunication Systems
Service-centric Networking

Bachelor Thesis

May 8, 2015

Supervised by:
Prof. Dr. Axel Küpper

Assistant supervisor:
Prof. Dr.-Ing. Sebastian Möller
Erklärung
Hiermit versichere ich, dass ich diese Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

I hereby declare that I have created this work completely on my own and used no other sources or tools than the ones listed.

__________________________________________________________
Berlin, May 8, 2015           Fabian Knaack
Preface

This thesis is submitted in partial fulfillment of the requirements for a Bachelor’s Degree in Information Systems Management (‘Wirtschaftsinformatik’) at TU Berlin. The supervisor of this thesis is Prof. Dr. Küpper.

Acknowledgments

My interest in the area of this thesis grew out of my studies at the faculty Service-centric Networking (SNET). Having attended and mastered nearly all offered courses, the topic cloud computing caught my interest and still fascinates me.

I owe special gratitude to Mathias Slawik and Dirk Thatmann for giving me the opportunity to write my thesis within the context of the interesting and challenging topic and their advice on my thesis. Through supporting me by having an open ear for all of my questions despite the demands on their time, they encouraged me to focus on the relevant parts on this broad topic.
Abstract

The primary purpose of this thesis is to design a domain-specific cloud storage service description vocabulary and implement the functionality to match cloud services by using this designed taxonomy and compare their characteristics through an overview. Therefore, the methodologies of the Design Science in Information Systems Research Framework by Alan et al. are applied. The aim is to contribute new approaches towards the Open Service Compendium – a crowd-sourced platform for cloud services.

In the first part, an extensive analysis of common cloud service characteristics and technologies is conducted. Typical concepts and concerns are identified in the current research progress. This analysis is extended by presenting a new defined cloud storage taxonomy. It shows that the complexity and heterogeneity of cloud service descriptions can be handled and reduced by using the Service Description Language - New Generation.

The thesis then proposes a concept for the implementation service matching and comparing extension. This concept includes the new designed cloud storage taxonomy and is based on current broker technologies that have been analyzed and identified.

Finally, the thesis provides an evaluation that takes the findings of the previous chapters into account. The thesis will prove that the previously made assumptions are valid and paves the way for the Open Service Compendium.

Keywords: Cloud Computing, Cloud Storage, Cloud Service Brokering, Open Service Compendium, Cloud Service Taxonomy
Zusammenfassung


Anschließend wird ein Konzept zur Auswahl und zum Vergleich vorhandener Cloud Dienste ausgearbeitet. Dieses Konzept nutzt dabei das Cloud-Speicher Vokabular und basiert auf Ansätzen aktueller Broker-Technologien.

Schließlich wird die Allgemeingültigkeit der Ergebnisse aus den vorangegangenen Kapitel bestätigt. Die Evaluierung zeigt, dass die bisher getroffenen Annahmen anerkannt sind und öffnet den Weg für zum Open Service Kompendium.

Keywords: Cloud Computing, Cloud Storage, Cloud Service Brokering, Open Service Compendium, Cloud Service Taxonomy
# Contents

1 Introduction
   1.1 Motivation ........................................ 1
   1.2 Research Questions ................................ 2
   1.3 Approach & Research Methodologies ............... 2

2 Background & Related Work .......................... 5
   2.1 Cloud Computing .................................. 5
      2.1.1 Definition .................................. 5
      2.1.2 Deployment Models ............................ 6
      2.1.3 Service Models ............................... 7
   2.2 Cloud Related Concepts & Technologies .......... 8
   2.3 Trends ............................................ 9

3 Analysis of Cloud Services .......................... 11
   3.1 Application Environment & Methodologies ......... 11
   3.2 Benefits & Concerns ................................ 12
   3.3 Cloud Research Standardization .................. 13
   3.4 Cloud Service Vocabulary ......................... 14
   3.5 Cloud Storage .................................... 15
      3.5.1 Scope & Definition ........................... 15
      3.5.2 Storage Characteristics ....................... 16
   3.6 Persistent Cloud Storage Vocabulary ............. 16
      3.6.1 Data Storage Properties ....................... 18
      3.6.2 Security .................................. 18
      3.6.3 Features ................................ 19
      3.6.4 Availability & Reliability .................... 19
      3.6.5 Charging ................................ 20
      3.6.6 Provider ................................ 20

4 Implementation ...................................... 21
   4.1 Cloud Service Broker .............................. 21
   4.2 Open Service Compendium Extensions ............... 22
      4.2.1 Persistent Cloud Storage Vocabulary ....... 22
      4.2.2 Faceted Search ............................... 23
      4.2.3 Cloud Service Comparison .................... 25
## Evaluation

5.1 Method Selection ......................................................... 27
5.2 Analytic Broker Vocabulary Evaluation .......................... 27
5.3 Empirical Persistent Vocabulary Evaluation ..................... 28
5.4 Experimental Persistent Vocabulary Evaluation ................. 29
5.5 Discussion .................................................................. 30

## Summary & Outlook

6.1 Recommendations .......................................................... 31
6.2 Future Work ................................................................. 32
6.3 Publication of the Source Code ........................................... 33

## Conclusion ..................................................................... 35

List of Figures .................................................................... 37

Bibliography ....................................................................... 39
1 Introduction

“The interesting thing about Cloud Computing is that we’ve redefined Cloud Computing to include everything that we already do... I don’t understand what we would do differently in the light of Cloud Computing other than change the wording of some of our ads.”

—Larry Ellison (Oracle’s CEO), quoted in the Wall Street Journal, September 26, 2008

1.1 Motivation

Situation  Over the past decade, technology has changed significantly. It began with the mainframe in the 1960s, followed by the distributed computing model of client/server computing, and subsequently the emergence of the Internet and web (Bakshi [5]). Nowadays, almost all devices are able to connect and communicate easily through the internet together. Through global networking and fast changing technology, usage of technology for fulfilling daily life especially over the internet increased. Internet users digitally interact with a growing number of different service providers on a daily basis (Florencio and Herley [25, p. 657]) exchanging information from the Internet of Things (IoT)\(^1\). This development enables the path for a new area of services: cloud computing.

Starting as a hype, the evolution of cloud computing over the past years has been impressive, right now it is the state-of-the-art. According to Armbrust et al. [4], cloud computing allows infinite computing resources available on demand, eliminates the up-front commitment by cloud users and empowers the ability to pay for use of computing resources on a short-term basis as needed. These key features create new opportunities for business companies and lower the entry barriers for start-ups. They do not need to invest in large scale IT infrastructure any longer and can access the required innovative technologies immediately for any (short) term needed (Marston et al. [44, p. 178]). Further, cloud computing plays a critical role as it is more and more becoming a utility service such as electricity, water, gas, and communication technologies. Consumers nowadays need to access these utility services everywhere all day long, which means that availability is required at any time.

\(^1\) The Internet of Things (IoT) is here today in the devices, sensors, cloud services and data your business uses (Microsoft [49]).
Chapter 1. Introduction

Problem  Meanwhile, the rapid development of new cloud computing services has gained momentum and leads to a huge marketplace with a variety of several cloud service solutions. Although there are multiple vendors offering similar solutions, the specifications of these cloud services are highly heterogeneous and less interoperable. Every provider has evolved its own solution using different technologies, proprietary interfaces, formats (Höfer and Karagiannis [36]) and providing non-formalized service description information (Slawik and Küpper [68]). Additionally, nearly all clouds do not share any information about internal workings - this creates a dependency (vendor lock-in effect) for the end user and makes it difficult to migrate from one vendor to another. This lack of common standards for cloud services, service descriptions and interfaces makes it impossible to compare different service solutions. Furthermore, vendors create a big marketing hype by relabeling existing services to fit it to the cloud movement (Gillett [31]).

Despite the lack of standardization, cloud computing raises questions about availability, reliability, legislation, privacy, and especially security (Subashini and Kavitha [71]). One of the main challenges preventing the adaption of cloud service models, according to an IDC IT Cloud Services User Survey, is security (Clavister [14]).

1.2 Research Questions

1. Analytic evaluation
   a) What are the current cloud broker approaches in science?
   b) Do the structured and modeled cloud service vocabulary assumptions of the current broker system still comply with the current research progress?

2. Scope: Cloud Storage Broker Vocabulary
   a) How can current cloud storage solutions be described with a formal vocabulary?
   b) What requirements on the classification of features do the consumers of cloud storage services have?
   c) Do the modeled cloud storage vocabulary capture the descriptions of existing cloud storage services?

3. Broker System
   a) How can a faceted search be implemented technically into the current broker system?
   b) How can different broker services be compared throughout the broker system? Can the selection of features be structured and modeled into categories?

1.3 Approach & Research Methodologies

In this thesis, I will develop an extension and propose an evaluation of the current TRusted Ecosystem for Standardized and Open cloud-based Resources (TRESOR) ecosystem developed by Mathias Slawik from the Technische Universität Berlin (TU Berlin). This paper will

2[https://www.cloud-tresor.de](https://www.cloud-tresor.de) the current broker system will be explained in detail in Chapter 4.
take up on the previous work done by Thatmann et al. [74], Slawik and Küpper [68], and Zickau et al. [89] and focus on the cloud computing storage topic along with current research methodologies, however performance benchmarks or security aspects and recommendations are out of scope. My contribution is one part of the development from the current TRESOR service broker towards an Open Service Compendium – an open crowd-sourced platform for easily describing, selecting and comparing every kind of (cloud computing) service.

To achieve this, I will follow the principles of the Design Science in Information Systems Research Framework by Alan et al. [2, p. 80]. Therefore, I have identified and designed two design artifacts: a persistent cloud storage vocabulary as well as an extension to easily search and compare different cloud services. According to the Design-Science Research Guidelines [2, p. 83], each of the seven guidelines will be applied for every artifact in specific context of (market-oriented) cloud broker systems.

Consequently, Chapter 2 introduces the application environment cloud computing, discussing major contributions regarding this broad area of research besides describing the relevance of cloud computing in the near future and the upcoming market-oriented cloud broker architectures. Then, current approaches in science will briefly compared and analyzed with the solution provided by the TRESOR broker system. By means of an analytic evaluation, weaknesses, potential possibilities, and improvements should be revealed or the implemented assumptions confirmed.

In Chapter 3, the existing cloud vocabulary will be extended in the domain-specific business pertinence of cloud storage. Therefrom, current cloud computing service characteristics will be identified by analyzing recent surveys and studies in the field of cloud computing along with the information provided by the cloud service providers. The identified cloud storage vocabulary is then explained in detail and possible options for each extension is given.

The design of the approach (the second artifact aims to extend the service broker with two new features) is presented in Chapter 4. A faceted search will be implemented allowing the consumer to easily discover and select services regarding his desired necessity. Therefore, the previously identified vocabulary will be used as possible filter choices. On the other side, a table-based comparison overview should allow the user to comfortably correlate different services.

In Chapter 5 an experimental evaluation verifies the defined vocabulary using existing cloud services, and a preliminary empirical evaluation – conducted by a semi-structured interview with a business company on one side and a survey with possible consumers on the other – verifies whether the identified vocabulary fits with the consumers’ needs.

Finally, Chapter 6 summarizes this contribution and outlines assessments and future work, while Chapter 7 concludes this thesis.
2 Background & Related Work

First, this chapter introduces a precise definition of cloud computing, highlighting its key concepts and differentiating it from the other existing academic and enterprise interpretations. Then, a general overview over the different deployment models is proposed followed by an explanation of three main service models. Finally, related concepts and technologies are compared and current trends presented. This chapter provides the basis for the analysis pursued in Chapter 3.

2.1 Cloud Computing

"The Cloud", as Cloud computing is often referred to, has many different definitions due to its rapid development over the past decade. It is a significant buzzword that is mainly used as a marketing term in a variety of contexts to represent many different ideas. However, there is little clarity what qualifies cloud computing. This is due to the fact that cloud computing, unlike other technical terms, combines existing technologies, such as scalability and virtualization, to fulfill today’s demand for information technology. To put it in a nutshell: Cloud Computing refers to the applications delivered as services over the Internet, the infrastructure or the systems platform in the data centers that provide those services (Armbrust et al. [4]). The vast amount of cloud computing services risen over the last years, as well as the lack of formalized descriptions and standards, lead to many different definitions.

2.1.1 Definition

Another reason why there is not one formal definition for cloud computing yet, is because of its complexity: the topic cloud computing covers a vast area of complex approaches, technologies, and products. Both academia and industry have been developing many formal definitions on standardizing the area cloud computing. For example, Vaquero et al. [77] studied around twenty different definitions from a collection of sources to confirm a standard definition. As a result, they conclude their definition is quite long due to of the several cloud characteristics. However, their main focus relies on usability and security. Gartner [30] focus on two aspects only: scalability and elasticity. Bakshi [5], on the other hand, defines multitenant environment besides those two properties of Gartner [30] as a key attribute. In contrast, the definition of Forrester Research [26] covers a pay-per-use model in self-service way, including cloud security.

Regarding those definitions, it is clearly visible that the definitions contain essential key characteristics, nevertheless all of them have a different focus. Therefore, this paper adopts the
Chapter 2. Background & Related Work

Definition of cloud computing provided by the National Institute of Standards and Technology (NIST) \[47\], as it is both broader and narrower than those articulated above and it covers, in my opinion, all essential key aspects of cloud computing:

**NIST Definition of Cloud Computing** \[47\]. 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 includes five essential key characteristics, three service models, and four deployment models. The last two properties will be summarized in Subsection 2.1.2 and 2.1.3. In our contribution a service will be a cloud service if the following essential characteristics are satisfied:

- **On-demand self-service.** The user should be able to provision the service capabilities on his own without any human service provider interaction. If the service is not longer required it can be released – capabilities only have to be paid-on-use.
- **Broad network access.** The computing capabilities should be available over the network and accessible with any kind of client platform.
- **Resource pooling.** The resources (including hardware and network capabilities) are pooled and served over multiple consumers according to their demand and to maximize the overall utilization. Very often, consumers cannot control the exact location of the provided resources.
- **Rapid elasticity.** The resource can be immediately provisioned and released on-demand – at any time. The available capabilities often appear to be unlimited to the consumer.
- **Measured service.** The system automatically controls and optimizes the service capabilities. Resource usages are metered and monitored to provide transparency for both the provider and consumer.

The other side of the coin is, however, that the definition of cloud computing is barely used in everyday life. Another problem linked to the definition is the question whether cloud computing services can be classified in a taxonomy. This will be analyzed in Chapter 3 and evaluated with common academic research evaluation methods in Chapter 5.

2.1.2 Deployment Models

When talking about the term cloud, it appears as a single point of access for all consumer computing needs. A distinction between the cloud concepts is important, as each concept captures certain aspects of the application area and cloud computing (Mather, Kumaraswamy, and Latif \[45\]). The following cloud classifications are adapted from Bohn et al. \[7\], IBM \[37\], Zhang, Cheng, and Boutaba \[85\], and Mell and Grance \[47\].

- **Public Cloud.** Public clouds are hosted, operated, and managed by a vendor in one or more data centers and offered on a fine-grained, utility-computing basis as a service to multiple organizations or individuals. The vendor is responsible for the security management and maintenance operations, thus, the end customer has a low degree of control. The main benefits of public clouds are the overall utilization of the hardware and the lower costs.
2.1. Cloud Computing

**Private Cloud.** Private clouds technically do not differ from public clouds apart from the fact that a private cloud is dedicated to a single enterprise business and is not shared with others. It integrates the traditional IT to deliver many cloud capabilities, targeting to provide more control of the overall resources, the data security and transparency, corporate and law governance, and reliability. Nevertheless, private clouds do not benefit from costs advantages or automatic management. As it is not public, private clouds are only accessible through a Virtual Private Network (VPN). In general, a variety of private cloud patterns have been emerged:

- **Dedicated** Private clouds are hosted internally or at a collocation facility, and operated by internal IT departments.
- **Community** Private clouds are located and managed externally by a third party vendor; regulations (security and compliance requirements) are bound by a contract.
- **Managed** Private cloud infrastructure is owned by the customer and managed by a vendor.

**Hybrid Cloud.** A hybrid cloud is a combination of different clouds: private, public or a mix. It can be a combination of a cloud and traditional IT or various public clouds provided by different vendors. Actually, the most frequently used case is a combination of an existing traditional IT infrastructure combined with a public cloud service provided by a third party. Thereby, companies manage workloads across multiple data centers and clouds. Hybrid clouds benefit by keeping critical core applications and sensitive data internally besides enabling elastic virtual resources and strategic advantages of public cloud services. However, the biggest challenge is the integration of the different cloud services and technologies due to vendor lock-in effects caused by proprietary interfaces.

**Community Cloud.** Community Clouds are similar to hybrid clouds as they are exclusively accessible to a specific community of consumers from organizations with the same shared concerns (e.g., security, policy, and compliance considerations). It can be hosted, managed, and operated internally by the organization or externally by a third-party vendor. Community consumers can access resources of other participants through the connections between associated organizations. In contrast, community clouds generate more benefits than private clouds (general utilization / cost is spread over more users) but less than public clouds.

2.1.3 Service Models

On top of cloud concepts, various categories of cloud services such as infrastructure, platform, and applications have been established. These services are available on demand, payed-per-use, easily scalable, and mostly publicly available. I will discuss the three most prominent services which are Infrastructure-as-a-Service, Platform-as-a-Service, and Software-as-a-Service and give examples for each service.

Apart from those three well-established services, there are many other models nowadays. The ongoing trend indicates a solution for every individual purpose also known as the generic term *Everything-as-a-Service (XaaS)*. This term is used to describe a range of business models.

---

1. XaaS (meaning everything or anything as a Service) is quite a new field of research. A good recommendation for further reading is the whitepaper *Where the Cloud Meets Reality: Scaling to Succeed in New Business Models* published by Accenture, analyzing and outlining the development, recommending common practices and evaluating it by a case study.
delivering nearly anything as a service: Database-as-a-Service, Security-as-a-Service, Storage-as-a-Service, Service-as-a-Service, BusinessProcess-as-a-Service, and tons of other new service models (Accenture [1]). Anyhow, the academic research has not yet evaluated and proven this development so far. The near future development will show whether this trend will continue or disappear.

Depending on the area of application, the NIST Cloud Computing Reference Architecture (NIST CCRA) [7] defines usage scenarios as follows:

**Infrastructure-as-a-Service (IaaS).** Infrastructure-as-a-Service provides the infrastructure and manages the hardware layer including networking, storage, servers, virtualization, and partly OS components. The consumer has the ability to adjust the overlying software and application layer regarding his requirements. Examples of key providers are: Amazon AWS [2], Google Cloud Platform [3], Oracle Cloud [4] and Rackspace [5].

**Platform-as-a-Service (PaaS).** Platform-as-a-Service provides a cloud-based environment without the complexity of managing the underlying hardware and infrastructure. The consumer has access to various interfaces, frameworks and platforms to build his own applications on top of it – the focus relies on the development. Examples of key providers are: Google App Engine [6], Heroku [7] and Microsoft Azure [8].

**Software-as-a-Service (SaaS).** Software-as-a-Service are cloud-based applications available in the cloud via the Internet. The services are fully managed, operated, and owned by a third-party vendor and usually accessible with a web browser. Consumers do not purchase, but rather rent software on a pay-per-use model based on the number of end users, the time of use, the network bandwidth consumed, the amount of data stored or duration of stored data. Examples of key providers are: Google Drive [9] and Salesforce [10].

### 2.2 Cloud Related Concepts & Technologies

In order to determine what exactly differentiates Cloud computing from other existing concepts and technologies, this section compares cloud computing with four other widely-adopted computing paradigms: Cluster Computing, Grid Computing, and Virtualization and Container Technology. In the end, a clear understanding of cloud computing besides other existing concepts should be achieved.

**Cluster Computing.** A computer cluster is a kind of parallel or distributed system containing a collection of inter-connected standalone computers (nodes) performing together as if they were a single integrated computing system. All nodes are cooperatively performing together the same task while being coordinated by a cluster software. Cluster computing enables per-
performance and availability. (Buyya et al. [11])

**Grid Computing.** A Grid is a computing paradigm that enables dynamic resource sharing, selection, and aggregation of geographically distributed resources including storage systems, and supercomputers. Therefore, standards, open protocols, and interfaces are used to solve large-scale resource-intensive problems and deliver various qualities of services including availability, response time, and security. (Foster and Kesselman [28] p. 47ff. and Buyya et al. [11] and Zhang, Cheng, and Boutaba [85])

**Utility Computing.** Utility computing represents a service provisioning model charging consumers based on resource usage demand. Through dynamically provisioning and pay-per-use pricing, overall resource utilization can be maximized and operational costs minimized. Cloud computing can be discovered as a realization of utility computing, whereas utility computing can be implemented without using cloud computing. (Zhang, Cheng, and Boutaba [85])

**Virtualization & Container Technology.** Virtualization technology describes the abstraction of physical hardware and provides virtualized resources for high-level applications. A virtualized server is commonly called a virtual machine (VM). Meanwhile, virtualization techniques are applied to all IT layers to provide the capability of pooling computing resources from clusters of servers, dynamically assigning or reassigning virtual resources to applications immediately and delivering services on-demand (VMWare [80]). The state-of-the-art in virtualization is the container-based virtualization method, for example Docker [11] providing a standardized environment to build, ship, and run distributed applications that work on any infrastructure, quickly and reliably. It takes care of the dependencies and responds to changes on-demand. Nowadays, the Docker platform can be even found and used in personal products.

To summarize, cloud computing shares certain aspects with grid computing and utility computing but differs from them in other aspects. Through virtualization and container technology it is achieved to provide computing resources as a utility that can be dynamically scaled and delivered on-demand.

### 2.3 Trends

Currently, cloud computing is the state-of-the-art and can be found nearly everywhere. With the Internet of Things spreading new features such as the GPS location, RFID, Bluetooth beacons, and even health data to our everyday objects and appliances, and datafication throughout machines and daily life services, almost everything is connected. All this information is transferred to huge data stores and analyzed, calculated, and evaluated immediately with the help of cloud computing. As reported by Cisco, more than three quarters of all data traffic will be processed through the cloud by 2018. Compared to 2013, the data traffic over the cloud is about to rise by a third [29]. However, the most impact of cloud computing still prevails in the enterprise business sector.

Most financial analysts belief that there will be a enormous expansion of cloud computing over the next few years. The International Data Corporation (IDC) [12] predicts that expenses for public IT cloud services will increase to more than $127 billion in 2018 ((IDC) [38]). In other

---


words, the growth rate for public IT cloud services will multiply about six times higher than for the overall IT market. This progress is supported by the Computerworld’s 2015 Forecast[17] from the International Data Group (IDG)[17] as investments in cloud computing increases up to 42% and in cloud storage up to 36%.

According to IBM[48], nearly all new software developed is only available as SaaS in the cloud. Predictions estimate that by 2016 over a quarter of all applications will be accessible over the cloud. An overall cloud market growth adoption of about $250 billion investment from enterprises is being expected by 2017. A recent Hosting and Cloud Study[43] commissioned by Microsoft confirms the upcoming forecasts. More enterprise business are planning to migrate traditional IT to the cloud, especially due to availability, performance, backup, and recovery profit (LLC[43]).

This overall progress gets reinforced by the Google Web Search Trends. The web search popularity, as measured for the terms cloud computing and cloud storage by the Google Web Search Trends during the investigation for this bachelor thesis, indicates in Figure 2.1 a high request for the topic cloud computing in general while cloud storage request is only slowly increasing. However, as forecasted by IDG earlier in this section, the cloud storage demand and interest will rise phenomenally in 2015 as there will be increasing demand.

![Figure 2.1: Google Trend: Cloud Computing & Cloud Storage](image)

Another trend indicating a high demand for cloud storage is recognized in the private area. Private Network Attached Storage (NAS) systems implement the functionality of personal private clouds that can be connected with existing public clouds by third-party vendors. Even the docker technology is integrated[14] – if required, a consumer can easily transfer his private system to a cloud environment without bothering with complex adjustment modifications.

3 Analysis of Cloud Services

Consumers of externally delivered services always desire a set of formalized requirements against which they can assess the differences of each service. This need is particularly present for cloud-based services, as their additional level of abstraction besides the lack of a formal description and taxonomy makes it more complicated to compare the variety of existing and emerging services against each other (Smith [69]).

The overall intention of this contribution is to provide a description vocabulary designed for private and corporate use. As some cloud computing services differentiate between corporate and private use, the formulated vocabulary should capture all necessary aspects.

This chapter outlines the characteristics of the application environment, presents the methods for the requirements analysis, and reviews main benefits and concerns in terms of application environment. Cloud research standardization approaches are introduced and previously stated contributions contrasted with the state-of-the-art of current academic researches. The end of this chapter presents a new refined persistent cloud storage vocabulary, used as the basis for the design evaluation in Chapter 5.

3.1 Application Environment & Methodologies

Application Environment. The application environment are service broker ecosystems providing a marketplace for cloud services to consumers and providers as well as third-party providers offering these services. One example is the TRESOR ecosystem, a cloud service interface supporting service selection decisions and secure service booking.

Methodologies. First of all, existing cloud-related literature was reviewed to become familiar with the application domain. Based on the literature search, current research approaches are analyzed to compare the current TRESOR cloud broker concepts with the state-of-the-art. Possible improvement opportunities should be identified, revealed, and if necessary, obliterated. In addition to the existing domain vocabulary, a domain-specific cloud storage vocabulary model will be introduced. The developed vocabulary model results from analyzing recent surveys and benchmarks, existing research approaches, studies in the field of cloud storage, an extensive market analysis, and comparing information descriptions provided by the cloud service providers. All intermediate results during the development progress were discussed with two experts from the cloud computing environment regarding their suitability in the specific context.
3.2 Benefits & Concerns

The adoption of cloud-based services is caused by its major advantages. By its rapid elasticity ability it can react on demand if the service utilization varies with time – e.g., if there is only seasonal or periodical demand – or even if the demand is unknown in advance. The consumer only has to pay-on-use and can scale up or down immediately. There are no longer any over-provisioning nor under-provisioning costs and problems. Generally, a higher expectation of performance compared to traditional IT services can be assumed. With cloud computing new opportunities can be realized through its efficiency boost: [1.] the price of an infrastructure service for 1000 hours is equal to [2.] the price for 1000x the same infrastructure service for only one hour. Nonetheless, it is possible to compute a result (especially movie frames) much faster in a shorter time with the second option. Another great benefit is the multi-tenancy efficiency: nearly all of the cloud-based applications are perfectly optimized to support multiple users at the same time without any performance loss. Altogether, consumers are able to save costs and shift risk[4, p. 11] while providers can save costs by improving the overall average server utilization, benefit from economical benefits such as electricity bulk discounts, and maintenance expertise. Dillon, Wu, and Chang [21, p. 30f.] and Harms and Yamartino [34] and Marston et al. [44, p. 180-182])

Still, cloud computing is not suitable for every business case. Initially, it should be considered whether the expected use case can be realized and is permitted by the law. Another aspect often left behind are data transfer bottlenecks. Armbrust et al. [4] and Bergen, Coady, and McGeer [6] argue that some enormous data-intensive services are mostly not qualified for the cloud due to time-consuming transfer costs. Especially in private environments the client bandwidth is almost always too low to make the cloud applicable for large data sets. Besides that, the major risks in cloud computing are still security concerns and the lack of privacy. The Gartner CIO survey presented by Harms and Yamartino [34] significantly indicates security and privacy as the most important cloud barriers to adoption. This statement is endorsed by Dillon, Wu, and Chang [21] and Subashini and Kavitha [71]. Further information can be found in detail in Mather, Kumaraswamy, and Latif [45], Grobauer, Walloschek, and Stocker [32], and Subashini and Kavitha [71]. Despite this estimation, Subashini and Kavitha [71] clarify as well that enterprises have a common interest in adopting to the cloud if security responds to regulatory drivers. On the hand, private users are eager to forfeit privacy for some extent, e.g., an added-value service like Facebook. (Cachin, Keidar, and Shraer [12])

Additionally, cloud services offer high availability with a barely present maintenance window. However, this is not always true regarding the latest reports. In the beginning of 2014, the Dropbox service went down for about three hours [33]. Later that year, initially Microsoft Azure had an eleven hour outage [82] followed by Amazon Web Services [41] a few days later.

1 Datacenter operators can benefit from various economic aspects: subsidization from the local government, electricity bulk discounts due to gigantic purchase volumes, and general expertise: huge datacenters need less administrators as a result of vast expertise and specialization (Armbrust et al. [4, p. 10f.])
2 [45] give a broad overview about security concerns in cloud computing, [32] try to accomplish a solid understanding and a general overview regarding essential cloud vulnerabilities while [71] evaluates cloud service models in detail in terms of security, privacy, and trust. Current security issues are outlined in addition to possible solution suggestions
3.3 Cloud Research Standardization

To overcome the problem of a lack of definition and standards several task-force groups, councils, forums, and unions have been formed including enterprise members, academic research institutions, and leading operations. The overall goal is to define global (open) standards, definitions, and interfaces that are successfully adopted all over the world, and prevent common concerns. Furthermore, benefits should be increased, achieving interoperability along with encouraged competition (Dillon, Wu, and Chang [21]). A detailed overview of current cloud standardization concepts can be found in Cloud Standards Wiki [15]. Below, this section will provide a brief overview of some approaches and discuss appearing challenges.

The main focus of the Cloud Standards Customer Council (CSCC) is the end customer. They support end customers in terms of security and supply practical guides according to cloud requirements and best practices regarding cloud architecture and performance.

The Distributed Management Task Force (DMTF) initially focused on standardizing interactions between cloud environments with the Open Cloud Standards Incubator. After task completion in July 2010, it was integrated into the Cloud Management Working Group (CMWG) that is trying to establish a set of prescriptive specifications. In addition, the DMTF provides the Open Virtualization Format (OVF), an “open, secure, portable, efficient and extensible format for virtual machines”3.

Another organization, approved by the European Union, is the European Telecommunications Standards Institute (ETSI). There, the Cloud Standards Coordination (CSC) is developing a variety of standards while the Technical Committee (TC) clusters Information and Communication Technologies (ICT) with the main emphasis on interoperability with mobile communications.

A standardization of network protocols and the interfaces is developed by the Global Inter-cloud Technology Forum (GICTF) while Open Grid Forum (OGF) provides forms of advanced distributed computing, architectural blueprints, and specific interfaces.

As already seen in the beginning of Chapter [2], the NIST is working on cloud definitions and specifications in collaboration with industry and government and tries to identify gaps in the development process and existing approaches.

OCC, the Open Cloud Consortium, provides several approaches for the area cloud computing: benchmarks, reference implementations, standards, and frameworks for interoperating between clouds. Another framework, the Unified Cloud Interface Project4, has been developed by the Cloud Computing Interoperability Forum (CCIF). However, it looks like the progress has been stopped as there are no activities nor changes any longer. Another interface, the Cloud Data Management Interface (CDMI) by the Storage Networking Industry Association (SNIA), could be established as an ISO Standard ISO/IEC 17826:20125.

Finally, the Cloud Security Alliance (CSA) worked on standards for security and privacy, at the moment they provide “education on the uses of Cloud Computing and best practices to help ensure a secure cloud computing environment”, along with Cloud Security certification (CSA [18]).

\[^{3}\text{http://dmtf.org/standards/ovf}\]
\[^{4}\text{https://code.google.com/p/unifiedcloud/}\]
\[^{5}\text{http://www.iso.org/iso/catalogue_detail.htm?csnumber=60617}\]
Chapter 3. Analysis of Cloud Services

Summary. A huge variety of different solutions is available offering empiric knowledge. Anyhow, the major shortfall is the support by the industry as none of the key cloud vendors, neither Microsoft, Google, nor Amazon implement or follow this standardization process. Current progress indicates that the so-called defacto standard by businesses (e.g., Google Cloud products) is widely adopted by developers. This fact leads to the question whether standardization is useful in terms of rapidly changing elastic software.

3.4 Cloud Service Vocabulary

The challenge of the service selection is to overcome the lack of definition, complexity of non-formalized descriptions, and semi-structured data. Therefore, multiple Service Description Languages (SDL) have been proposed that are applied in service broker systems. This section introduces several SDLs in comparison to the SDL used by the TRESOR Broker. An analytic evaluation of the persistent business vocabulary assumptions by Slawik and Küpper [68] is presented in Chapter 5.

Service Description Languages. The Linked-USDL by Pedrinaci and Leidig [55] is based on the Unified Service Description Language (USDL) presented by Oberle et al. [53]. The goal is to capture service descriptions in “an open, scalable, and highly automated manner” [54]. The author concludes its broad adoption because of the usage of Linked Data. Nevertheless, “Linked USDL does not currently provide support for capturing how providers deliver services in terms of resources needed [and] further work is required for covering aspects such as complex dynamic pricing models” [54].

Another early approach proposed by Trastour, Bartolini, and Gonzalez-Castillo [75] uses RDF models in combination with an ontology. Yet, it is limited by the design of the ontology as it cannot cover all property aspects precisely. SOCCA, the Service-Oriented Cloud Computing Architecture [76], also uses ontologies to separate services into three independent categories. Admittedly, the modeled ontologies seem to be very limited in this proposal.

In the early stages of SDLs DAML-S [16] was formed. By that time it was very impressive but complex as it could not accurately describe a domain or concept expressively enough (Nawoj and Gorniak [51, p. 4 - 13]). Later, OWL-S extended this concept focusing on domain-specific web service descriptions. The Semantic Cloud Service Broker by Ngan and Kanagasabai [52] adopts OWL-S to capture complex service constraints. However, the modeled ontologies are mainly focused on hardware specifications and developer features. With the help of Semantic Web Rule Language (SWRL) rules the used OWL-API discovers matching semantic constraints. These matchmaking rules limit the approach as they are quite time-intensive due to their complexity.

In contrast, the Service Description Language - Next Generation (SDL-NG) framework [68] by Slawik and Küpper [68] is tackling these main concerns by providing a concept that benefits from the implementation based on Ruby. Services can be described by using natural language so everybody can describe and understand service descriptions. This relies on the fact Ruby has a “low syntactic noise” [68, p. 7]. The framework also enables to export and import (existing) service descriptions to reduce redundancy. Still, it differs from existing SDLs as it allows to

---

6 http://www.w3.org/Submission/OWL-S/
7 https://github.com/TU-Berlin-SNET/sdl-ng
parse HTML descriptions dynamically. In conclusion, the SDL-NG framework lowers effort by providing a simple tool to describe services understandable for everyone. More information about the SDL-NG are explained in detail in Slawik and Küpper [68].

3.5 Cloud Storage

Since the announcement of Dropbox, a cloud storage service provider using AWS S3, cloud storage services gained popularity rapidly. Nowadays, hundred of millions of users are using cloud storage services and the number is still growing. The main benefit of cloud storage is its utilization: clients can easily upload and backup their data into the cloud with relatively little effort. In contrast, reliability, privacy, and security concerns arise as clients do not fully trust providers. These concerns are strengthened through the latest scandals induced by the NSA.

This Section gives a brief introduction of the topic cloud storage, identifying and highlighting major cloud storage characteristics. The previously discussed persistent TRESOR vocabulary will be extended and refined regarding the domain cloud storage. The end of this section presents and explains the designed cloud storage taxonomy model, used as the basis for the design evaluation in Chapter 5.

3.5.1 Scope & Definition

The intention of this contribution is to cover all necessary aspects for both enterprise and individual customers. The most (commercial) products are designed for industry – including small and medium businesses (SMBs), start-ups, and enterprise companies – however, the private area is increasing rapidly and obtaining interest. On the other hand, there is no implication that those existing commercial services are restricted for the industry only.

**Definition.** In this context, cloud storage stands for (sensitive) data stored remotely on cloud infrastructure. It is accessible with interfaces through the Internet and seems to be infinitely available. According to Borgmann et al. [8], cloud storage services can be divided into two categories:

- **Basic Cloud Storage.** Basic cloud storage is designed to be accessed using an application programming interface (API). Depending on its type, it is separated into block storage, backup storage, and file storage. Examples of such basic cloud storage services are Amazon S3, Microsoft Azure, Google Cloud Storage, and Rackspace.

- **Advanced Cloud Storage.** Advanced Cloud Storage is based on basic storage but provides a sync client or web applications to easily manage existing data. Usually third-party applications are integrated to achieve an added-value service. Examples of advanced cloud storage services are Dropbox, Google Drive, Mega, and OneDrive.

---

8 http://qr.ae/0Sw2W
9 https://aws.amazon.com/s3/
10 https://azure.microsoft.com/services/storage/
11 https://cloud.google.com/storage/
12 https://www.rackspace.com/cloud/block-storage/
13 https://www.dropbox.com
14 https://drive.google.com
15 https://mega.co.nz
16 https://onedrive.live.com
3.5.2 Storage Characteristics

So far, no cloud storage taxonomy has been developed. Only one study, “On the Security of Cloud Storage Services” by Borgmann et al. [8], targets the domain cloud storage and can be considered as an approach in this field. The following part will adapt the concept by Borgmann et al. [8] and extend it by intensive literature studies. Seven dimensions could be identified by clustering properties occurring in current research papers and surveys [83, 71, 84, 35, 79, 39, 12, 46, 56, 61]: security, data storage properties, features, availability, reliability, charging costs, and provider information. The designed cloud storage model is going to be explained in the following section.

Then, the properties have been identified in the most relevant publications: as mentioned earlier, the concept presented by Borgmann et al. [8] covers the main cloud storage characteristics almost completely. This has been approved as all identified cloud storage-related research [70, 23, 86, 22, 12, 46, 33] contains the mentioned terms by Borgmann et al. [8]. Nevertheless, Borgmann’s et al. [8] main focus relies on the security, other important terms could be identified, for example max. storage size, max. storage capacity, and file locking. The benchmarks presented in Drago et al. [23, 22] and Hu, Yang, and Matthews [35] resulting in the conclusion that reliability does matter, particularly regarding response time and bandwidth. However, its measurement is complex and cannot be realized with the current broker system.

After studying the current cloud storage providers, including Amazon [3], Box [9], SugarSync [72], rackspace [56], TeamDrive [73], and mozy [50], and comparison portals [17], additional features could be identified, some of them are currently not covered by research so far, e.g. revocable access control and fine-grained secure audit control.

3.6 Persistent Cloud Storage Vocabulary

The design model of the proposed persistent cloud storage vocabulary is presented in this section. It is based on the results from the analysis of the previous section and will be evaluated in Chapter 5. Some of the designed properties already exist in the current broker vocabulary, however, they are identified as important and are reintroduced again. Again, the scope of this model is to support the decision-making process for individuals and companies. Therefore, not every property might be suitable according to the specific application area.

The classification of the cloud storage features (copy, backup, synchronization and sharing) as described by Borgmann et al. [8] have been perceived. However, they were not identified appropriately as the borders of the above mentioned features intermingle. Thus, they will not be considered in the following context.

http://www.cloudorado.com/cloud_providers_comparison.jsp
http://www.cloudwards.net/comparison/
http://www.cnet.com/how-to/onedrive-dropbox-google-drive-and-box-which-cloud-storage-service-is-right-for-you/-Blog
3.6. Persistent Cloud Storage Vocabulary

Figure 3.1: Persistent Cloud Storage Vocabulary
3.6.1 Data Storage Properties

The dimension data storage properties covers all important aspects of the underlying architecture capabilities and contains techniques to reduce storage and transmission overhead.

**Compression.** To save bandwidth and reduce the upload time, the data can be compressed before starting the upload. This can be combined with delta encoding and data encryption to achieve maximum utilization.

**(Physical) Data Location.** The physical data location is important in terms of regulatory compliance with the law.

**Deduplication.** Deduplication is a commonly used technique in server environments to reduce overhead of storage space by storing only a single copy of each duplicated data. There are different options possible to implement deduplication, depending on the desired level of security and privacy. A detailed explanation can be found in Borgmann et al. [8].

**Delta Encoding.** Delta encoding can be used to reduce transmission overhead, save bandwidth and transfer costs. Instead of uploading an entire file, delta encoding only uploads the changes to a previously uploaded file.

**Maximum File Size.** The actual amount of disk space a file is allowed consume.

**Maximum Storage Capacity.** The total volume of data that can be stored. Some providers offer unlimited storage space, but internal dependencies limit this ability.

**Replication.** Replication indicates whether data is saved in a redundant manner across multiple servers and different locations, so called availability regions. In this context, replication does not imply data integrity checks and there will be no further specification as it is provided by Amazons S3 redundancy type [9].

**Recovering.** Recovering is closely coupled with the replication feature to provide functionality in terms of outages or disaster situations. Depending on its implementation it has additional control features, can be based on policies and monitor protection automatically.

**Version Control.** Versioning can be used for data retention and archiving as it offers an additional level of protection. It can be compared to a backup that tracks changes of a file using numbered versions. Some providers restrict the functionality to a number of available versions or time-limited accessibility.

3.6.2 Security

This category was built as previous introduced trends indicated that security has the most concerns. It is a main reason why most of the consumers struggle to adopt cloud services in general. All important properties regarding the general security, regulation, certification, and encryption have been therefore analyzed and identified.

**Auditing Control.** Internal access control mechanisms log every performed action of a user or any change on a file. The audit log enables the ability to trace events of the past.

**Authentication.** In general, every provider provides and supports a simple login procedure. However, enterprise companies need the option to integrate their existing internal user
database with the service. Therefore, a single sign-on mechanism is required. On the other hand, the login procedure should be secured with the help of a two-factor authentication.

**Authorization.** Authorization is the process of granting a user specific access rights. In a cloud service environment, a granular permission system is required. As most of the services are able to serve multiple users at the same time and they can collaborate together on a same file, a file locking mechanism should provide data loss. Furthermore, time-based access rights and automated access revocation mechanisms should exist.

**Certification.** Certifications are sometimes required due to regulations, however, they indicate and provide a high level of trustworthiness.

**File Encryption.** Encryption specification (parameters): encryption base (container-/folder-based), encryption algorithm, and the key strength.

**Law / Act.** There are specific legal regulations depending on the type of the data, the data location, the data security, and the data responsibility and accountability.

**Monitoring.** Monitoring describes the automated logging, observation, supervision, and performance mechanisms to track service parameters.

**Transmission Encryption.** The appropriate encryption function, including the algorithm and the key strength, that is used to encrypt the data during the communication process.

### 3.6.3 Features

Besides the general data storage properties and its security, additional features can be provided to an additional value.

**Multi-Tenancy.** The ability to provide a service to multiple customers. It is achieved by the application design.

**Platform Compatibility.** Specifies the interoperability of the service with the available operating systems, mobile devices, and the availability of Software Development Kits (SDK).

**Sharing.** Besides the possibility of (public) shareability, trends indicating the demand of collaborative, joint working. The functionality can be separated into private, internal, or public shareability.

### 3.6.4 Availability & Reliability

The following parameters are so-called Quality of Service parameters. Availability of a service can be treated as the major critical criterion because an outage or downtime costs money every single minute. On the other hand, data loss is completely unacceptable regarding data reliability.

**Service-Level-Agreement.** Agreement between the customer and provider about the service level, service parameters, and regulations.

**Bandwidth.** In this context, the bandwidth describes the maximum throughput of the communication process.

**Durability.** The level of durability specifies the average expected loss of a data object.

**Fault Tolerance.** The approach how the service will respond in the event of a failure.

**Response Time.** Often coupled with the bandwidth; the time period (latency time) of a message from a host to a destination.

**Packet Loss.** Indicator of how many packets or messages fail to reach their destination.

### 3.6.5 Charging

Reichman encourages consumers to profit from the cloud computing (cost) benefits. He reveals significant cost differences in comparison to traditional IT infrastructure. Nevertheless, only parts (the first two properties) of this category have been integrated into the SDL-NG because of their distinctiveness.

**Duration.** The minimal contract duration the consumer is bound to.

**Payment Options.** The available payment options the user can choose from.

**Licenses.** Some providers require to buy additional licenses. Mostly, this can be considered as a per-user property.

**Storage Cost.** The price per storage amount.

**Request Cost.** The price per request.

**Transfer Cost.** The price for “incoming” and “outgoing” network data transfer.

### 3.6.6 Provider

Basic provider information is useful to support the decision process or in terms of problems.

**Customer References.** An overview of well-known adapters.

**Contact Information.** Contact details, including the provider’s address, communication numbers, and sometimes additional resources or information.

**Support.** Information about the availability of the service support and the hotline number.
4 Implementation

Cloud Service Brokers are tools to handle the complexity and diversity of cloud service marketplaces. They act as a mediator between consumers and vendors by providing a platform to distinguish between current services. As cloud services are spreading globally and gaining momentum, end users require the possibility to compare different services to maximize their utilization and overcome the restriction of the lock-in effect caused by proprietary interfaces to switch between various providers. Most broker systems have implemented a matchmaking module allowing a service selection against defined criteria along with specific cloud-connector interfaces to enable interoperability across clouds.

This Chapter will briefly specify cloud service broker characteristics using current research approaches. Related broker concepts and interfaces are outlined followed by developed extensions using the TRESOR cloud broker technologies.

4.1 Cloud Service Broker

**Definition.** According to Bohn et al. [7], a broker acts as a mediator between cloud providers and cloud consumers. It takes care of the specific service requirements and manages them in an automated fashion. The heterogeneity of current clouds services and the lack of formalized descriptions remain complex challenges in this application environment.

These challenges can be negotiated by the used technologies of the TRESOR project, a broker entity acting as a supporting mediator between customers and providers. It handles complexity and diversity by unifying all different information through its service description language. Furthermore, pertinent business vocabulary formalizes empiric knowledge in order to support service selection. Additionally, the service broker provides an interface allowing consumers to comfortably query the formalized data as well as authoring service descriptions. The trusted open cloud assessment provides interoperability between different cloud providers and leads towards an open service compendium. Further components of the cloud ecosystem architecture are presented in [74].

**Related Broker Concepts.** Aside from the TRESOR broker, several other platforms have been developed. The widely used open source infrastructure stack Eucalyptus, now known as HP Helion Eucalyptus [1] enables interoperability with the proprietary interfaces of the AWS products. Together with the system provided by MetaCloud [2] and cloudscaling [3] it is part of the

1 [https://www.eucalyptus.com](https://www.eucalyptus.com)
2 [http://www.metacloud.com](http://www.metacloud.com)
Chapter 4. Implementation

OpenStack\footnote{http://www.openstack.org/community} However, not every OpenStack vendor can remain on the market for a long time. Nebula\footnote{https://www.nebula.com} recently announced it “is shutting down”. Other approaches, the Unified Cloud Interface\footnote{https://code.google.com/p/unifiedcloud/} Reservoir\footnote{http://www.manjrasoft.com/aneka_architecture.html} and Aneka\footnote{http://rubyonrails.org} from the academia, tried to take root but failed due to missing acceptance and adaption. The CloudSim toolkit by Calheiros et al.\cite{13} and the InterCloud environment by Buyya, Ranjan, and Calheiros\cite{10} are policy-driven approaches with the main focus on virtual machine performance and sensor measurement. Moreover, the functionality is limited – it looks like they are still in their early beginnings. A quite similar approach to the TRESOR broker is the Cloud-RAID implementation by \cite{63, 64, 65, 67}. Several interfaces to other cloud storage services are separately implemented and a raid-like storage functionality is provided by the system. This concept allows to distribute encrypted chunks to different providers with redundancy. Disadvantages are storage overhead caused by the redundancy and a quite sparse vocabulary. Finally, even a patent has been developed by deMilo et al.\cite{20}.

4.2 Open Service Compendium Extensions

In this section, an extension to the existing current TRESOR broker system is presented. Two new tightly coupled components are implemented – the so called faceted search and a service comparison overview. The implementation is based on Ruby on Rails\footnote{http://rubyonrails.org}, Slim Templates\footnote{http://slim-lang.com} and JavaScript. Concerning HTML and CSS, the user interface follows the paradigm of “graceful degradation”, meaning that if a technology is used that may be unavailable, that technology must fall back to something usable\cite{19}. This means that the application should also be usable with older browsers that do not support HTML5, CSS3 or JavaScript.

4.2.1 Persistent Cloud Storage Vocabulary

The analyzed and designed cloud storage model, previously described in section 3.6 has been migrated with the SDL-NG. Therefore, the analyzed vocabulary was separated in three vocabulary files security, features, and storage properties based in a new domain-specific vocabulary folder storage. Altogether, 19 type classes (including 3 sub types) with a total of 37 type properties, 44 type instances, and 5 service properties are newly created and can be used to describe cloud storages. The vocabulary of the analyzed category provider information is already covered by the current persistent vocabulary of Slawik and Küpper\cite{68}, so it can already be used to describe cloud storage services. However, the category charging is not implemented, as its properties are mostly too specific and only covered by approximately 5 storage service providers.

While testing the new implemented vocabulary with dummy services four problems were discovered and reported.

Firstly, if an existing service should be changed, the integrated Ace editor is not loaded. The developer log console reports the following error message: “Reference Error: Can’t find
variable: ace”. It looks like a race condition issue, because the Ace editor works after a full reload of the web page.

Another error appears after updating an existing service: if the service description contains a multi-value type property, the values are saved twice in the database. This looks like an issue of the update process when saving serialized values to the MongoDB.

After testing several possibilities I was not able to create a service with the service property data_location (Type class location). Again, this could be a bug in the SDL-NG that it type classes containing only sub types cannot be used correctly.

Finally, the latest version of the SDL-NG framework is not capable to differ from type instances with the same name.

4.2.2 Faceted Search

The current broker system functionality allows service providers to register new services throughout the simple broker interface. New services can be created by using the predefined service description vocabulary and have to be approved before being listed. Changes to existing services are stored individually with a version to preserve the exact service properties of past services. On the other hand, cloud users can currently only access a list with all available services. The interface does not provide any filter mechanisms or categorization.

**Analysis.** As no clear guidelines were given, miscellaneous online marketplaces were studied. Besides cloudorado, all marketplaces categorizes their items and provide a direct search functionality. The search options can be refined by predefined features and all search results are presented in a list with additional information. Cloudorado is the only marketplace that has no additional features than presenting the services and their properties in a comparison table. These four functionalities are identified as important, however, the usability in the context cloud computing has to be analyzed.

**Categorization.** Currently, a clear separation of cloud services is quite complex because of their heterogeneity. On the other hand, the latest version of the TRESOR broker system has not yet the functionality to distinguish cloud services. This option will be available in the Open Service Compendium.

**Direct Search Functionality.** This functionality can lead very fast to the desired result, however, it expects that the user knows exactly what he wants. Most of the time, this requirement is not fulfilled. On the other side, a more precise approach requires complex regular expressions and algorithms. Therefore, this approach is not pursued.

**Filter Mechanism.** A predefined filter mechanism enables a fine-grained search based on predefined values. This concept suits with the broker vocabulary as it contains predefined values.

**Concept & Implementation** First of all, with the help of explorative prototyping, several prototypes have been developed to test the usability and fit with the broker system. The attempt
to categorize services with the current service vocabulary was realized by extending the existing type class `:service_category` with predefined instances (`:storage` and `:crm`). Then a new controller was created, displaying only available predefined instance values of the type class `service_category`. As this prototype was working, the next prototype was faced – focusing on the filter mechanism. Therefore, an approach was to use the complete broker vocabulary. A new controller and several templates have been created, parsing the complete service `compendium` types and properties and displaying according to the type the form input types. This prototype indicated soon that these filter possibilities were too enormous and confusing. As a result, this prototype was adjusted to the current approach: displaying only predefined values (in the broker context `instances`) as previously analyzed.

Some parts of the prototypes could be adopted, however, the `faceted search` implementation applies the Model-View-Controller (MVC) design patterns that are integrated in the Ruby on Rails framework. Therefore, only a new controller and five view templates are implemented. As the model was already developed by the SDL-NG framework and can be reused, it is not necessary to introduce another one. The existing `service.rb` model of the SDL-NG framework was extended by a filter method, returning only available approved services with the given parameters. Possible parameters are `key-value pairs` with a name as the key and a predefined value or the Boolean value `TRUE` as a value. The functionality of the implemented controller is very simple. On request, it invokes the filter method of the service model and updates the main view template with the result provided by the model.

The view layer contains most of the "logic". In the main view template `search.html.slim`, the basic skeleton is provided. On top, the filter grid is displayed while the bottom panel contains the available services. The available service properties are parsed in partial views and displayed according to their type. The complete layout of the faceted search model can be seen in figure 4.1.

![Figure 4.1: Implemented Faceted Search functionality with current broker vocabulary](image)

If selected criteria are submitted, the controller returns matching services and the overall view is updated. Only matching services are visible in the bottom panel (see Figure 4.2). Up to 4 services can be selected to be compared in a detailed comparison overview, however, when no
service is selected, a JavaScript event-handler disables the form submit functionality. On the other hand, JavaScript is used to prevent superfluous, resource-wasteful redirects on service comparison. If JavaScript is not available, a fallback mechanism detects the loss and preserves functionality at the expense of additional redirects.

![Figure 4.2: Filtered broker services regarding the selected criteria](image)

Figure 4.2 displays a list of matching services according to the selected criteria. Currently, no further service-specific properties besides the service name are displayed. Nevertheless, in earlier stages a prototype concept was developed that contained additional information. However, this concept was discarded due to the fact that the current cloud service properties are too heterogeneous. So far, there is no survey available regarding the fact which criterion is important.

4.2.3 Cloud Service Comparison

**Requirements.** As the faceted search enables the functionality to filter available services according to their properties, it is required to compare selected services. Therefore, a table-based comparison overview functionality should be implemented, clustering existing service properties into defined categories. Currently, the latest version TRESOR broker system does not yet provide the functionality to cluster properties.

**Design.** A few available providers\(^1\) could be identified that provide the functionality to compare cloud storage services. With the help of these providers design criteria have been identified. The final design result is a simple clean interface with the focus on usability.

**Implementation.** The overall challenge was first to determine a solution to categorize the existing vocabulary and secondly to identify identical vocabulary properties that are used and display them in a matrix. New routes were created in the `config/routes.rb` file that link the request to the correct controller action. The `CompareController` has the main task to receive the requested services and update the view layer. Therefore, the controller requests the required services at the `Service` model by invoking the function `names` with the given `service identifier names`. The `Service` model call returns corresponding MongoDB Documents that are used by the controller to identify all present vocabulary properties and merge them in a `Hash`. If no matching services are be found, an error message is displayed. Otherwise, the number of services, the services with each properties, and the `Hash` containing all identified used vocabulary are forwarded to the view layer.

As in the previous extension, most of the "logic" part is located in the view layer. The services are displayed in a comparison table with their properties. Every row contains a specific prop-

---

\(^1\) [http://www.tomshardware.com/reviews/cloud-storage-provider-comparison,3905.html](http://www.tomshardware.com/reviews/cloud-storage-provider-comparison,3905.html)
[https://www.cloudorado.com/cloud_providers_comparison.jsp](https://www.cloudorado.com/cloud_providers_comparison.jsp)
[http://www.cloudwards.net/comparison/](http://www.cloudwards.net/comparison/)
[http://www.top10cloudstorage.com/compare-specs/](http://www.top10cloudstorage.com/compare-specs/)
[http://www.cloudstoragecomparison.net](http://www.cloudstoragecomparison.net)
property with its name and the corresponding values of the selected services while a col describes a complete service. In general, the designed structure of the vocabulary is flat. However, to realize a category-like feeling, the cloud storage vocabulary had to be restructured. A category is now a *type class* that contains other properties (type classes) or subcategories. This provided solution is not ideal, yet, the current TRESOR broker system does not provide this functionality yet.

Therefore, the *Hash containing all identified used vocabulary* is being parsed. Because a property can contain other properties, instances or sub types the rendering of the partial templates have to be invoked in a recursive manner. In the invoked partial views, the property type and value are going to be analyzed and then parsed correctly. The *number of services* is being used in the partial templates to indicate the col span when a property should present a (sub)category. Figure 4.3 presents the final result of the parsing process and the designed layout over the service comparison overview.

![Figure 4.3: Implemented table-based broker service comparison](image)

**Summary** However, the current implementation has room for improvements:

- The parsing of the service properties is currently quite complex and has duplicated code. It could be simplified by differing the type in separate partial templates.
- The categorization of services should be made by the SDL-NG and not in the broker system.
- The two newly created functions `names` and `filter` using similar queries. Therefore, the common code could be optimized by combining the same functionalities.
- In general, the development process for external users could be enhanced by providing detailed code (functionality) descriptions.
5 Evaluation

This chapter carries out different types of evaluations to prove the assumptions presented in the previous chapters. In the following sections, the selected evaluation methods are briefly introduced. Then each evaluation is explained in detail by presenting the discovered results. It concludes the results by summarizing and reviewing them by means of a critical discussion.

5.1 Method Selection

Qualitative evaluation and research methods are applied to prove the reliability and trustworthiness of the developed approaches, defined models, and implemented functionalities (of the vocabulary). The concepts and procedures are critically scrutinized according to their general practical applicability. The selected methods according to the Design Evaluation Guideline by Alan et al. [2, p. 86] that are applied and described in detail in the next sections are briefly introduced.

Analytic evaluation. An analytic evaluation of the persistent business vocabulary assumptions by Slawik and Küpper [68] is presented with regard to the current research progress. Therefore, plenty of current literature research papers have been reviewed and several cloud service and cloud storage providers extensively analyzed.

Empirical Evaluation. A preliminary empirical evaluation was carried out, contrasting this contribution to its requirements and evaluating the assumptions made by Slawik and Küpper [68] in conformity with its correctness and accuracy. This was achieved by applying quantitative research methods in form of a survey and qualitative research methods in the form of an semi-structured expert interview.

Experimental Evaluation. The experimental evaluation applies the analyzed and developed persistent cloud storage vocabulary to capture the service descriptions of twenty-one cloud storage services by sixteen different cloud storage providers. It proves the suitability of the developed model and emphasizes its general practical applicability.

5.2 Analytic Broker Vocabulary Evaluation

As the previous section stated, SDLs pave the way to describe services without great effort. However, this depends on the previous modeled vocabulary and its expressivity. So far, some taxonomies for describing cloud services have been developed. Nevertheless, their scope spread among a broad area depends on the focused use case. The following part analytically evaluates the assumptions made in the SDL-NG framework. Current research progress results are examined to identify possible distinctions.
The existing TRESOR broker vocabulary is presented in Slawik and Küpper [68]. Its assumptions are based on the previous work done by Thatmann et al. [74]. The present service vocabulary contains 37 type classes, 31 service properties and 52 type instances capturing fairly all information to describe current cloud services. There has been made an analytic, experimental, and empirical evaluation before, though this contribution aims to support and prove the appropriations. Therefore, various research papers [4, 77, 71, 61, 59, 11, 27, 85, 87, 59, 88, 52, 76, 40, 21] and books [78, 45] have been identified, analyzed, and studied. Nearly all of them depend on the actual infrastructure, hardware, and virtualization architecture features despite annotating security properties. Especially Mather, Kumaraswamy, and Latif [45], Subashini and Kavitha [71], and Zhou et al. [87] focus on current cloud security vulnerabilities and privacy concerns. Zhang, Cheng, and Boutaba [85] rely on the cloud characteristics defined by the NIST in addition to the geo-redundancy features while Jrad, Tao, and Streit [40] indicate to use hard measurable non-functional criteria including performance and monitoring aspects. Besides those hard measurable criteria, all mentioned service aspects can be described with the current implementation.

Besides the characteristics identified in the papers mentioned above, two available cloud taxonomies could be determined. Höfer and Karagiannis [36] describe their cloud taxonomy with nine common characteristics. Furthermore, they identified service-model-specific characteristics and cloud performance attributes. The design of the taxonomy is modeled as a tree structure with separate taxonomy levels. All characteristics except the performance measurements are fully covered by the SDL-NG. The other taxonomy design by Rimal, Choi, and Lumb [60, p. 24] can be nearly regarded equal with the one by the TRESOR project as both approaches split the identified properties in similar categories. Moreover, though, Rimal, Choi, and Lumb [60] put a stronger emphasis on performance and management.

**Summary.** The concept by Slawik and Küpper [68] and Thatmann et al. [74] can be regarded beneficial as it clearly stands out from the crowd. The vocabulary has been validated with the state-of-the-art in research and proven correct, furthermore no gaps could be identified. The evaluation has proven that the assumptions of the broker system still comply with the current research progress.

### 5.3 Empirical Persistent Vocabulary Evaluation

**Survey.** To evaluate the designed persistent cloud service description vocabulary, including the new modeled cloud storage approach, a LimeSurvey questionnaire was created. The main goal was to identify the relevance, usability, and suitability of the design approach in daily life. Therefore, a clear structure was developed to guide the participant through the complete survey with a red thread. After consultation with the personal mentor, the survey was send to more than fifty fellow students via email, an institute (Fachgebiet Grundbau und Bodenmechanik) at the Technische Universität Berlin with around twenty employees, and some small local business companies. On the other hand, a request to send this survey to all students of the faculty four was submitted to the “Projekt- und Datenmanagement” (PDM), however, no response was received. Altogether, thirty-five users had participated in this survey. Of these thirty-five participants, only eighteen have completed the entire survey. That is why the following results and statements can be regarded as hypotheses only.

The participants were asked about their experience and main purpose with cloud storage services in general. Except for one user all participants have used a cloud storage service. However, roughly half of the participants are using cloud storage services very often, their main purpose is to share and sync data. In the rest of the survey, the participants had to state their personal relevance on a five point scale from irrelevant up to indispensable according to all available service vocabulary properties. The results for persistent service description vocabulary developed by Slawik and Küpper [68] are very sparse and diverse. Only for the topic security there is a clear tendency visible: as previously stated in the analysis and trends, participants have a high demand for security. Besides that, the distribution of the given answer for the rest of the properties is nearly homogeneous. Possible causes will be discussed in Section 5.5.

In contrast, the results of the newly developed cloud storage vocabulary characterize a clear direction. Besides the properties development features and development finely granular assignment of rights, all other properties have been identified as very important or indispensable. There were nearly no submitted responses with the value no answer. This indicates that questions or uncertainties are non-existent. Nevertheless, because of the small quantity of responses this is only an assumption. Further research and validation is necessary.

In conclusion, this survey was intended to reach a broad audience, however only a small group of users finished the evaluation survey completely. Potential issues together with improvement opportunities will be discussed in Section 5.5.

5.4 Experimental Persistent Vocabulary Evaluation

This section provides a technical comparison of several cloud storage providers. In total, twenty-one different cloud storage services from sixteen providers have been identified and analyzed using the persistent cloud storage vocabulary. The resulting service descriptions are integrated into the SDL-NG source repository and prove the pertinence of the defined storage

model. No challenges have been encountered in the process of describing services with the newly created cloud storage vocabulary. A detailed table-based overview of all characteristics of the studied services can be found in the service descriptions in the SDL-NG framework. The following services have been identified, studied and are briefly outlined now:


[AS3, Box, C42, MA, Mozy, RS, TD] are mainly focused on enterprise customers, however, everybody is able to use this service. In contrast, [AS3, MA, RS] are basic storage providers, offering only interfaces and no specific "sync" client. Nevertheless, those three services allow the most customization. Almost all services do provide additional details about their service, but not [CM, Mega]: trying to figure out any information is a mess. Again, the experimental evaluation with those services has proved the suitability of the defined cloud storage model. Except [CM, Mega], service descriptions could easily be implemented.

5.5 Discussion

The evaluations have clearly proven the analyzed and designed artifacts, however, the entirety of participants is small and not necessarily representative of the broader population. All these results are preliminary and have to be treated as assumptions. It is difficult to know how far the results can be generalized. Additionally, the results can be biased of both the participants and the researcher’s own opinion – this has not been covered so far.

In general, there can be several lack of rigours: direct feedback by participants remarked that the content of the survey will maybe fit for the enterprise and academic area of research but not for private consumers. This could be one reason of dropout at an early stage of the survey. The survey did not measured the target group, it could have been helpful to knew the type of consumer (private, SMBs, enterprise). The general performance of cloud storage offerings could not been measured and linked with the problem of the client bandwidth. Further information about this field of research are presented by Zhang, Feng, and Qin [86] and Drago et al. [22].

So far, no study has covered the question yet whether experts and consumers require the same properties for cloud service selection. In regard to the survey, it more or less looks like private consumers needs conflicting with those of experts. Another uncovered area is the categorization of cloud services. Are there different types of service classes? How would they look like?
6 Summary & Outlook

In this thesis, cloud computing benefits, concerns, and lacks have been analyzed and identified. Several new approaches towards an Open Service Compendium have been provided and their suitability has been proved.

First, the relevance and impact of cloud computing in today’s world and the near future has been outlined. The specific application environment was introduced and separated from other related technologies.

Then, a domain-specific cloud storage service taxonomy has been designed based on an analysis of the factors relevant in this field. So far, the standardization of cloud services is still ongoing, however, the overall adoption is rare. In this context, the question which requirements cloud consumers require to be able to select services has been researched by evaluating a survey regarding the needs of users of cloud services.

An implementation of new broker technologies extended the existing TRESOR system with new functionalities was provided. In this regard, mechanisms to improve service selection were proposed. These mechanisms provide common service selection algorithms to simplify the decision-making process and a comparison overview to support it.

Finally, various evaluation methods surveyed the results from the analysis and the implementation. The previously developed and the already existing service description vocabulary has been proved to be generally applicable.

6.1 Recommendations

The cloud computing paradigm is a tool to create, enable, and benefit opportunities. Almost always, these opportunities are followed by hidden, inconspicuous obstacles. Marston et al. [44] outlines risks and benefits on cloud migration. The movement to the cloud should be planned and tested in detail in advance. Not every application is suitable for cloud usage due to costs, regulations, restrictions, and compatibility challenges.

Apart from that, cloud providers are praising their unattainable availability but reality is far from it. Time and again, recent news[1] demonstrate outages lasting several hours. As clouds availability should be ensured by no single-point-of-failure, most of the time the provider is the point-of-failure (Armbrust et al. [4]). Furthermore, even the worst-case scenario has to

be planned: over the past years, two big cloud providers (Nirvanix\(^2\) and Nebula\(^3\)) suddenly shut down.

A further ignored point, especially by individuals and SMBs, is data transfer bottleneck. The current amount of digital data grows incredibly but the clients bandwidth is still not capable enough to handle the demand. Bergen, Coady, and McGeer \([6]\) survey this issue by providing a benchmark on Amazon S3. Transmitting enormous data sets into the cloud is still a challenge where further intensive research is required.

When thinking about moving to the cloud, it is recommended to analyze the use case, identify possible difficulties and compare potential providers. Li et al. \([42]\) provide a good audit of cloud providers while Harms and Yamartino \([34]\) introduce the economical view on cloud computing taking availability, variability, overall impact and implications into account. A good overview of parameters that have to be taken into account are presented by Armbrust et al. \([4]\) and \([78], p. 307ff.\].

### 6.2 Future Work

Until now, the fundamental steps and concepts have been already established. This contribution extended the existing system with new functionality and supported the path to the *Open Service Compendium* – an open crowd-sourced platform for easily describing, selecting and comparing every kind of (cloud computing) service. Nevertheless, there is space for improvement and several open challenges have been identified:

**Performance & Monitoring Measurement.** In addition to Emeakaroha et al. \([24]\), the compliance of current Service Level Agreements (SLAs) and Quality of Service (QoS) is difficult to prove. Yet, no provider offers a possibility to measure the availability, bandwidth, response time, and delay of the consumed service. The broker system should provide internal benchmarks and historical data.

**Recommendation & Feedback.** The service selection and description registry could be enhanced to give recommendations based on additional customer information, preferences or past purchases. Moreover, the service matching algorithm could be expanded to recommend compatible services. Complementary, a rating module could support decision-making with provider feedback, including customer satisfaction or mean indicators (e.g., in combination with the performance and monitoring extension).

**Property Selection Model.** Currently, all service properties are weighted equally. However, there should be differentiation according to the importance of each property matching the users requirements. On the other side, an indicator should provide the user with the information whether the compared property values are higher or lower. One existing approach around this topic is presented in Rehman, Hussain, and Hussain \([57]\).

**Automated Price Calculation.** The latest version of the broker system enables the functionality to describe and compare existing services. Still, there is no option available to calculate the price based on the desired configuration automatically. Wang et al. \([81]\) explores the capabilities related with this concept besides revealing insights.

\(^2\)http://www.wired.com/2013/10/nirvanix-bankrupt/
\(^3\)http://www.techweekeurope.co.uk/cloud/cloud-management/nebula-openstack-cloud-shuts-down-165846
6.3. Publication of the Source Code

The complete source code of broker extension implementation is accessible at the GitLab service provided by Technische Universität Berlin at [https://gitlab.tubit.tu-berlin.de/faaabs/osc-thesis/tree/master/osc-broker](https://gitlab.tubit.tu-berlin.de/faaabs/osc-thesis/tree/master/osc-broker). To comply with the examination regulations, the source code is also contained on the CD attached to the printed form of this thesis.

LimeSurvey Layout

The LimeSurvey Layout was partly adapted with the Bootstrap Framework\(^4\) for the Evaluation Section. Most parts of the questionnaire are now responsive and the usability has improved a lot. The adapted sources accessible at the GitLab service provided by Technische Universität Berlin at [https://gitlab.tubit.tu-berlin.de/faaabs/limesurvey-bootstrap](https://gitlab.tubit.tu-berlin.de/faaabs/limesurvey-bootstrap).

---

\(^4\) Bootstrap Framework v3.3.2; URL: [http://getbootstrap.com](http://getbootstrap.com)
Currently, cloud computing is the state-of-the-art and present in every day life. Everybody can easily benefit by its advantages, but very often, its disadvantages are neglected or even ignored. The requirements have to be considered before moving or migrating to the cloud to avoid critical challenges.

This thesis identified that current cloud service approaches are primarily superficial. There is a huge marketplace with a variety of several cloud service solutions, however, finding suitable cloud services is still a problem. The Open Service Compendium is facing this issue by offering support to cloud consumers in their service selection. Therefore, a rigorously designed formal cloud storage taxonomy has been developed and integrated into the SDL-NG repository. An extension handling the complexity and diversity has been developed. Cloud users are now able to compare various cloud services without any extra effort.

However, there is still room for improvement as described in Section 6.1 and 6.2. The lack of general standards and the rapid changing development progress of technologies needs to be considered. On the other hand, further research has to been done on identifying potential service classifications.

In contrast, current cloud approaches are not utilizing their capabilities sufficiently enough. Its transformative potentials in business has been underestimated so far. Cloud services have the potential to transform daily routines into entirely new applications that have not been imaginable before.
List of Figures

2.1 Google Trend: Cloud Computing & Cloud Storage ........................................ 10
3.1 Persistent Cloud Storage Vocabulary ............................................................. 17
4.1 Implemented Faceted Search functionality with current broker vocabulary ........ 24
4.2 Filtered broker services regarding the selected criteria ................................. 25
4.3 Implemented table-based broker service comparison ...................................... 26
Bibliography


