MÄLARDALEN UNIVERSITY SWEDEN

School of Innovation, Design and Engineering

Architecting for the Cloud

Master Thesis in Computer Science

Student

Ivan Balatinac

ibc14001@student.mdh.se

Iva Radošević

irc14001@student.mdh.se

Supervisor

Hongyu Pei-Breivold

hongyu.pei-breivold@se.abb.com

Examiner

Ivica Crnković

ivica.crnkovic@mdh.se

May, 2014

Abstract

Cloud Computing is an emerging new computing paradigm which is developed out of service-orientation, grid computing, parallel computing, utility computing, autonomic computing, and virtualization paradigms. Both industry and academia have experienced its rapid growth and are exploring full usage of its potentials to maintain their services provided to customers and partners. In this context, a key aspect to investigate is how to architect or design cloud-based application that meet various system requirements of customers' needs. In this thesis, we have applied the systematic literature review method to explore the main concerns when architecting for the cloud. We have identified, classified, and extracted existing approaches and solutions for specific concerns based on the existing research articles that focus on planning and providing cloud architecture or design for different concerns and needs. The main contribution of the thesis is a catalogued architecture solutions for managing specific concerns when architecting for the cloud.

Table of Contents

List of Figures

Chapter 1 - Introduction

Cloud Computing has emerged as one of the most important new computing strategies in the enterprise. A combination of technologies and processes has led to a revolution in the way that computing is developed and delivered to end user [3]. Cloud computing is defined by National Institute of Standards and Technology (NIST) [S72] as: “a model for enabling 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”.

The cloud computing paradigm enhances agility, scalability, and availability for end users and enterprises [11]. Cloud Computing provides optimized and efficient computing platform, and reduces hardware and software investment cost, as well as carbon footprint [1]. For example, Netflix, as it began to outgrow its data centre capabilities, made a decision to migrate its website and streaming service from a traditional data centre implementation to a cloud environment. This step allowed the company to grow and expand customer base without building and supporting data centre footprint to meet its growth requirements [11].

With the rapid development of everyday life and rapid increase of internet traffic, both industry and academia are searching for help to maintain their services. The statistics of “Your Digital Space“ [15] have stated that we are sending nearly 3 million emails per second, in one minute we are uploading 20 hours of videos to YouTube , Google processes 24 petabytes of information, publishing 50 million tweets per day, nearly 73 products are ordered on Amazon for every second. For another example, Facebook experienced a growth of 1382% in one month (Feb-March 06) [6]. According to IDC (International Data Corporation) research conducted in 2013 [16], globally spending on public IT cloud services will reach $47.4 billion in 2013 and is expected to be more than $107 billion in 2017. As this being said, it is important to choose appropriate cloud architecture solution to cope with different needs.

1.1. Advantages of Cloud Computing

Cloud Computing has many advantages [11], some examples are:

- Masked complexity - upgrades and maintenance of the product or service can be hidden from users, without having them to participate;
- Cost flexibility - with the cloud computing there is no need to pay dedicated software license fees, or to fund the building of hardware and installing software;
- Scalability - cloud enables enterprises to add computing resources at the time they are needed;
- Adaptability - cloud computing helps enterprises to adapt to various user groups with a various assortment of devices;
- Ecosystem connectivity – cloud facilitates external collaboration with consumers and partners which leads to improvements in productivity and increased innovation;

In the traditional computing, lessons learned from one environment must be duplicated in other environments, but in Cloud Computing improving some parts are valid for all consumers. Cloud Computing resources can be scale up and down automatically, but in traditional computing, human intervention is needed for adding hardware and software. Cloud Computing environments are usually virtualized, whereas traditional environments are mostly physical. Figure 1.1 shows comparison between traditional computing and cloud computing.

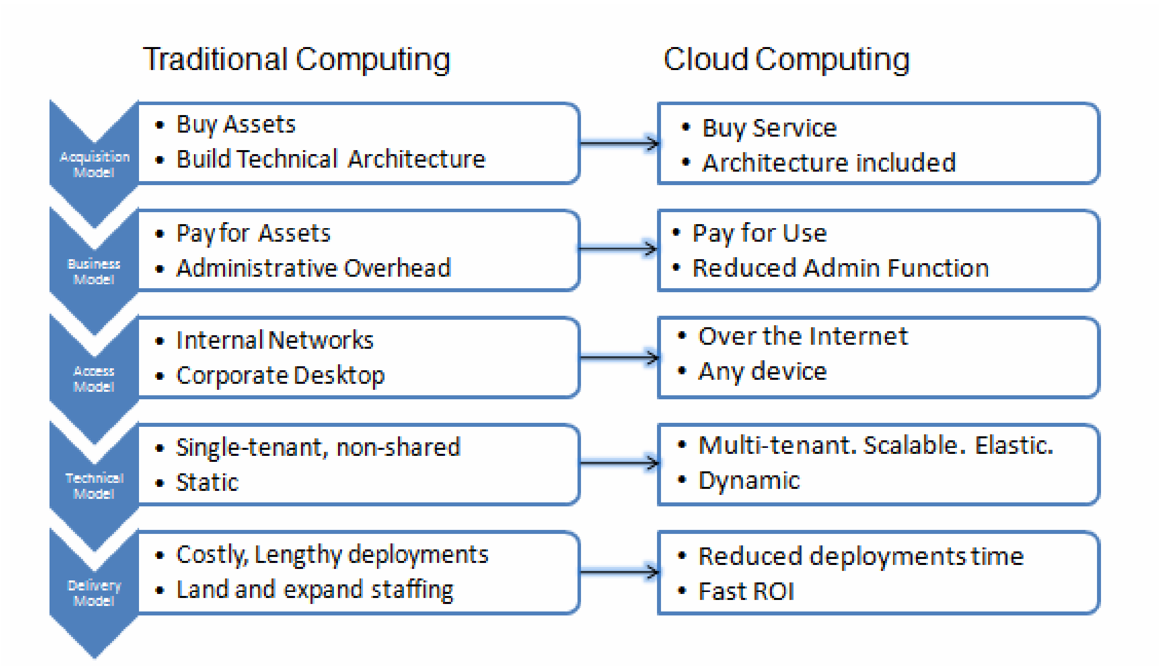
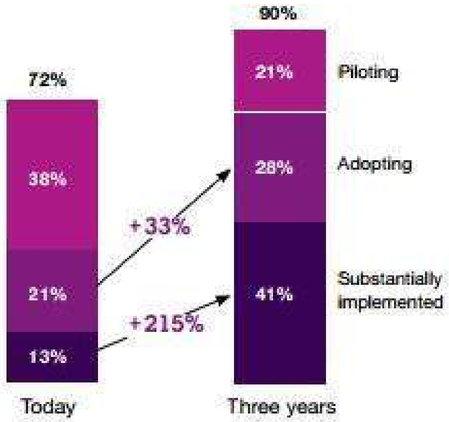


Figure 1.1 Traditional computing and cloud computing [17]

Cloud computing is changing the services consumption and delivery platform as well as the way businesses and users interact with IT resources. There is a growing interest of cloud computing topic within industry. In 2008, IEEE Transaction on Services Computing adopted Cloud Computing to be included in taxonomy as a body of knowledge area of Computing Services [19]. In 2012, European Commission outlined a European Cloud Computing strategy to promote the rapid adoption of Cloud Computing in all sectors of the economy, because of that; they funded research activities in Cloud Computing such as: REMICS [25], CLOUDMIG [26], ARTIST [27] etc. Many organizations have begun wither shifting to the Cloud Computing model or evaluating such a transition. In 2011, IBM in conjunction with the Economist Intelligence Unit conducted a survey which involved 572 business and technology executives across the globe in order to determine how organizations use Cloud Computing today, and how(or if) they plan to use its power in the future [11]. As shown in Figure 1.2, almost 75% of organizations had piloted, adopted or substantially implemented cloud in their organizations (and the rest expect to have done so in three years).



Source: 2011 IBM Institute for Business Value/Economist Intelligence Unit Cloud-Enabled Business Model Survey.

Figure 1.2 Results of IBM 2011 survey about the Cloud Computing adoption in organizations [11]

The survey also showed that cloud adoption is not limited to large companies, 67% of companies with revenues less than US$1 billion and 76% of those with revenues between US$1 and 20 billion have adopted cloud at some level. When it comes to the quality attributes, more than 31% of executives answered that cost flexibility is a key reason for considering cloud adoption. After cost flexibility, come security, scalability, adaptability and masked complexity.

1.2. Application Areas of Cloud Computing

Today's industrial systems are characterized by a strong dependency on comprehensive IT infrastructure at the customer's site. In the whole lifecycle of such systems, the costs for the IT hardware, infrastructure, and maintenance are high. Cloud computing provides a new way of delivering industrial software and providing services to customers on demand. There are major opportunities for industry in terms of providing cloud services, which in turn increase competitiveness by providing cutting edge cloud solutions for interacting with and controlling complex industrial systems.

Some examples of Cloud Computing in the application domain include:

- Online email: or web-based e-mail is any e-mail client implemented as a web application and accessed via the Internet. (developed by e.g. Microsoft, Yahoo, Google)

- Online storage services: offer services of storing electronic data with a third party services and accessed via the Internet. (e.g. Humyo, ZumoDrive, Microsoft's SkyDrive)
- Online collaboration tools: "refer to Web, social and software tools used to facilitate website customer communication for increased sales and satisfaction on the Internet in real time" [14]. (e.g. Google Wave, Spicebird, Mikogo, Stixy)
- Online office suite: refer to a collection of programs implemented as web applications that are used to automate common office tasks. (e.g. Google Drive, Ajax13, ThinkFree, Microsoft Office Live)

1.3. Challenges of Cloud Computing and Thesis Motivation

"Building new services in the cloud or even adopting cloud computing into existing business context, in general is a complex decision involving many factors. Enterprises and organizations have to make their choices related to services and deployment models as well as to adjust their operational procedures into a cloud oriented scheme combined with a comprehensive risk assessment practice resulting from their needs [S65]". Generally, cloud itself cannot experience failure; it is the service that goes down. For example, when Gmail had an interruption for 30 hours on the 16th of October, 2008, it was not a cloud failure; it was a service failure [S17]. "Outage is the most critical issue that is making news in the cloud computing area. It refers to the non-availability of service in the cloud over a particular time [S32] ". For example, in 2009, Microsoft Sidekick outage resulted in a loss of the users data (millions of users lost their data that were stored in the cloud) [S32]. Table 1.1, shows some of the outages in cloud by study [S17].

Service	Duration/Days/Hours/Minutes	Date
Facebook	1 h	Aug 10, 2011
Amazon	11 h	Apr 21, 2011
Foursquare	4 h	Aug 9,10,25, 2011
Microsoft Sidekick	6 d	Mar 13,2009
Google Gmail	30 h	Oct 16, 2008
Google Mail, Google Apps	24 h	Aug 15, 2008

Table 1.1, Some of the outages in the cloud

In April, 2011, Amazon elastic compute cloud (EC2) experienced service disruption because of the incorrect network change performed few days before the outage occurred [20]. The network change supposed to be a regular test of scalability. Amazons service was unable to read and write operations. Microsoft Sidekick experienced a massive outage [21] in 2009, which left his customers without access to their services. The data loss resulted from a system failure, this happened because of the lack of Microsoft disaster recovery policy. In 2008, Google Gmail experienced two outages. The problem was connected with availability concern. Since then, Google Apps offer a premiere edition for $50, in which customer gets 24*7 phone and email support in order to be able to access Google services at anytime (even in case of outages). Foursquare, in 2011, experienced several outages [22] and their service was unavailable to the customers. This happened because of the lack of scalability and their servers could not manage to scale enough so they crashed. Facebook also experienced an outage; customers were unable to log in. It was explained due to the site's experimental features which were being tested at the moment of the outage [23].

Although cloud computing is gaining more and more influence in information industry, adoption is going slower than expected [S32] as – cloud computing poses new challenges to evolving software intensive systems. For instance, executives are more likely to trust existing internal systems over cloud based systems because of the security concerns, e.g., loss of control of data and system outages in the cloud computing systems. The motivation of the thesis is thus to investigate:

(i) the main challenges and concerns of when designing cloud-based solutions and building cloud-based architectures; and

(ii) different architectural approaches and design considerations to meet specific embedded system requirements in terms of e.g., availability, performance, reliability, scalability, etc.

In this thesis, we are going to investigate the existing cloud architectural approaches that are ready for enterprises to implement. The question how to design an application for the cloud will be our main research question and we will conduct a systematic literature review which will help us on finding the answer from the full overview of existing studies.

Chapter 2 - Cloud Computing

Cloud Computing term goes back to the 1950s when “server rooms“ were made available to schools and businesses. Multiple users were able to access “server rooms“ by terminals. The term cloud and its graphical symbol have been used for decades in computer network literature, first to refer to the large Asynchronous Transfer Mode (ATM) networks in the 1990s, and then to describe the Internet (a large number of distributed computers) [2]. This chapter presents the key characteristics of Cloud Computing the deployment models and service models, and stakeholders involved.

2.1. Key Characteristics of Cloud Computing

The main characteristics of Cloud Computing [2] are:

- On demand self-service - consumer of the service can automatically request the service based on their needs, without the interaction with the service provider;
- Easy to access standardized mechanisms - it should be always possible to access the service from the Internet, when policies allow this;
- Resource pooling and multi-tenancy - sharing resources between multiple tenants can increase utilization, and reduce the operation cost;
- Rapid elasticity - the ability to scale in and out, provides the flexibility to provision resources on-demand;
- Measured service - monitoring condition of services, measuring services enables optimizing resources;
- Auditability and certifiability - services should provide logs and trails that allow the traceability of policies for ensuring that they are correctly enforced.

2.2. Cloud Computing Deployment Models

There are four main Cloud computing models: public, private, hybrid and community cloud.

- Public cloud: “The cloud infrastructure is made available to the general public or a large industry group and is owned by an organization selling cloud services [S72] “.

Consumers need to pay only for the time duration they use the service, i.e., pay-per-use which helps in reducing costs. They are less secure comparing to other cloud models since all the applications and data are more opened to malicious attacks. Proposed solution to this concern is security validation check on both sides.

- Private cloud: "The cloud infrastructure is operated solely for an organization. It may be managed by the organization or a third party and may exist on premise or off premise [S72] ". It is a data centre owned by a cloud computing provider. "The main advantage is that it is easier to manage security, maintenance and upgrades and also provides more control over the deployment and use. Compared to public cloud where all the resources and applications are managed by the service provider, in private cloud these services are pooled together and made available for the users at the organizational level. The resources and applications are managed by organization itself [12] ".
- Community cloud: "The cloud infrastructure is shared by several organizations and supports a specific community that has shared concerns (e.g., mission, security requirements, policy, and compliance considerations). It may be managed by the organizations or a third party and may exist on premise or off premise [S72] ".
- Hybrid cloud: "The cloud infrastructure is a composition of two or more clouds (private, community, or public) that remain unique entities but are bound together by standardized or proprietary technology that enables data and application portability(e.g., cloud bursting for load-balancing between cloud) [S72]". Hybrid cloud is more secure way to control data and applications and allows the party to access information over the internet.

2.3. Cloud Computing Service Models

There are five main different layers of Cloud Computing Architecture: client, application, platform, infrastructure, and server as shown in Figure 2.1.

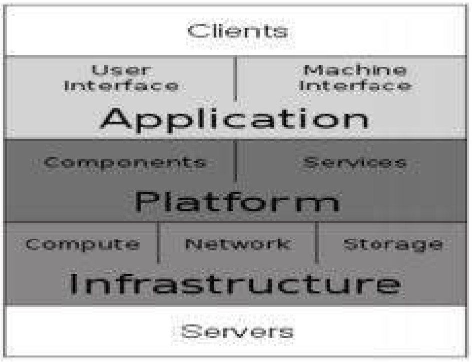


Figure 2.1, Cloud Computing layers [13]

A service model represents a layered high-level abstraction of the main classes of services provided by the Cloud Computing model, and how these layers are connected to each other [2]. The first layer, a cloud client, “consists of computer hardware and/or computer software which relies on cloud computing for application delivery [13] “. Cloud application, platform and infrastructure layer, deliver cloud service models.

- Software as a Service (known as SaaS) is a service which allows the end user (consumer) to access and use a provider software application owned and managed by the provider. Software as a service allows software to be licensed to a user on demand [7]. The consumer does not own a software but rents it, e.g. for a monthly fee.
- Platform as a Service (known as PaaS) is a service hosted in the cloud and accessed by users through their web browser. Platform as a service is a provisioning model that allows the creation of web applications without the need to buying and maintaining expensive infrastructure like hardware and software [7]. PaaS allows users to create software applications using tools given by the provider.
- Infrastructure as a Service (known as IaaS) is a service which allows the service consumer to rent infrastructure capabilities based on demand. Consumers do not have direct access to resources but have the ability to select and configure resources as required based on their needs [2].

The last layer is server which “consists of the characteristics computer hardware and/or software required for the delivery of the above mentioned services [13] “.

2.4. Cloud Computing Stakeholders

Figure 2.2 shows the conceptual view of the Cloud Computing architecture with three main Cloud stakeholders – the provider, consumer and the broker [3].

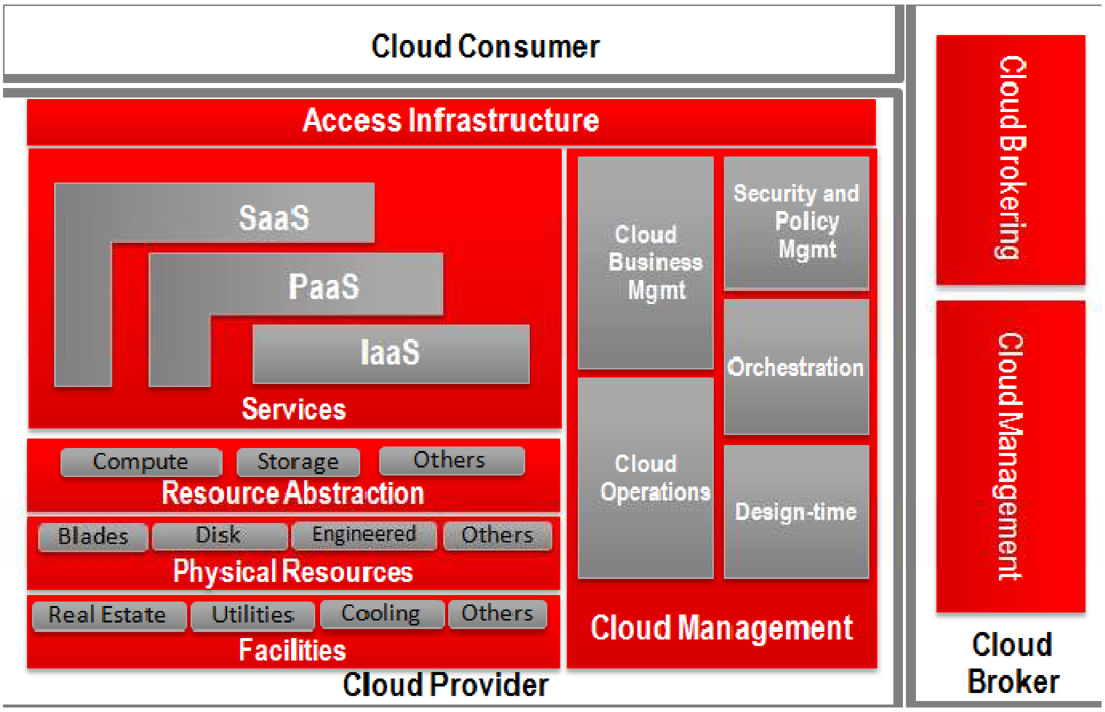


Figure 2.2 Conceptual view of the architecture [3]

- A Cloud provider is a company or an individual that delivers cloud computing based services and solutions to consumers [4].
- A cloud consumer is a company or an individual that uses a cloud service provided by a cloud service provider directly or through a broker.
- A cloud broker is an intercessor between cloud providers and cloud consumers.

The NIST Cloud Computing reference architecture (which will be detailed in sub-chapter 2.5) defines two additional roles: cloud auditor and cloud carrier.

- A cloud auditor “is a party that can perform an independent examination of cloud service controls with the intent to express an opinion thereon. A cloud auditor can evaluate the services provided by a cloud provider in terms of security controls, privacy impact, performance, etc. [S72]”.

- A cloud carrier is an intercessor between cloud consumers and cloud providers which provides connectivity and transport of cloud services. Cloud carriers provide access to consumers.

Figure 2.3 explains interactions between the main roles in Cloud Computing.

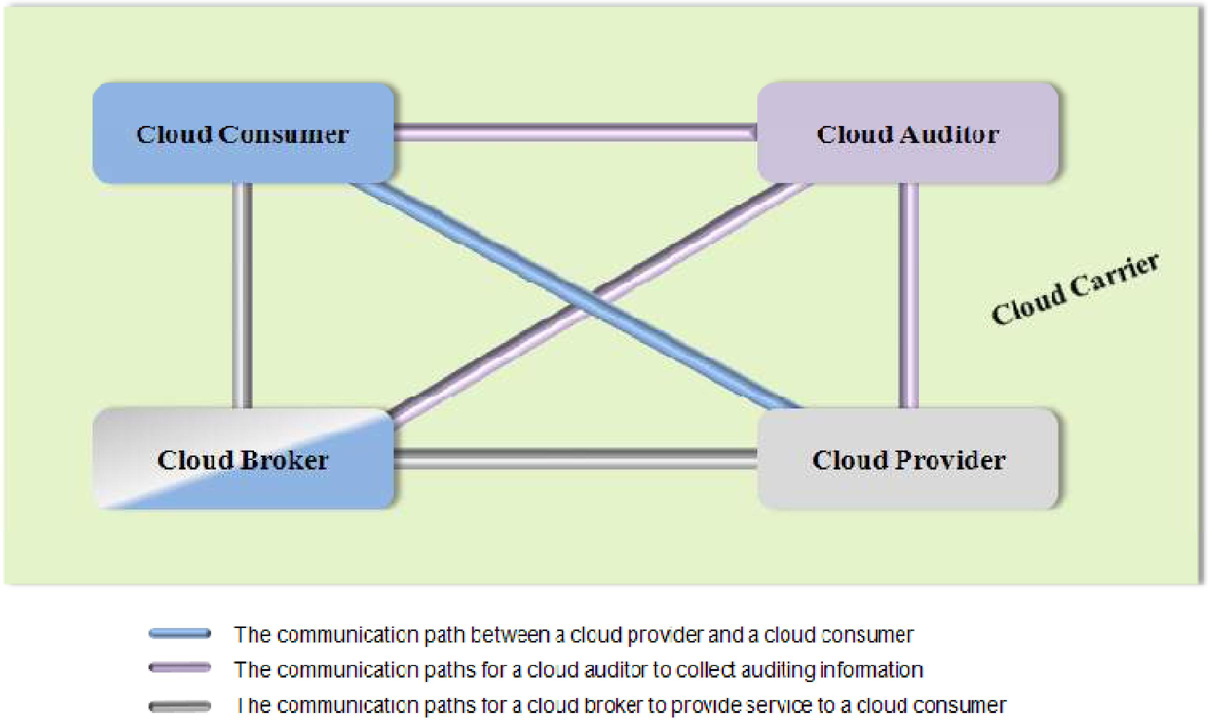

Cloud Consumer
Cloud Auditor
Cloud Carrier
Cloud Broker
Cloud Provider
The communication path between a cloud provider and a cloud consumer
The communication paths for a cloud auditor to collect auditing information
The communication paths for a cloud broker to provide service to a cloud consumer

Figure 2.3, Interactions between the main roles in Cloud Computing [S72]

In this thesis, through systematic literature review we will analyze different types of services known as “Anything“ as a service (XaaS) , where the term “anything“ refers to the different types of services such as storage, privacy.

2.5. Cloud Computing Reference Architecture

Successful Cloud adoption requires guidance around planning and importing Cloud to the existing services and applications. Industry and academy that want to implement Cloud solutions seek for more information about best practices for migrating and adopting a Cloud Computing. “Defining a Cloud Reference Architecture is an essential step towards achieving higher levels of Cloud maturity. Cloud Reference Architecture addresses the concerns of the key stakeholders by defining the architecture capabilities and roadmap aligned with the business goals and architecture vision [3]". Figure 2.4, presents the NIST Cloud Computing reference architecture which defines major actors, activities and functions in Cloud Computing.

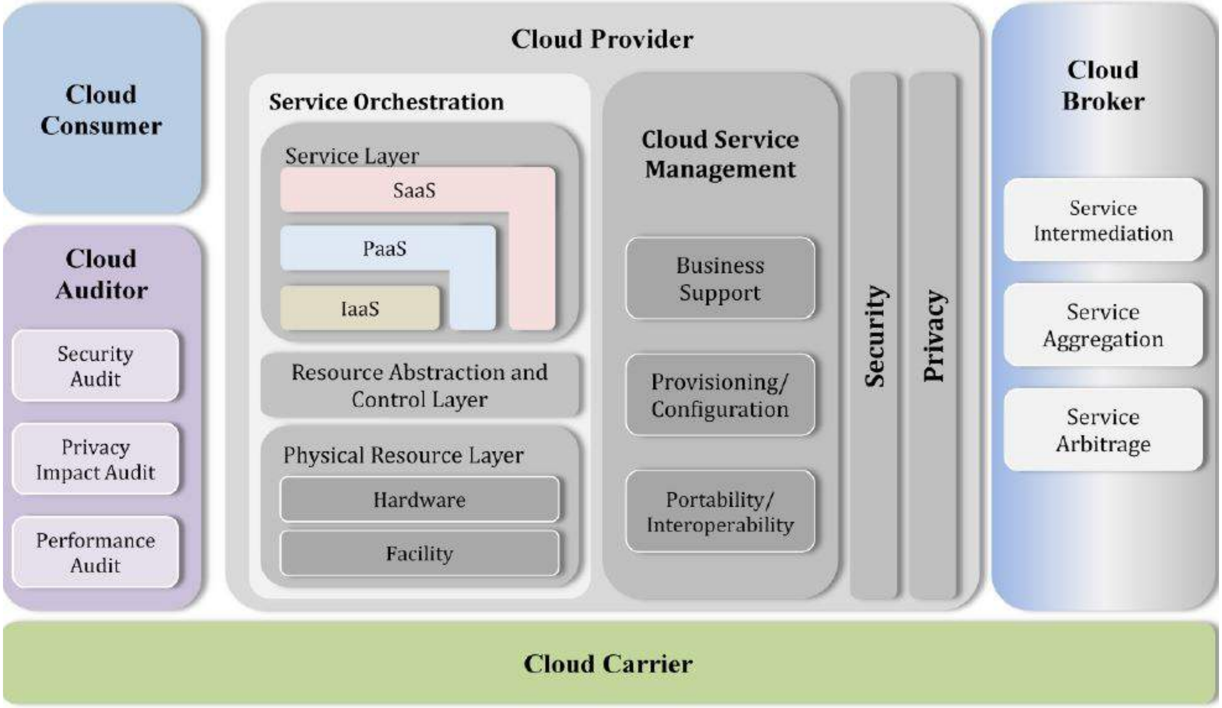


Figure 2.4, NIST Cloud Computing Reference Architecture overview [S72]

Service orchestration refers to the system components for supporting cloud provider in order to provide cloud services to cloud consumers. It consists of service layers (sub-chapter 2.3), resource abstraction and control layer and physical resource layer. Resource abstraction and control layer "contains the system components that cloud providers use to provide and manage access to the physical computing resources through software abstraction [S72]". It needs to ensure efficient, secure and reliable usage of the underlying physical resources. The lowest layer; physical resource layer, includes all the physical computing resources (computers, storage components, networks etc) and facility resources (heating, ventilation, power, etc.). The third level of the presented cloud reference architecture consists of service intermediation, service aggregation and service arbitrage. Service intermediation enhances a given service by cloud broker to one or more service consumers. Service aggregation (similar to service arbitrage) combines multiple services into one or more new services. The difference between service aggregation and service arbitrage is that the service arbitrage provides flexibility for the service aggregator.

2.6. Virtualization, Service Oriented Architecture and Cloud Computing

Virtualization and Cloud Computing

Virtualization is a core technology for enabling cloud resource sharing. It enables abstraction of services and applications from the underlying IT infrastructure [S20, S25, S55]. Study [S25] gives an explanation of key cloud infrastructure evolution phases and architectural enablers for cloud data centres. The second phase [S25] is abstraction, data centre assets are abstracted from the services from which they are provided, enabled by virtualization. There are two basic approaches for enabling virtualization in the Cloud Computing environment [S14]: *hardware virtualization and software virtualization.* "Private clouds hold their own virtualization infrastructure where several virtual machines are hosted to provide service to their clients [S29] ". Studies [S31, S20] introduced the term *server virtualization.* "Server virtualization is the spark that is now driving the transformation of the IT infrastructure from the traditional server-centric computing architecture to a network-centric cloud computing architecture [S31] ". With server virtualization lays the ability of creating logical server independent of the underlying physical infrastructure or their physical location [S31]. There may occur some security problems due to virtualization which are beyond the control of cloud service providers [S55]. In order to solve this security concern, study [S55] proposed the TCSL architecture with the Reliable Migration Protocol.

Service Oriented Architecture (SOA) and Cloud Computing

"SOA is an architectural pattern that guides business solutions to create, organize and reuse its computing components, while cloud computing is a set of enabling technology that services a bigger, more flexible platform for enterprise to build their SOA solutions. SOA and cloud computing will co-exist, complement and support each other [S37] ". Study [S14] combines the power of SOA and virtualization in the context of Cloud Computing ecosystem. Study [S28] proposed Enterprise Cloud Service Architecture (ECSA) as a hybrid cloud architecture for "Enterprise service-oriented architecture (ESOA) which is designed to tackle the complexity and build better architectures and solutions for enterprise [S28]“. Studies [S16, S37] connected SOA and cloud computing and proposed a Service Oriented Cloud Computing Architecture (SOCCA). Study [S27] designed an e-learning ecosystem architecture using cloud infrastructure based on principles of service oriented architecture. They researched two service-oriented cloud computing architectures: *Mandi service-oriented architecture* and *Aneka Platform for Operative Cloud Computing Applications.* However, most existing Cloud Computing platforms have not adopted the service oriented architecture that would make them more flexible, extensible, and reusable [S14].

Chapter 3 - Systematic Literature Review Method

A systematic literature review (also known as a systematic review) is a process of identifying (planning), interpreting (processing) and evaluating (analyzing) of all available research articles connected to a previously defined research question. Individual studies contributing to a systematic review are known as primary studies, what makes a systematic review as a form of secondary study [5]. The need for a systematic review arises from the requirement of researches to gather all existing information about some topic in a thorough and unbiased manner so it has a scientific value. The reason for choosing systematic literature review as a research method for this thesis is to get an overview of the existing research in the field of Cloud Computing architecture and that there was no previous research related to this topic done by performing this research method. Our main goal was to identify, classify, and systematically compare the existing research articles focused on planning and providing cloud architecture or design. As mentioned, in the process of doing a systematic literature we aimed to answer the following main research question:

- How to design an application for the cloud

The following section explains the three main phases which are planning, processing and analyzing.

3.1. Planning

Planning is the first phase of doing a systematic review. It starts by identification of the need for doing the research. This identification is done by investigating previous research, questioning about issues that have not been properly covered (researched), setting up the goals and outcome for a new research.

The most important in the planning phase is specifying the research questions. Properly defined research questions [5] are ones that:

- Are important to practitioners as well as researches;
- Will lead either to changes in current practice or to increased confidence in the value of current practice;
- Will identify deviations between existing beliefs and reality.

In the whole process of planning, it is relevant to have an external expert who will guide through the process and provide with feedback about the research. According to the study [5], it is recommended to do the procedure of selecting studies more than once to reduce bias.

While doing the primary reading and investigating the concept of cloud computing we realized that there are many concerns for adopting cloud computing or migrating to it in both academic or industry level.

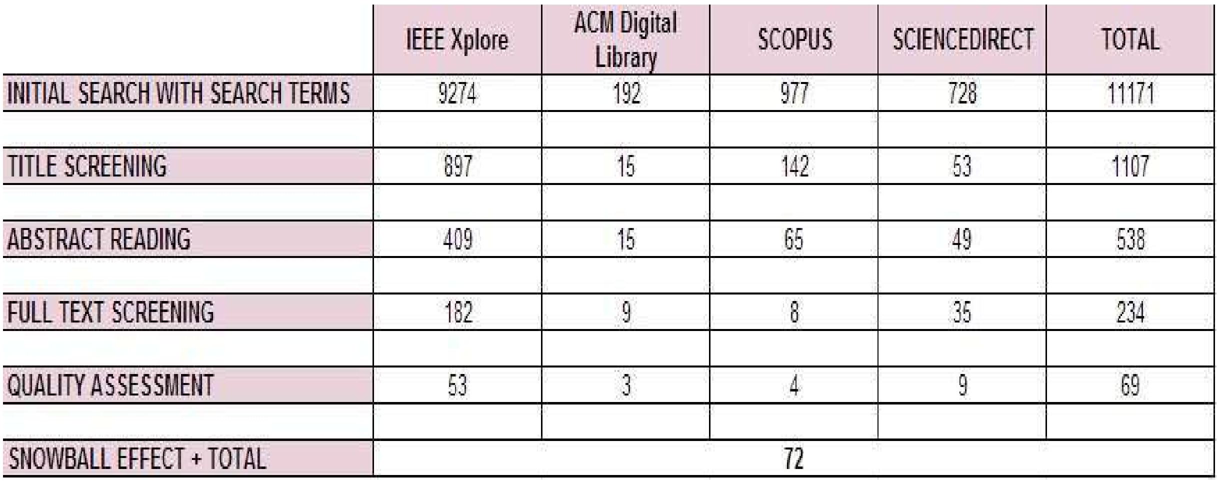
'Cloud computing era' has experienced rapid growth and there still has not been a thoroughly done research that would contain all the information needed and that would tackle all concerns people worry about (while thinking about the cloud computing). Since the field of cloud architecture is really wide we decided to focus on the architecture for cloud based applications that have a pre-defined level of maturity, the main concerns and quality attributes. Pre-defined level of maturity was that the study, if it is proposing architecture; implemented, had an experiment or done a case study with its proposed architecture. To answer our main research question, i.e. How to design an application for the cloud, we needed to answer some sub-questions:

- What are the main concerns in architecting for the cloud?
- What are the existing architectural approaches?

The chosen electronic libraries for conducting a systematic literature review are:

- IEEE Xplore (http://www.ieee.org/web/publications/xplore/)
- ACM Digital Library (http://portal.acm.org)
- ScienceDirect (http://www.sciencedirect.com)
- Scopus (http://www.scopus.com)

In the initial search we used the search terms 'cloud architecture' OR ' cloud architecting' OR ‘cloud design’. In the Table 3.1 is shown how many researched materials were found in each library.



	IEEE Xplore	ACM Digital Library	SCOPUS	SCIENCEDIRECT	TOTAL
INITIAL SEARCH WITH SEARCH TERMS	9274	192	977	728	11171
TITLE SCREENING	897	15	142	53	1107
ABSTRACT READING	409	15	65	49	538
FULL TEXT SCREENING	182	9	8	35	234
QUALITY ASSESSMENT	53	3	4	9	69
SNOWBALL EFFECT + TOTAL	72				

Table 3.1, Complete statistics of systematic literature review phases

3.2. Processing

Processing is the second phase of doing a systematic literature review. It consists of selecting primary (first studies), process of excluding irrelevant materials (and including relevant) and the final selection of the most important articles based on quality assessment.

First step of this phase is the title screening, then the abstract reading. After the abstract reading, additional excluding criteria needed to be defined. The next and final step is full text screening having in mind all the pre-defined excluding criteria. After the step of defining research questions, this is the most difficult part of doing a systematic literature review.

3.2.1. Title screening

Excluding criteria in the phase of title screening was that we excluded everything not connected to our research questions. While doing this part of exclusion we were not concerned on any particular year of publishing, or the region, conference where the article, research material comes from. Also, we included only materials written in English.

The criteria for the inclusion and exclusion materials:

Inclusion criteria:
Articles addressing on the research questions
Articles focused on solving issues concerning quality attributes
Articles explaining the concerns when architecting for the cloud environment
Articles explaining a proposed architecture for coping with a specific concern
Different perspectives regarding cloud architecture
Exclusion criteria:
Articles written in language other than English
Articles not giving the answer to research questions
Duplicated articles
Articles published as whitepapers

Table 3.2. Inclusion and exclusion criteria

3.2.2. Abstract reading

In this step (abstract reading) we were following pre-defined inclusion and exclusion criteria. We used the tool EndNote[1] for all search phases and easier categorization of studies. Exclusion criteria are given in the table 3.2. If in doubt, we decided to leave those articles for the next step of this phase. The next step is the full text screening where it will be more visible whether to include or exclude an article for the final list.

3.2.3. Full text screening

Full text screening is the third step of the second phase of doing a systematic literature review. While doing a full text screening we were following exclusion and inclusion criteria shown in the table 3.2. We excluded irrelevant studies based on analysis of the full text. The following phase is evaluation.

[1] EndNote (www.endnote.com) is a simple programme for sorting references and making different libraries while doing a research.

3.3. Evaluation

Evaluation is the final phase of doing a systematic literature review in which we evaluate (analyze) articles after the full text screening and define our final selected studies. According to the results of a quality assessment and setting up the levels of maturity, we will thoroughly analyze and classify all the given and existing approaches to the topic architecting for the cloud.

It is important to follow guidelines for the each step to reduce chances for mistakes or excluding relevant materials. With the number of 240 articles after the full text screening we needed to do a quality assessment of materials to determine which article should be in the final list and which we could exclude. We took into consideration articles that had a pre-defined level of maturity which led to easier assessment of the maturity in general and future dimensions of cloud architecture.

Quality assessment
Is the motivation of the research paper and its definition clearly presented?
Are the results of the research clearly presented?
Is the research specifically focused on the topic of architecting for the cloud?
If the research paper proposes cloud computing architecture, is it solving the main concerns?
If the research paper proposes cloud computing architecture, is it only an idea or it has already been implemented and tested?

Table 3.3, Quality assessment criteria

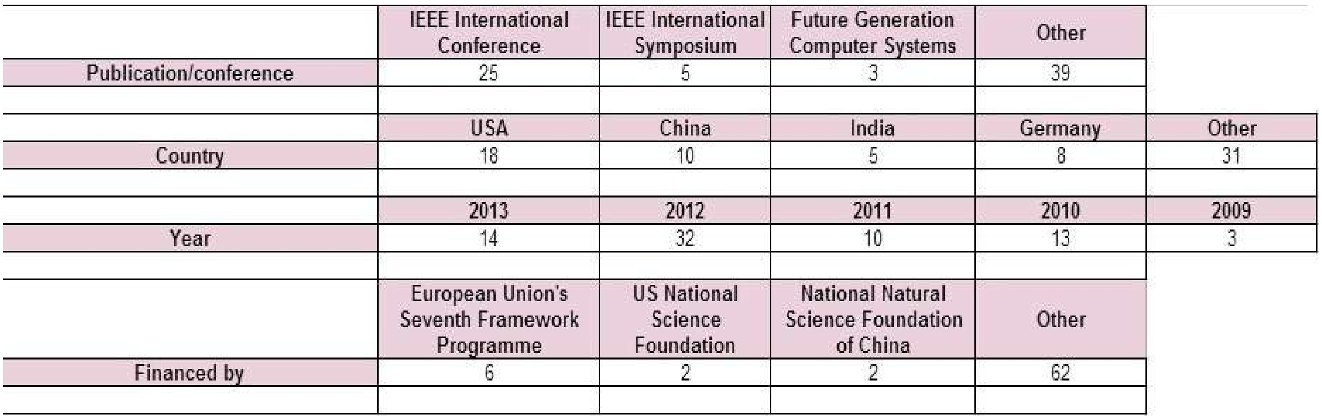
Each article was subject to the quality assessment criteria shown in the table 3.3. After completing the quality assessment process, we got the final number of selected studies (72 studies). The complete list of selected studies is shown in the table 3.1.

Chapter 4 - Analysis

The final list of selected studies is provided in the appendix. This chapter will be organised as follows. General statistics about chosen studies will be shown, main quality attributes will be addressed and the main concerns will be explained, challenges and proposed architectures with will be analysed.

4.1. General Statistics

Table 4.1, shows general statistics about the selected list of studies. It includes: year of publication, if the studies were presented on conference, from which country the studies come from and who financed the research studies.



	IEEE International Conference	IEEE International Symposium	Future Generation Computer Systems	Other	
Publication/conference	25	5	3	39	
	USA	China	India	Germany	Other
Country	18	10	5	8	31
	2013	2012	2011	2010	2009
Year	14	32	10	13	3
	European Union's Seventh Framework Programme	US National Science Foundation	National Natural Science Foundation of China	Other	
Financed by	6	2	2	62	

Table 4.1, General statistics about the selected list of studies

The criteria for doing this statistics was that in the each category were at least two studies conducted. Although the cloud computing concept is not new, the research about the topic has mainly started from the 2009. Also, the statistics are pointing out two leading countries in the cloud computing research area and those are USA and China.

4.2. Quality Attributes

Clouds, as made available by Amazon, Google and 3Tera, use the Software-as-a-service or Infrastructure-as-a-service model. This means that payment for the services of the cloud are made on the basis of CPU-hours used as well as storage used which is more economical than purchasing processors and storage devices. However, cloud providers make no guarantees about the Quality of Service attributes being provided by them [8]. One concern when architecting for the cloud is related to how to design cloud-based architecture that meets different quality attribute requirements. Table 4.2, summarizes the main quality attributes that are addressed by our final list of studies. The studies, included in the table, focus and analyze specific quality attribute concerns. Security, as the main concern, is on top of the list with highest number of studies which are defining different approaches for solving it (discussed in the sub-chapter 4.2.1). The rest of the concerns are as follows: privacy, dependability, interoperability, availability, portability, elasticity and scalability.

Security:
[S4], [S10], [S11], [S18], [S26], [S36], [S39], [S54], [S55], [S56], [S60], [S65]
Privacy:
[S3], [S60], [S65]
Dependability:
[S1], [S39]
Interoperability:
[S30], [S41]
Availability:
[S7]
Portability:
[S1], [S62]
Elasticity:
[S2], [S20], [S53], [S58], [S70]
Scalability:
[S35], [S50], [S57]

Table 4.2, Main quality attributes, list of studies

4.2.1. Security

One main concern for adopting cloud computing architecture is security. “Security, refers to information security, which means protecting information and information systems from unauthorized access, use, disclosure, disruption, modification, or destruction [S72]“. The internet faces 90% of site security problems [9]. With cloud computing platform protection, the site can be secure, but when the illegal attack focuses to a server in the cloud computing platform, limited server capacity and crash may occur. There are several types of concerns with respect to security:

Infrastructure, Infostructure and Metastructure Concerns

Nowadays, when both industry and academia are moving or thinking about moving to cloud-based services, security is high on the list of concerns. “The cloud changes security's role. Security no longer just provides structural boundaries at the infrastructure level. Instead, security is an active participant in a dynamic, fluid environment [S26] “. Security threats defined in the study [S65] are: abuse and nefarious use of cloud computing, insecure interfaces and APIs, malicious insiders, shared technology issues, data loss or leakage, account or service hijacking, unknown risk profile, privileged user access, regulatory compliance, data location, lack of data segregation, lack of recovery, investigate support and long-term viability. According to the study [S26], security concerns have part in both infostructure (applications and data) and metastructure (policy). Infostructure concerns include:

- Service bindings
- Service mediation
- Message and communication encryption
- Message and data integrity
- Malicious usage

Metastructure concerns include:

- Security token changes
- Security policy management
- Policy enforcement points
- Policy decision points
- Message exchange patterns
- Detection services
- Key management processes

To solve such concerns, study [S26] proposed four technology patterns: gateways, monitoring, security token services and policy enforcement points; which help security architects to “address security policy concerns in the metastructure and improve the runtime capabilities in the infostructure”. The four patterns enable visibility into security events, context-specific security tokens, fine-grained access control, and they attack surface reduction.

Resource Sharing Concerns

Studies [S4, S56] explain security concerns with the resource sharing. “An inadequate or unreliable authorization mechanism can significantly increase the risk of unauthorized use of cloud resources and services“[S4]. Therefore, study [S4] defined several authorization requirements in order to build a secure and trusted distributed cloud computing infrastructure:

- Multitenancy and virtualization – lack of authorization mechanisms cause side-channel attacks;
- Decentralized administration – each service model retains administrative control over its resources;
- Secure distributed collaboration – the cloud infrastructure should allow both horizontal and vertical policy interoperation for service delivery in order to support decentralized environment;
- Credential federation – access control policies must support a mechanism to transfer a customer’s credentials across layers to access services and resources;
- Constraint specification – semantic and contextual constraints must be evaluated when determining access to services and resources.

In order to fulfil these authorization requirements, study [S4] proposed three types of collaborations (federated, loosely coupled, and ad hoc).

Infrastructure Management Concerns

Researches of the study [S10] fear that some benefits of cloud computing (e.g. cost efficiency, scalability, improved availability etc.) “come at the price of negative properties such as requiring trust in the provider, reducing control and isolation and impacting security and data protection.“ Cloud Computing is becoming 'a game changer' in today’s enterprise, still there is a number of major issues which need to be resolved before enterprise could adapt it as a secure infrastructure. The security management of the infrastructure offered by most of the cloud providers is not so well developed and lacks some form of disaster recovery so far [S39, S1]. Though some providers do offer secure cloud solutions, users are forced to redesign and adapt their applications in order to fit to the provided environment [S54]. Study [S11] investigated that individuals, corporations and governments feel unsecure about storing their data on the same server as their competitors or adversaries. Solution they proposed lies in building a dynamic access control which allows the system to have built-in intelligence to comply with current policy, investigate the requesting entity's identity and location based on the current threat level while logging that activity and marking the metadata.

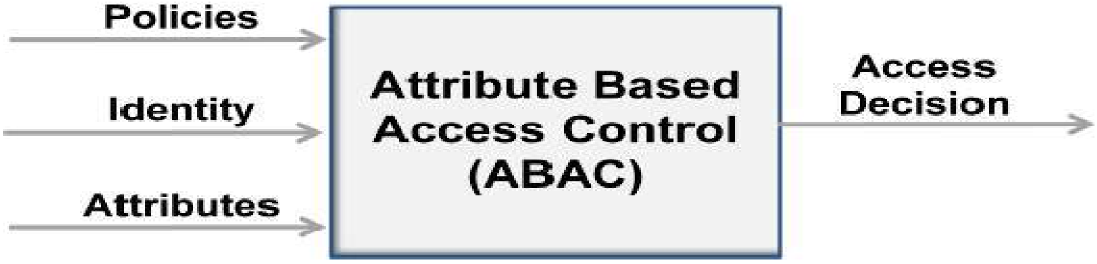


Figure 4.1, Achieving Secure Identity and Access Management through ABAC [S11]

Security Attacks

Researches in the study [S18] defined Intrusion Detection System (IDS) as the best solution to protect the cloud from security attacks such as: resource attacks against cloud service, resource attacks against service provider, data attacks against cloud provider, data attacks against service provider and data attacks against service user. Resource attacks regard the misuse of resources. "Intrusion detection system is the process of monitoring the events occurring in a computer system or network and analyzing them for signs of violations of computer security policies, acceptable use policies, or standard security practices [S18]". Study [S18] uses IDS for monitoring the events occurring in a computer system and analyzing them for signs of possible accidents.

Security Issues in the Mobile Internet Domain

Study [S36] classifies security issues in the mobile internet domain. It says that "introducing cloud computing into mobile internet leads to the changing of mobile internet's architecture, and arises many new security problems such as cross-domain data security and privacy protection, virtual running environment security and cross-domain security monitor". Cloud Computing can be found also in online commerce, main disadvantages are concerns about data privacy and security and the dependency on connectivity [S60]. Study [S65] defined requirements for a cloud based system development methodology to support security analysis:

- It should include concepts from both cloud and organization areas such as dependencies, infrastructure, information management, portability;
- It should provide techniques to select appropriate cloud deployment models to support organizational needs, requirements and it should address the identified threats and risks;
- It should enable the usage of a defined set of concepts and notations during the analysis and design process;
- It should allow developers to evaluate cloud providers.

4.2.2. Information privacy

"Information privacy is the assured, proper, and consistent collection, processing, communication, use and disposition of personal information and personally identifiable information throughout its life cycle [S72]". Privacy issues within its definition are similar to security issues. However, there is justified concern that comes from users when it comes to privacy. Today exists an increasing awareness for the need for designing privacy from both companies and governmental organizations [S3]. In 2007, the cloud service provider: Salesforce.com, sent a letter to a million subscribers describing how customer e-mails and addresses had been stolen by cybercriminals [S3]. Therefore, privacy should be built into every stage of the development process of the cloud. Privacy risks for cloud computing stakeholders include [S3]:

- Cloud service user – being forced or persuaded to be tracked or give personal information against their will;
- Organization using the cloud service – loss of reputation and credibility;
- Implementers of cloud platforms – exposure of sensitive information stored on the platforms, legal liability, lack of user trust;
- Providers of application – loss of reputation, legal non compliance;
- The data subject – exposure of personal information.

In order to solve this mentioned privacy risks, study [S3] proposed different phases of design cloud based application:

- Initiation – setting high level recommendations;
- Planning – describing privacy requirements in detail;
- Execution – identifying problems relating to the privacy solutions, if necessary considering alternative solutions and documenting issues;
- Closure – using change control methods in the production environment, privacy protection during backup, disaster recovery;
- Decommission – ensuring secure deletion of personal and sensitive information.

4.2.3. Dependability

Dependability as a Cloud Computing quality attribute enables users to rely on cloud computing as an external source for their enterprise and as processing facilities for creating their business on top [S1]. European Commission, in the period of 2010-2013, funded five projects: Cloud-TM [28], Contrail [29], mOSAIC [S62], TClouds [S39] and VISION Cloud [30]. All projects deal with dependability concerns in cloud computing considering different application scenarios [S1].

Cloud-TM

The Cloud-TM platform (Figure 4.2) is formed by two main components which are the Data Platform: storing, retrieving and manipulating data across a dynamic set of distributed nodes; and the Autonomic Manager: automating the elastic scaling of the Data Platform. The Cloud-TM project developed a self-optimizing Distributed Transactional Memory middleware in order to help cloud computing programmers to focus on delivering differentiating business value. For achieving optimal efficiency with any workload at any scale, the Cloud-TM middleware integrated autonomic mechanisms with the purpose of automating resource provisioning and self-tuning the various layers of the platform.

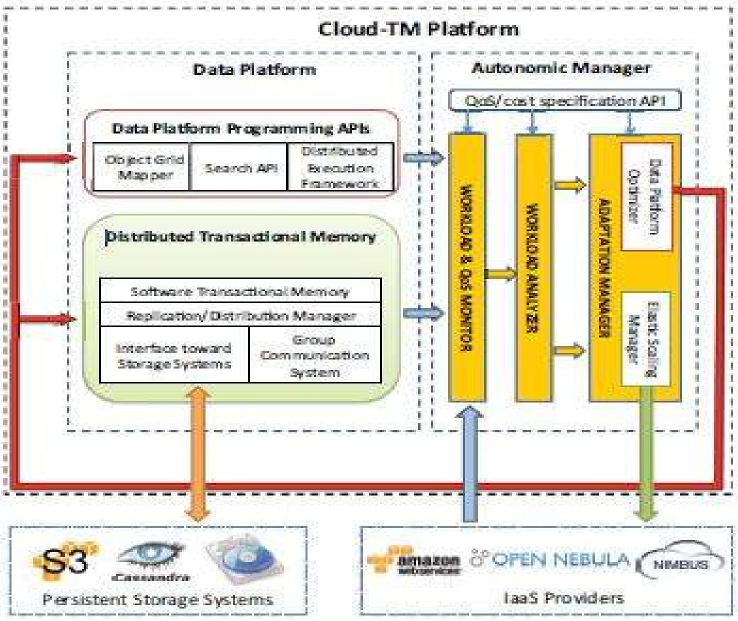

Cloud-TM Platform
Data Platform
Autonomic Manager
Data Platform Programming APIs
Object Grid Mapper
Search API
Distributed Execution Framework
QoS/cost specification API
Distributed Transactional Memory
Software Transactional Memory
Replication/Distribution Manager
Interface toward Storage Systems
Group Communication System
WORKLOAD & QoS MONITOR
WORKLOAD ANALYZER
ADAPTATION MANAGER
Data Platform Optimizer
Elastic Scaling Manager
S3
Cassandra
Persistent Storage Systems
amazon webservices
OPEN NEBULA
NIMBUS
IaaS Providers

Figure 4.2, The Cloud-TM architecture [S1]

Contrail

In Contrail project, user does not have to manage the access to individual cloud providers and can focus on specifying the service or application. “Contrail implements a dependable cloud by guaranteeing the availability of the computational resources and having strict guarantees in terms of quality of service and quality of protection, that customers can specify in the service level agreement, when submitting their requests, and monitor during the execution of the application [S1]”. The Contrail architecture (Figure 4.3) is designed to be extensible, allowing the reuse of some components in different layers and giving the possibility for exploiting components independently.

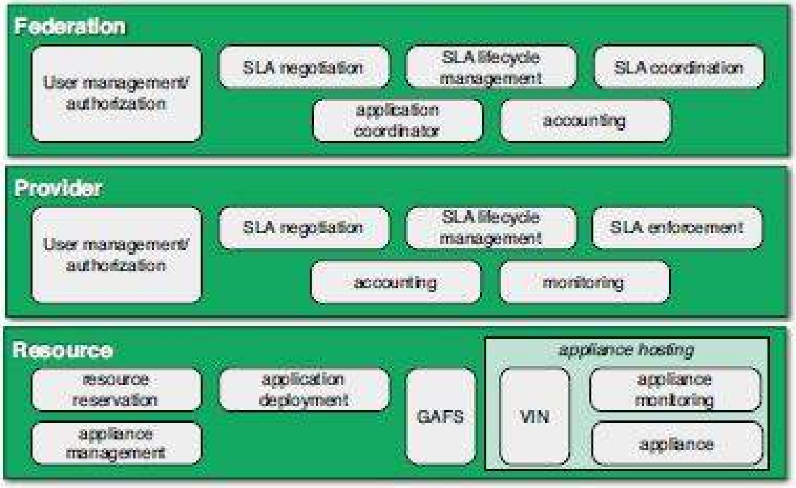


Figure 4.3, Contrail architecture [S1]

mOSAIC

Project mOSAIC [S62] designed a set of open APIs (Figure 4.32) that: introduce new level of abstractions in order to make Cloud computing infrastructure programmable and use an event-driven approach and message passing. In order to increase dependability, mOSAIC solved concerns such as fault tolerance, availability, reliability, safety, and maintainability. To ensure the fault tolerance, the application components are communicating only through message queues presented as Cloud resources. In case of faults in one component instance the messages are redirected to another instance of the same component. To ensure the maintainability, mOSAIC designed that components of the application can be stopped and restarted during the application execution. It is made possible by discovery services which are part of the platform. To ensure the reliability and safety of the application, event-based programs are designed. Event-based programs tend to use one thread to process events, avoiding problems. Ensuring availability is explained in the next sub-section.

TClouds

TClouds [S39], in order to solve dependability concern, considered a general reference architecture that can be instantiated in different ways. Different solutions are required at different levels, depending on the applications requirements. Because of that, the project TClouds (Figure

4.37) only provides a set of tools and methods that need to be adapted for specific application scenarios. This project is further explained in the sub-chapter 4.3.13.

VISION Cloud

VISION Cloud provides an advanced storage cloud solution which solves limitations such as data lock-in, separation between compute and storage resources and security. In order to ensure dependability, Service Level Agreement management has a central role. Contrary to existing commercial offerings, in VISION Cloud, a tenant is able to define different requirements to the platform such as: latency, durability levels, availability, geographic preference, geographic exclusion, security.

4.2.4. Availability

Availability is a quality attribute which means "ensuring timely and reliable access to and use of information [S72] ". With the increasing development of cloud computing architecture, application availability becomes a valid concern. Organizations, in case of doing software updates, still need to shut down the service for the period of update. It is also important to determine for the administrator how to detect problems in cloud services early for being able to take corresponding remedial actions [S7]. Project mOSAIC [S62] ensures availability of the application by allowing the application deployer to request to the Cloud agency the re-allocation of new Cloud resources, or by allowing developing application on the developer desktop (through the usage of the Portable Testbed Cluster).

4.2.5. Interoperability and Portability

Interoperability means "the capability to communicate, to execute programs, or to transfer data among various functional units under specified conditions [S72]". Study [S30] shows that interoperability concern is increasing because of the increased usage of *Intercloud* [S30] models. They are addressing problems with multi-domain heterogeneous cloud based application integration and interoperability, including integration and interoperability with legacy IT infrastructure service. Currently, there are no widely accepted semantic interoperability standards for the Cloud [S41]. Study [S30] proposed Intercloud Architecture (ICAF) which consists of four components:

- Multilayer cloud services model – for integration and compatibility that defines both relations between cloud service models and other functional layers;
- Intercloud control and management plane – for controlling and managing intercloud applications, resources scaling and objects routing;
- Intercloud federation framework – allows independent clouds belonging to different cloud providers and administrative domains;
- Intercloud operation framework – includes functionalities for supporting multi-provider infrastructure operation, defines the basic relations of resource operation, management and ownership.

Some providers do not allow customers software application and data to be moved from their platform, but enterprises developing applications and storing data should be able to easily choose between different Cloud providers or to move to another Cloud provider if necessary. Lack of interoperability, or known as vendor lock-in, is most visible in cloud applications such as Xing, LinkedIn, Facebook from which is impossible to retrieve your own data after once stored [18]. "Solution is to standardize the APIs so that a SaaS developer could deploy services and data across multiple Cloud Computing providers [S70]".

Another related quality attribute is portability, which is "the ability to transfer data from one system to another without being required to recreate or re-enter data descriptions or to modify significantly the application being transported. It is the ability of software or of a system to run on more than one type or size of computer under more than one operating system [S72] ". Study [S1] shows that the fear of the cost of moving a service from one provider to another is so high that enterprises usually comply with a provider, even in conditions of poor performance, higher than expected costs etc. "The portability in a large market of Cloud offers should allow the consumers to be able to use services across different Clouds by seamless switching between providers or on-demand scaling out on external resources other than the daily ones [S62] ".

4.2.6. Elasticity and Scalability

Elasticity is “the capability to dynamically increase or decrease available resources on demand [S72] “.Todd Papaioannou, Vice President of Yahoo's cloud architecture, quoted in [10] “My biggest problem is elasticity. Ten to 20 minutes is just too long to handle a spike in Yahoo's traffic when big news breaks such as the Japan tsunami or the death of Osama bin Laden or Michael Jackson.“. Ideally a cloud platform is infinitely and instantaneously elastic, but the real clouds are not. Study [S2] shows that there is inevitable delay between when resources are requested, and when the application is running and available on it. It continues defining the factors that the resource provisioning depends on: the type of cloud platform, the availability of spare resources in the requested region, the demand on the cloud platform from the users, the rate of increase of the workload [S2]. Figure 4.4, shows the main included components of an elastic cloud computing model. Study [S2] explored different elasticity scenarios for the three applications (BigCO, Lunch&COB, FlashCrowd). Elasticity scenarios are:

- Default (10 minute spin-up time) – default settings for illustrating typical elasticity characteristics;
- Worst case elasticity – no elasticity mechanisms, relies on fixed over-provisioning of resources;
- Best case elasticity – assuming zero spin-up time, the most elasticity that can be achieved on a cloud platform;
- Perfectly elastic scenario – no time delay between detecting load changes and changing resourcing levels;
- Elasticity break point – break point defines where the platform is not elastic enough.

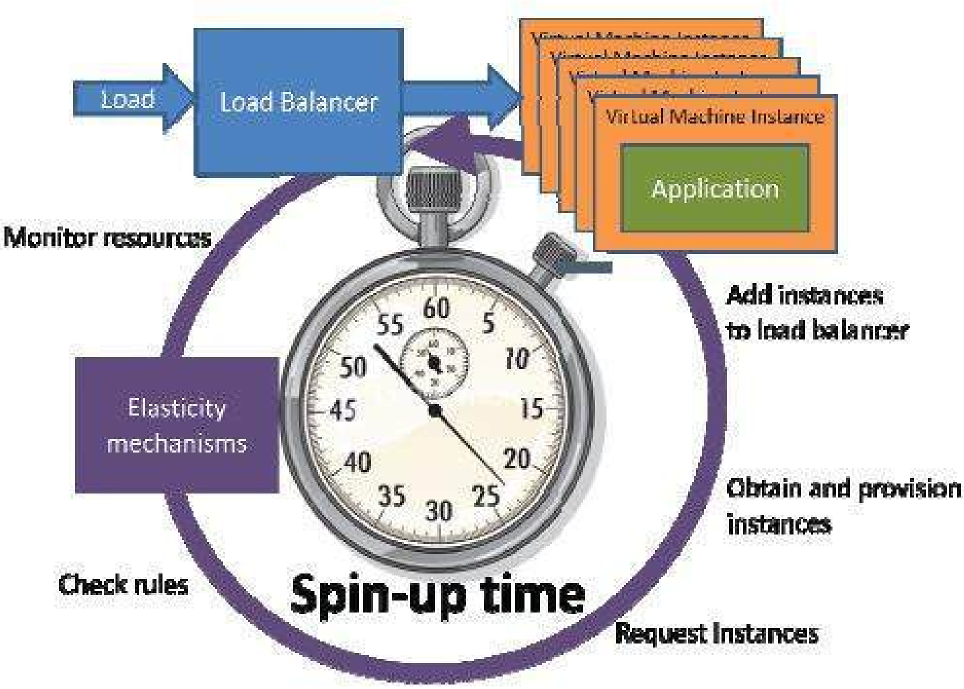


Figure 4.4, Cloud elasticity architecture [S2]

Results of the scenario cost analysis are shown in the Figure 4.5.

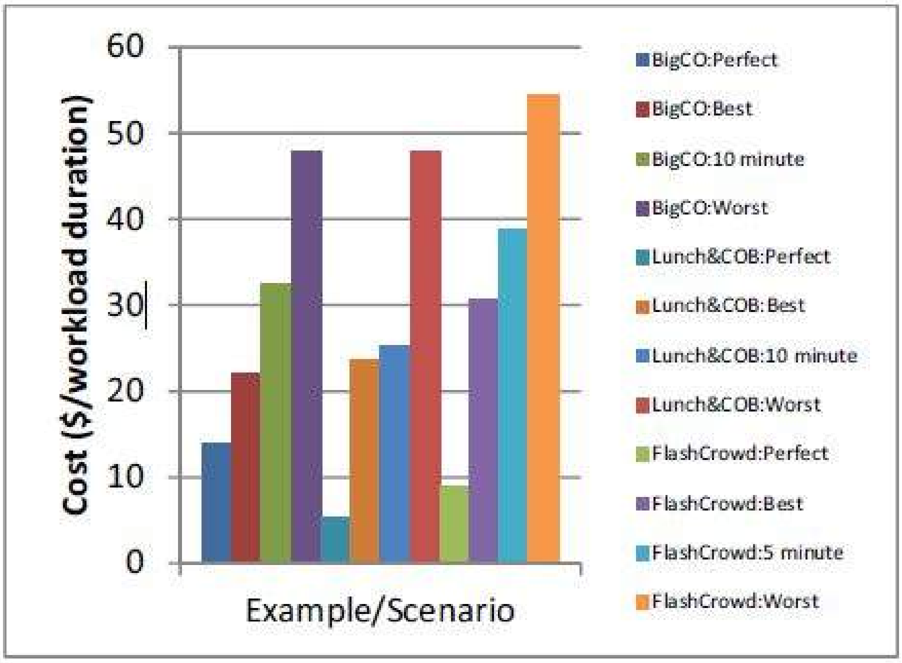


Figure 4.5, Scenario cost analysis results [S2]

Studies [S20, S58] define elasticity as the necessary architectural design requirement or a system property. “Although resources can usually be scaled manually, dynamic elasticity through automated scaling mechanisms is the desired by the majority of the cloud service users [S53] “. Elasticity is important to big companies as well as start-ups. “For example, Target, the nation's second largest retailer, uses Amazon Web Services for the Target.com website. While other retailers had severe performance problems and intermittent unavailability on “Black Friday“(November 28, 2008), Target's and Amazon's sites were just slower by about 50% [S70] “.

Cloud developers sometimes get confused with the differences between elasticity and scalability, some of the key differences [31] are:

- In a scaling environment, the available resources may exceed to meet the future demands; while in the elastic environment, the available resources match the current demands;
- Scalability enables enterprises to meet expected demands for services with long-term strategic needs; elasticity enables enterprises to meet unexpected changes in the demand for services with short term, tactical needs;

Scalability

Scalability refers to the ability of a system to handle growing amount of work with stable performance with proportional new resources [S57]. There are two solutions to scale a software system: scale-up and scale-out [S35]. Scale-up means running the application on a machine with the best configuration, while the scale-out expression means running the application distributed on multiple machines with similar configuration. Examples of the growing amount of traffic that system needs to handle: popular search engines such as Google and Bing can generate multiple TBs of search logs every day, on Facebook are around 130TB of user logs created and 300TB of photos uploaded each day [S50]. Study [S35] defines scalable design principles for application servers as:

- Divide-and-conquer – the system tasks should be divided into smaller tasks with single functions, system should be partitioned into components;
- Asynchrony – work can be done at the moment resources are available, it includes distributed self scheduling and background processing;

- Encapsulation – system components and layers need to be well encapsulated;
- Concurrency – tasks can be done in parallel taking advantages of the distributed nature of hardware and software;
- Parsimony – the design considers the cost efficiency.

4.3. Cloud Computing Architectures

In previous chapter 4.2, we discussed about the main concerns that worry both industry and academy regarding the adapting or moving their service to the cloud. In this chapter we will analyze issues with current Cloud Computing architectures and give guidelines for designing a cloud solution. Furthermore, we will analyze proposed architectures in the selected studies, categorizing them by cloud models (private, community, hybrid, intercloud and public cloud) or services (IaaS, PaaS, SaaS and XaaS).

4.3.1. Issues with current Cloud Computing Architectures

According to the study [S16], existing Cloud Computing architectures solved some concerns such as service migration, multi-tenancy supports, cloud computing architecture principles etc, but there are some problems which current architectures have not solved. Those problems are [S16]:

- Users are often tied with one cloud provider - it is difficult to migrate the same application onto a different cloud;
- Computing components are tightly coupled - current cloud implementations do not allow flexibility to customize selection of included components;
- Lack of Security and access control supports - most current architectures do not consider security and access control management;
- Lack of common use of supports - building a scalable and reusable cloud computing architecture to support sharing resources still faces challenges;

Besides the above mentioned problems, study [S37] points out an additional one:

- Lack of flexibility for User Interface - user interface composition frameworks, have not been integrated with cloud computing.

By the study [S67], most important issues while architecting for the cloud are networking and data management. Some other issues with current Cloud Computing architectures are addressed in studies [S35, S58] with respect to scalability, development complexity, not balanced workload and lack of availability. Examples of such architectures are [S35]: Salesforce.com, Yahoo! PNUTS hosted data serving platform, Amazon DynamoDB service and BigTable Family distributed storage system. Study [S31] stated that "the current approaches to enabling real-time dynamic cloud infrastructure are inadequate, expensive and not scalable to support consumer mass-market requirements". Study [S33] shows comparison statistics of current cloud architectures and if the main concerns are solved or not, table 4.4 (red colour indicates that the concern is not solved, green colour indicated solved concern). (*SOCCA: Service Oriented Cloud Computing Architecture, EC2: Elastic Compute Cloud from Amazon Web Services*)

Quality Features	**Service-Oriented Computing**	**Eucalyptus cloud platform**	**OpenNebula cloud platform**	**Google's Open Social API**	**Virtualization**	**SOCCA**	**EC2**
Availability	🟩	🟥	🟥	🟥	🟥	🟥	🟩
Reliability	🟥	🟥	🟥	🟥	🟥	🟥	🟩
Security	🟥	🟥	🟥	🟩	🟥	🟩	🟩
Scalability	🟥	🟥	🟥	🟩	🟥	🟩	🟩
Data integrity	🟥	🟥	🟥	🟥	🟥	🟥	🟩
Easy-to-use framework	🟥	🟥	🟥	🟥	🟩	🟥	🟩

Table 4.4, Comparison statistics of multiple quality attributes in current cloud solutions [S33]

As shown by the table 4.4, Elastic Compute Cloud (EC2) from Amazon Web Services seems to offer the best service. According to the study [S70] EC2 is selling 1.0-GHz x86 cloud instances for 10 cents per hour, and a new instance can be added in 2 to 5 minutes. However, studies and experiments have shown that it is possible to break its secure cloud environment despite strong encryptions [S47].

4.3.2. Architectural view and concerns from different stakeholders perspectives

As described in the second chapter, there are three main Cloud stakeholders: cloud provider, cloud consumer and cloud broker. In this section, we will present the main activities, concerns and architectural viewpoints for each stakeholder.

Cloud provider

Becoming a Cloud Computing provider in a way of "building, provisioning, and launching such a facility is a hundred-million-dollar undertaking [S70] ", but still, because of the rapid growth of interest, many large companies (Amazon, eBay, Google, Microsoft etc.) became cloud providers. Study [S21] defines cloud provider benefits in architecting cloud solution as publishing and sharing manufacturing resources, publishing and sharing manufacturing business and getting corresponding income.

Activities and challenges of cloud provider

Maintaining, monitoring, operating, and managing are the main activities for cloud providers [S40, S68, S72]. Study [S72] defined activities (Figure 4.6) of cloud provider regarding three main layers (IaaS, SaaS, PaaS). According to the study [S72], for SaaS, the cloud provider deploys, maintains and updates the operation of the software applications on a cloud. For PaaS, major activities of the cloud provider are to manage the computing infrastructure for the platform and support the development and management process of the PaaS cloud consumer. For IaaS, the cloud provider runs the cloud software which is necessary to make computing resources available to the IaaS cloud consumer.

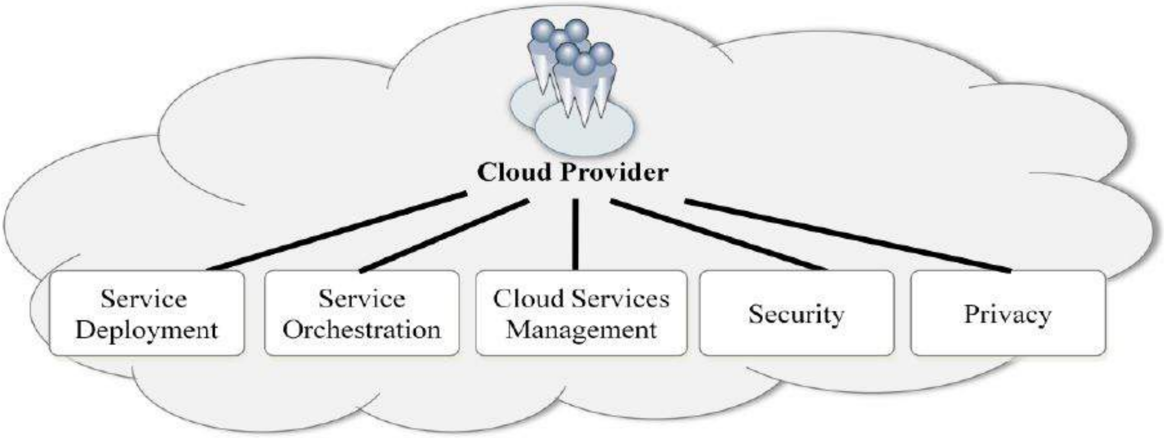


Figure 4.6, Main activities of cloud provider [S72]

Study [S57] defined cloud provider challenges as good server utilization factor, right brokering and resource allocation policies. It proposed a "goal-oriented simulation approach (*CloudSim*) for cloud-based system where stakeholder goals are captured, together with such domain characteristics as workflows, and used in creating a simulation model as a proxy for the cloud-based system architecture(*myki*). Results of the simulation have shown that using two data centres (DC1 and DC2) have higher possibility for meeting stakeholder goals (Figure 4.7). Both [S57, S68] studies define Cloud stakeholder tasks and challenges while architecting for the cloud.

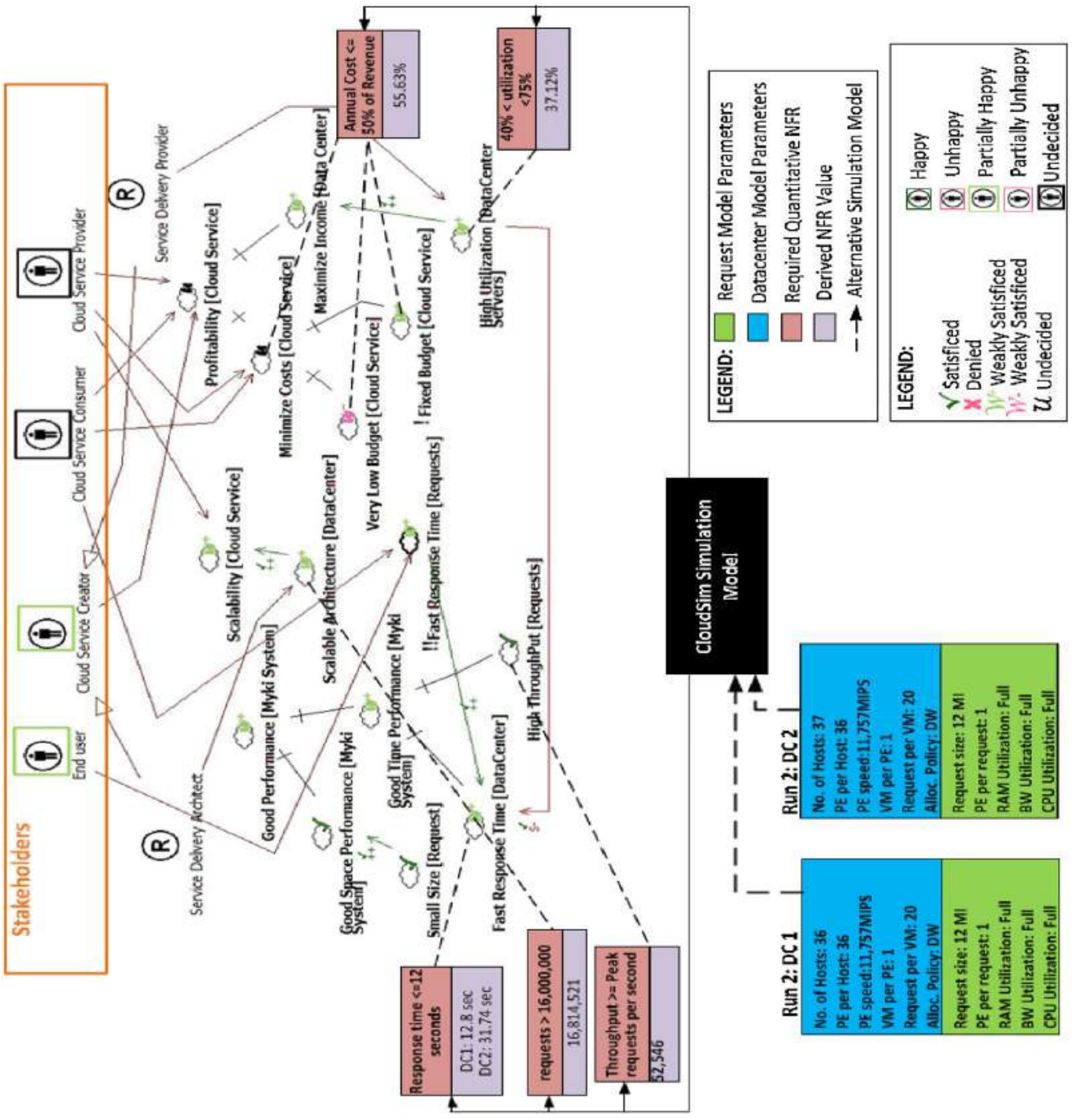


Figure 4.7, Proposed myki simulation to meet stakeholder goals [S57]

Cloud consumer

Studies [S28, S68, S72] defined characteristics of the public or private cloud service consumers as follows: self-service, standard API for accessing cloud services, rapid service provisioning and pay-per-use. Study [S72] defined which role cloud consumer holds in layers (SaaS, PaaS and IaaS). Cloud consumers of SaaS can be organizations that provide access to software application, end users who use the application or software application administrators who configure applications. Cloud consumers of PaaS can be application developers who design and implement application software, application testers who test applications, application deployers who publish applications into the cloud, or application administrators who configure and monitor application performance. Cloud consumers of IaaS can be system developers and system administrators.

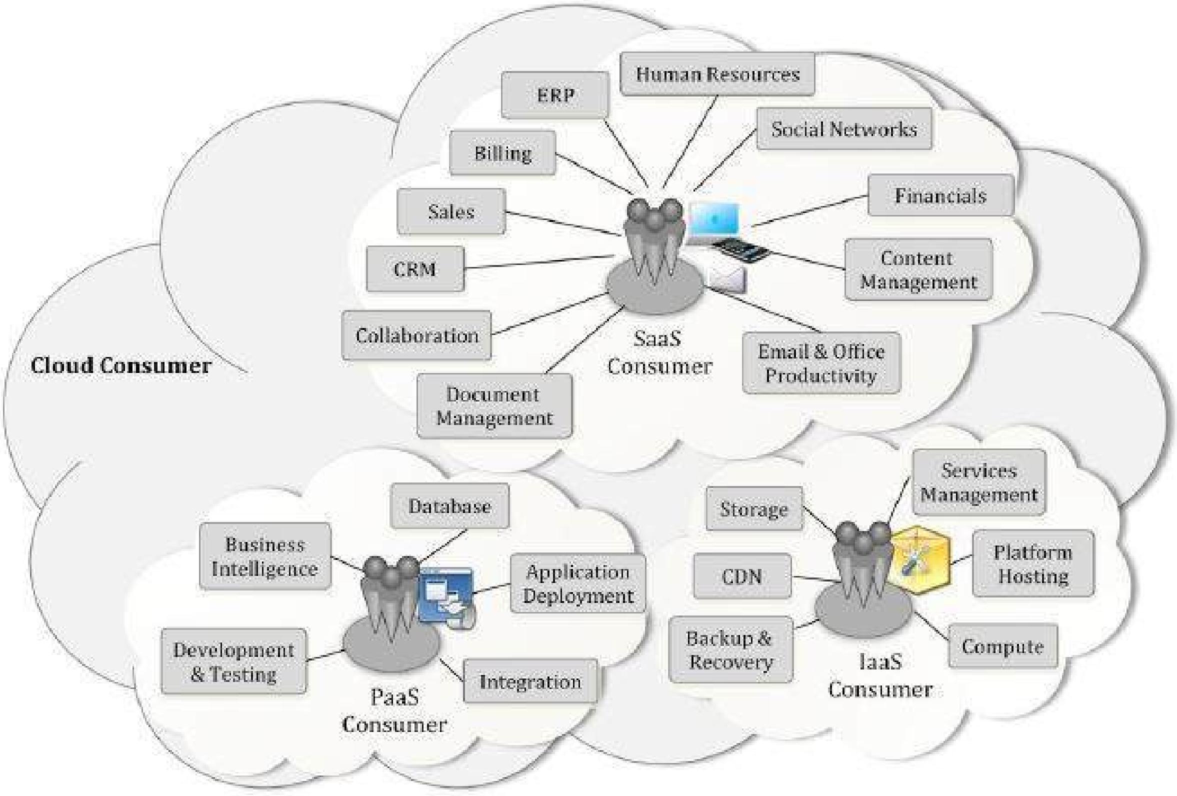


Figure 4.8, Available services to cloud consumer [S72]

Cloud consumers concerns and risks

Major concern for user in using cloud architecture is unawareness from the location of data since cloud providers do not provide that information [S33]. Study [S45] defines user target dimensions as flexibility, costs, scope and performance, security and compliance, reliability and compliance, service and cloud management. Study [S42] has classified user risks of using cloud computing architectures, in more detailed way, into five main categories:

- Operational risks - availability, reliability, integrity, fit, maintainability;

- Contingent risks - survival, major service interruptions, frequency of interruption and resilience, compatibility, flexibility;
- Security risks - service security, data security, authentication and authorisation, susceptibility do denial of service attacks;
- Business risks - cost, customer service, privacy breach, legal compliance;

"As cloud computing platform is transparent to users who generally do not know where and which virtualization platform their virtual machines built in, the customers will never know the 'neighbours' of their virtual machines has been utilized by attackers [S55]". Because of this concern, study [S55] proposes Trusted Cloud based on Security Level (TCSL); an integrated, secured and trusted architecture (Figure 4.9).

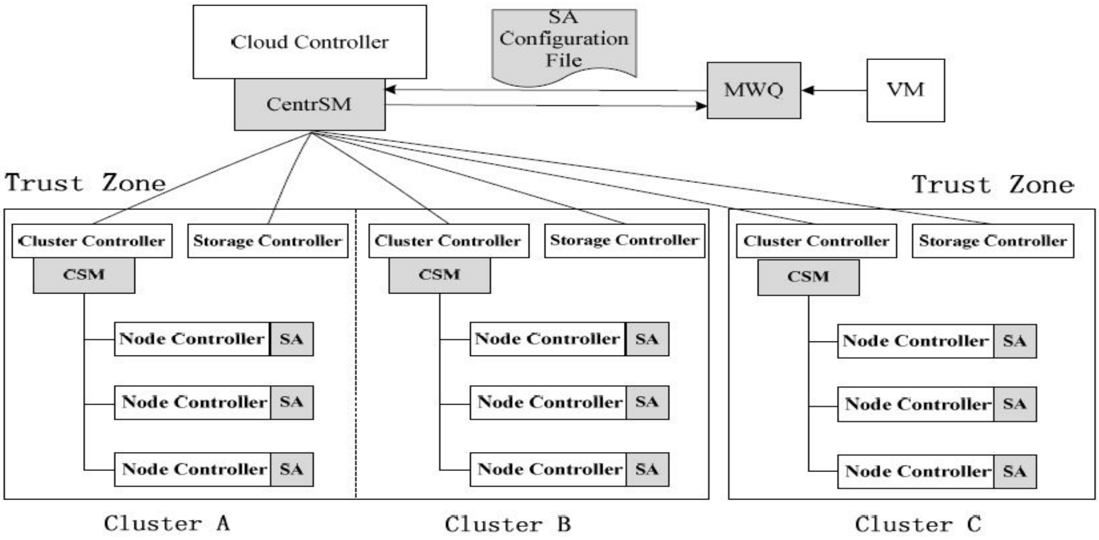


Figure 4.9, Trusted Cloud based on Security Level Architecture [S55]

TCSL architecture divides cloud environment in different zone with different security levels. There are four design principles which guided designing the TCSL architecture [S55]:

- Simplicity – TCSL includes both IaaS cloud architecture and several functional modules such as reliable migration module;
- Isolation – security services are provided by the trusted zone with different security levels preventing the corresponding security problems;
- Flexibility – it provides a general framework with the security division. The security-based architecture can adapt to any practical cloud application;

- Scalability – cloud services can improve security levels according to the customer's potential needs, corresponding security attributes can be added or modified continuously.

4.3.3 How to architect for the Cloud - Principles and Considerations

"Unlike development of traditional web applications, cloud application development has a phase of simulating the application on a local PC environment and a phase of staging it on the cloud for verification before switching it to a production state [S38]." Study [S38] classified lifetime phases of architecting for the cloud as follows:

1. Designing application
2. Implementing application locally
3. Simulating application locally
4. Deploying application to cloud environment
5. Staging application on cloud environment
6. Operating application on cloud environment

Client-side platforms cover first to fourth phase, while cloud-side platforms cover fifth and the last phase. Study [S71] designed an architectural pattern language for designing cloud-based applications. Study presents a new architecture (*STAR*) which helps developers to develop cloud computing applications in more systematic manner [S38]. Proposed architecture consists of three stages and eight layers (Figures 4.10, 4.11, 4.12):

- Requirement/Specification stage – this stage starts with an application business requirements and ends with the application selected service model, service provider and deployment model. It consists of four layers:
  - Specify application requirement layer – all the services, business needs and an estimation of the pricing model are defined and provided to developers.
  - Select service model layer – according to the application functions, services, security and cost, one or more of the cloud service models will be chosen.
  - Select deployment model layer – decision on which cloud infrastructure will the application be deployed.
  - Select service provider layer – selection of service provider to achieve the application functions.

- Development and deployment stage – this stage ends with the application running on the cloud. It consists of two layers:
    - Application development layer – the application is developed by managing files which can be: a new file, shared file, or a follow file.
    - Application deployment layer – all application objects and files are deployed on the selected cloud provider.
- Management and maintenance stage – it includes all the service-related components necessary for the management of the developed application and its functions. It consists of two layers:
    - Application management layer – it includes two parts: application management and security management.
    - Application change management layer – it handles changes in the cloud application requirements.

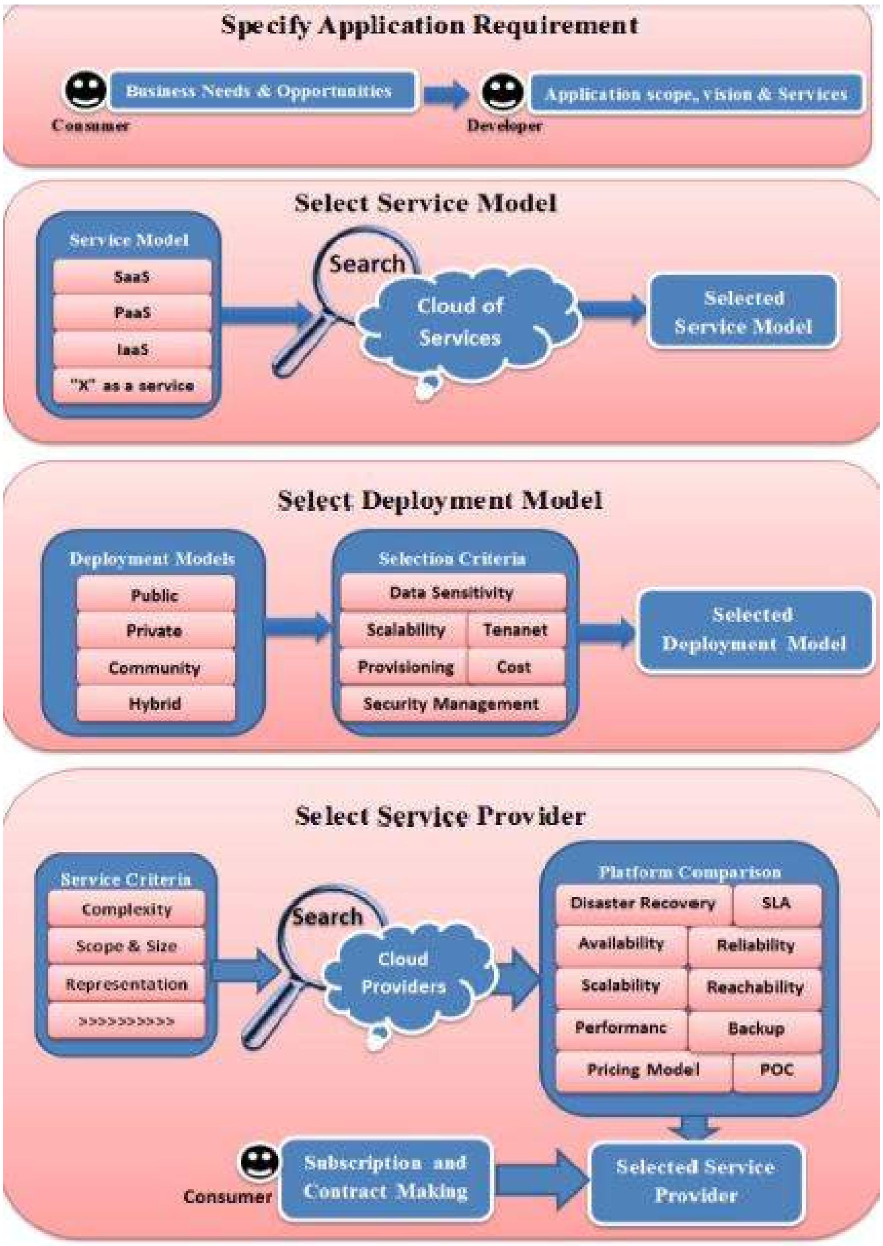


Figure 4.10, Proposed STAR architecture; Specification/Requirements stage [S38]

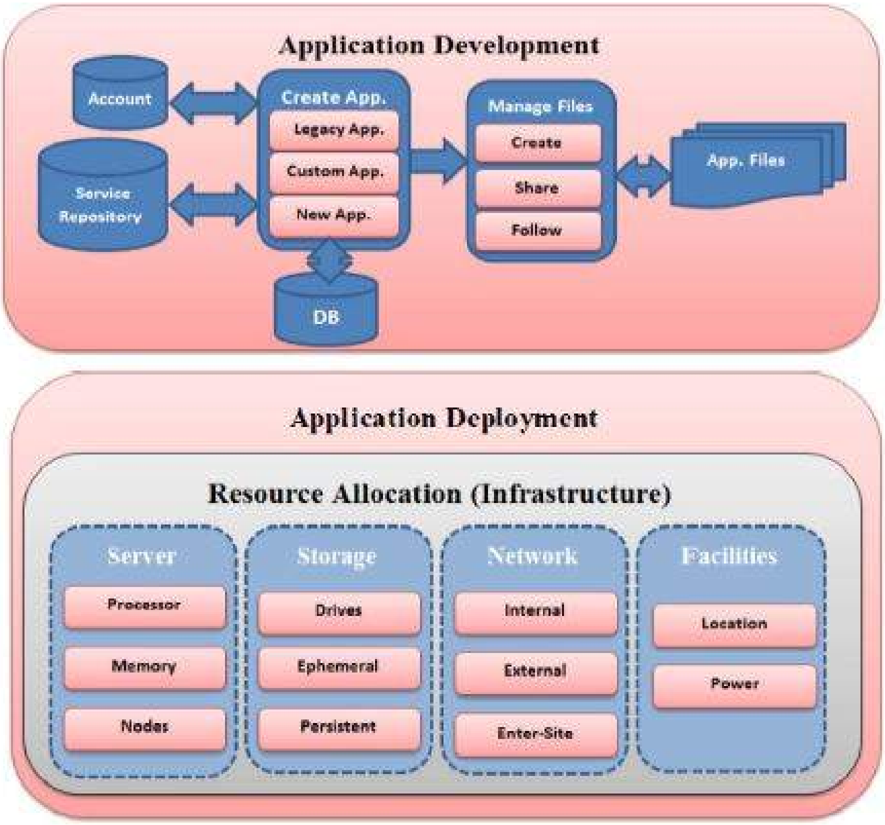


Figure 4.11, Proposed STAR architecture, Development and Deployment stage [S38]

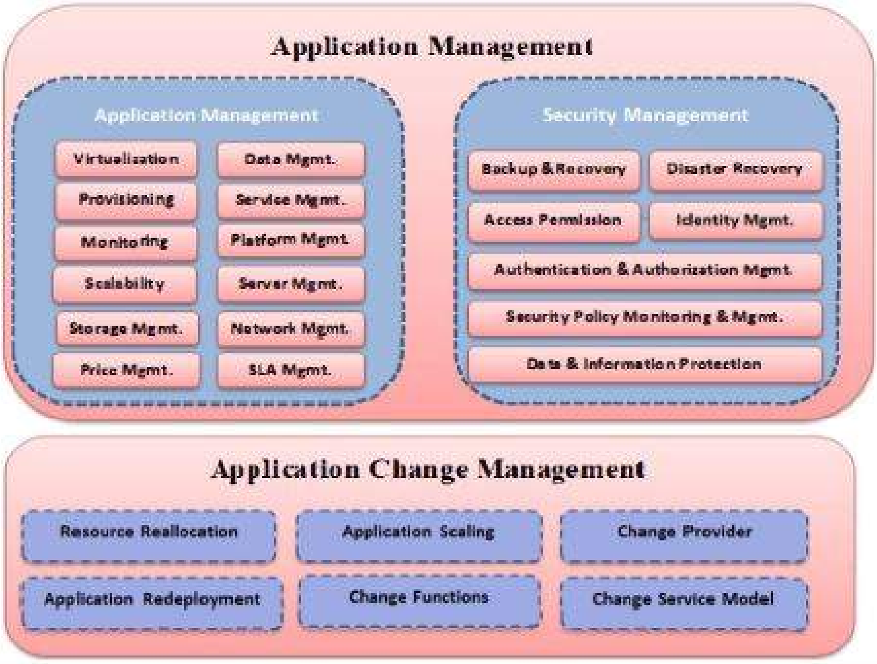


Figure 4.12, Proposed STAR architecture, Management and maintenance stage [S38]

When application is moving to the cloud, there are some key design considerations to think about in order to allow elasticity and dynamic usage: service composition, interoperability and information centric design [S47]. Top six recommended privacy practices for cloud system designers, architects, developers and testers , by the study [S3] are: minimise personal information sent to and stored in the cloud, protect personal information in the cloud, maximise user control, allow user choice, specify and limit the purpose of data usage, provide feedback. In case of construction and maintenance of a service, challenges are: use of volunteer computing, service data management (data upload and transfer, data processing, data and network isolation, data analysis, data security, data disposal) [S47]. Study [S19] defined challenges for a cloud computing strategy such as management and security.

Next generation service

Study [S47] has identified six important aspects of the lifecycle of the *next generation service* which are: design, engineering, deployment, usage measurement, support and maintenance, experience. Next generation service is defined as a transformed traditional service offered by an IT service company, which will be measurable, highly affordable by offering extremely granular usage [S47, S31]. Study [S68] gave an overview of scenarios (creating a cloud service and publishing it in the service catalogue, requesting/using the service, and managing the cloud environment) involved in any cloud implementation and how the architecture provides the appropriate capabilities to address that scenarios. Study [S25] defined steps for adopting cloud data centres in a private cloud deployment model.

Architectural methods

Regarding the research methods, one of the method is *separation of concern* which means concentrating on one aspect and momentarily disregarding the others. Studies [S46, S54, S31] use this method for designing a cloud solution. Study [S14] defined seven principles of cloud computing architecture which are: ecosystem enablement, cloud infrastructure and its management, service-orientation, cloud core on provisioning and subscription, compostable cloud offerings, cloud information architecture and management, cloud quality analytics. Those principles present logical separation for isolating concerns of details of each module during the design process. Also, in our selected studies there were four studies [S67, S22, S45, S65] which used systematic literature review as a research method.

4.3.4. Private Cloud Architectures

Challenges and Issues

There are several issues addressed by the study [S34], and those are: heterogeneity of network and limitations of mobile devices. Furthermore, study [S34] includes major components for the provision of Quality of Service into their proposed model such as: monitoring, load balancing, traffic management and security. Study [S56] researched about enhancing data flexibility and reliability. Study [S57] addressed needs for better rational decision making and cost savings for Cloud Computing. Challenges for the future are how their approach [S57] can be applied to other layers in the XaaS model.

Proposed Solutions

“Since private cloud enables the enterprises to preserve their existing IT infrastructure and provide flexibility and scalability, enterprises can have their own private clouds for efficiency. To build a private cloud, the enterprise needs to consolidate the infrastructure and virtualized servers [S34]“. Companies that have proposed their private cloud solutions are: *vCloud from Vmware, ECI Datacenter from Microsoft, Virtualized Multi-Tenant Data Centre from Cisco* [S34]. Study [S57] focused on private cloud deployment model, in which one entity (Cloud creator) “owns and operates the data centre but gives shared access to other entities who subscribe on a pay-as-you-go basis“. “Building a cloud computing system or the power system can maximize the integration of data resources and computing power of the power system [S56] “. Combined with the characteristics of the power system (accumulation of large amounts of data) researches in the study [S56] proposed a private cloud (Figure 4.13) of power system (Electric power private cloud).

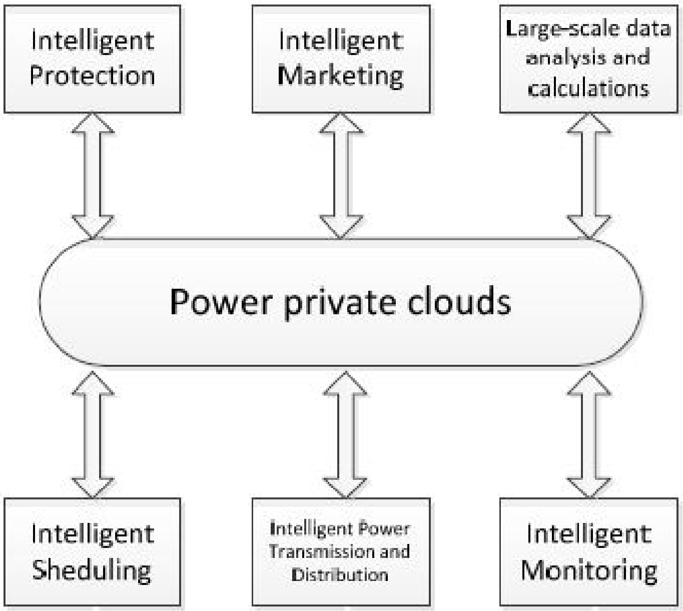


Figure 4.13, Electric power private cloud model [S56]

Data storage in the proposed architecture is service oriented and it consists of five layers: storage service layer, data management layer, transport layer, application service layer and access layer. Application service layer is the most flexible part of proposed cloud model and it includes technologies such as safety, recovery and backup technology. In the study [S34], architecture to build an enterprise private cloud for multimedia services, is presented (Figure 4.14). Quality of service management is implemented in the infrastructure layer. Users are able to add new streaming applications to the system and share the resources.

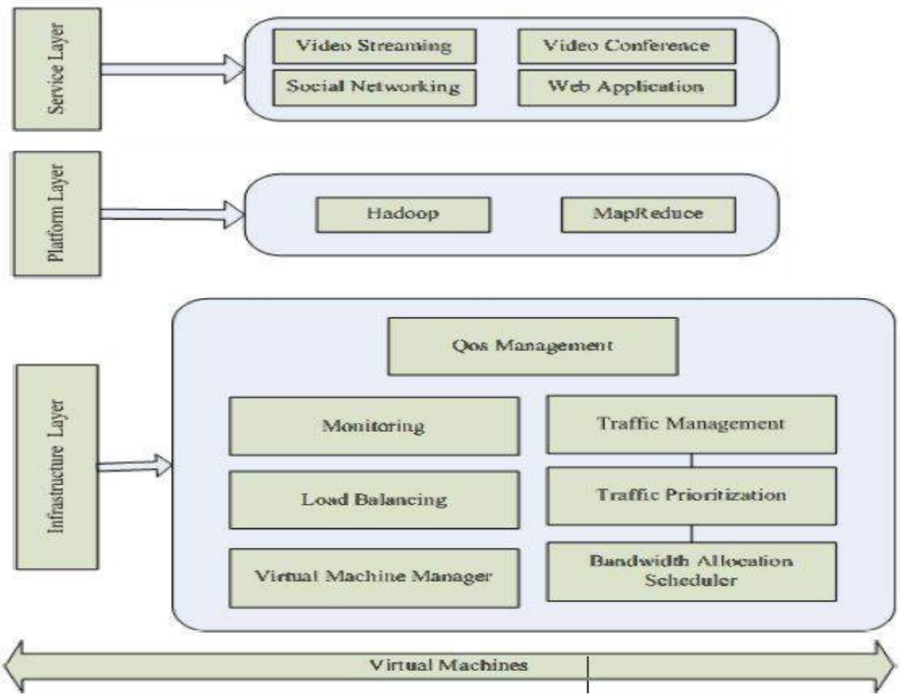


Figure 4.14, Proposed multimedia cloud computing architecture [S34]

4.3.5. InterCloud Architectures

Challenges and Issues

Study [S30] addresses issues with multi-domain heterogeneous cloud based applications integration and inter-provider and inter-platform interoperability. Study [S25] focuses on inter cloud data centre network, integration APIs and software applications aspects of cloud data centres.

Proposed Solutions

The possible ways of cooperation between clouds (known as *InterCloud*) have become more popular both in industry and academy [S9]. Study [S30] presents InterCloud architecture (further explained in the 4.2.6 sub section) which “is based on the development and implementation of its different components in a few cooperating projects such as GEYSERS (first chapter), GEANT, MANTICHORE and NOVI, which have needs for more general approach to complex multi-provider cloud based infrastructure services". "In the long term, the Intercloud will emerge as a public [S25] ".

4.3.6. Hybrid Cloud Architectures

Challenges and Issues

Study [S59] addresses the main issues of building hybrid cloud architecture such as flexibility and scalability. Furthermore, they considered different performance metrics such as deadline violation rate and job slowdown. Study [S44] with its proposed solution solved issue of interoperability, extensibility, unified and centralized management requirements. Study [S60] addresses issues of adopting cloud solutions in the non developed countries. Concerns that have not been solved, in the study [S44] are data security, isolation, integrated monitoring and event management in hybrid cloud environments. Challenges for the study [S59] are to run real experiments based on proposed strategies. Further researches of the study [S60] will be identifying best practices for addressing pedagogical, technical, political and economic issues for adopting proposed cloud infrastructure.

Proposed Solutions

Hybrid Cloud is experiencing increasing attention, "in order to realize the full potential of the hybrid Cloud platform, an architectural framework for efficiently coupling public and private clouds is necessary [S59] ". Study [S59] proposed flexible and scalable hybrid Cloud architecture along with failure-aware resource provisioning policies. Study [S44] described the support for all three integration patterns (provisioning, monitoring and data integration) in the hybrid cloud framework (Figure 4.15). "A key feature of that architecture is the integration platform that allows development and deployment of function specific integration plug-in components and ability to control integrations using policies [S44]".

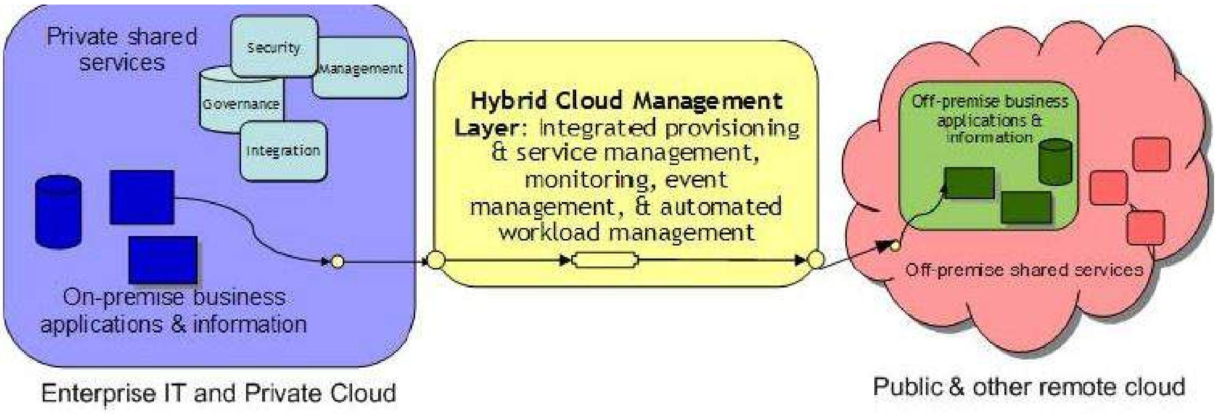

Private shared services
Security
Management
Governance
Integration
On-premise business applications & information
Enterprise IT and Private Cloud
Hybrid Cloud Management Layer: Integrated provisioning & service management, monitoring, event management, & automated workload management
Off-premise business applications & information
Off-premise shared services
Public & other remote cloud

Figure 4.15, Proposed architectural solution [S44]

Study [S60] proposes hybrid cloud architecture (Figure 4.16) for the online commerce from a retailers' point of view.

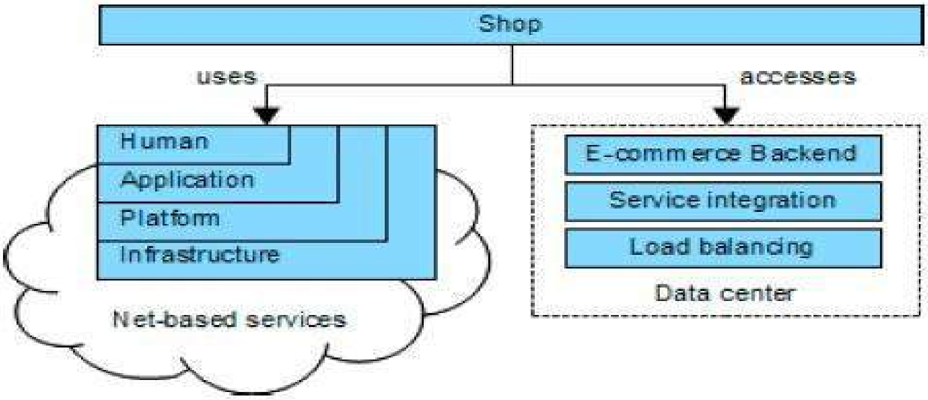


Figure 4.16, Integration of different service types in an online shop [S60]

Study [S60] integrated net-based services in an online shop. The data-centric catalogue component is distributed in net-based data-centres to support application scaling in peak situations. "Hybrid architecture combines the stable development and deployment model of a shop's core with the rapid application development and deployment of a web-based platform. This model also allows testing the acceptance of experimental features before integrating them into the core system [S60]".

4.3.7. Community Cloud Architectures

Challenges and Issues

Issues that the study [S9] is solving are scalability and availability. Future challenges for the study [S9] will be expanding the policy management, making it possible to intercept requests only once.

Proposed Solution

"A community cloud serves a group of Cloud Consumers which have shared concerns such as mission objectives, security, privacy and compliance policy, rather than serving a single organization as does a private cloud [S72]". Study [S9] introduces the *Community Intercloud* (Figure 4.17) term. Conceptually the Community Intercloud is a large distributed system that links together clouds from different administrative domains; problems of such systems are that there is a possibility that parts of the system might become temporarily unavailable [S9]. Study [S9] proposed an architecture that ensures cloud consumers that even if parts of the system are unavailable the remaining parts of the system will be able to work properly.

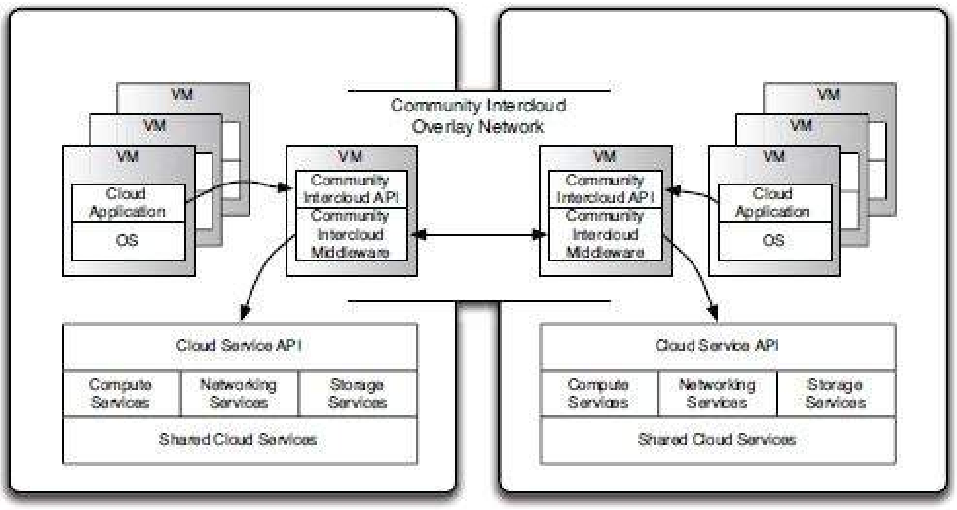


Figure 4.17, proposed Community Intercloud architecture [S9]

Cloud applications are deployed in both clouds. The middleware has the function of a gateway for the cloud applications.

4.3.8. Public Cloud Architectures

Challenges and Issues

Issues that the study [S10] is solving are isolation, security and data protection. Furthermore, they removed limitations in the need for migrating to a cloud programming model.

Proposed Solutions

"Service providers build public clouds to offer on-demand, secure, multi-tenant, pay-per-use IT infrastructure to businesses and government agencies that use cloud services to offload, or augment, their initial resources using a public cloud infrastructure [S25]". "Organizations shifting to a public cloud infrastructure face potential hurdles regarding control and security, and must acquire a new set of best practices regarding developing and deploying to a cloud infrastructure [S10]". Study [S10] proposed reference architecture (AERIE) for a virtual private cloud (Figure 4.18) built on cross provider that increases control and isolation, improves security and data protection.

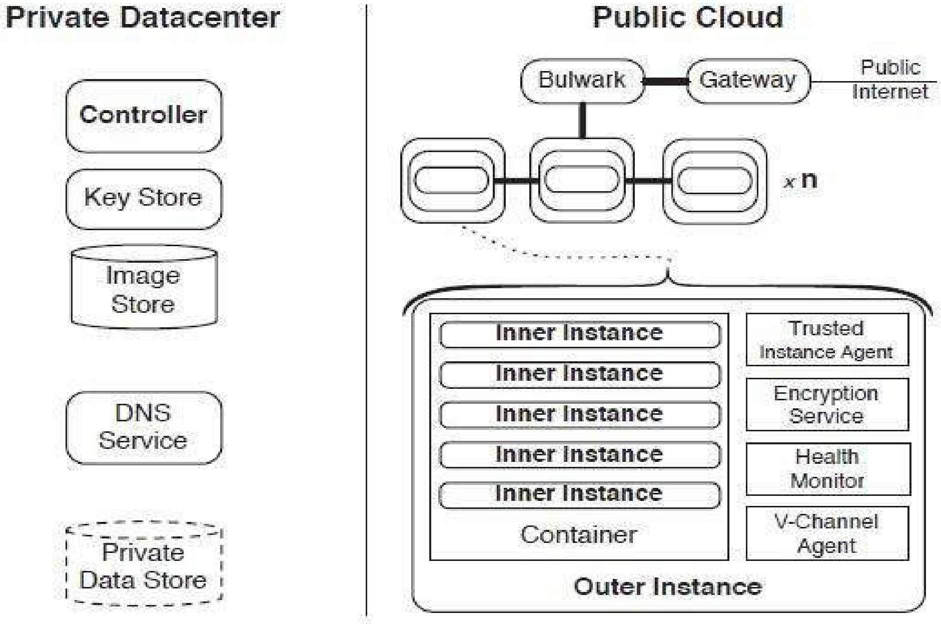


Figure 4.18, Logical components of the AERIE reference architecture [S10]

AERIE reference architecture defined two classes of components: components included on the outer instance and the supporting components which manage the outer instances and the supporting network nodes. The outer instance is launched by the controller using the API provided by the cloud provider. AERIE reference architecture achieves [S10]:

- Isolation – to improve the isolation between the outer and inner instances, the inner instance could run a different operating system;
- Network security – connecting each nested instance over a virtual channel allows routing and encrypting all traffic between the application and the database that might have not been encrypted at the moment of deploying to a private data centre;
- Confidentiality – solved by storing sensitive data in a private cloud, accessible only by the virtual channel already in place for the deployed application;
- Integrity – only an instance running an unmodified image is given the keys and certificates required for joining the topology, it helps to protect the outer instance;
- Availability – ensured by deploying to multiple public cloud data centres and using round-robin DNS to load balance.

4.3.9. Cloud “as-a-Service“ Provider Architecture

IaaS

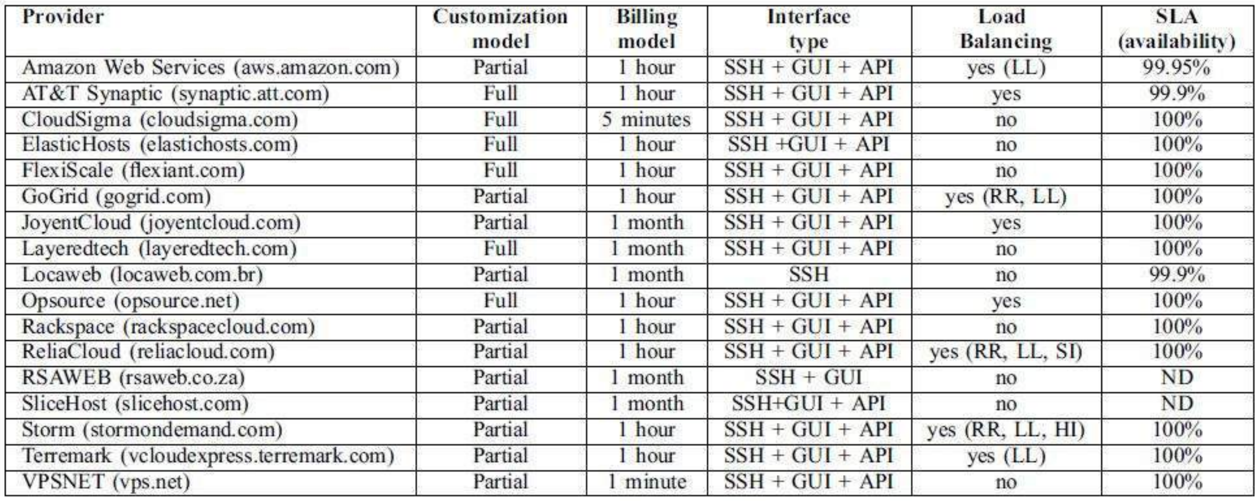
Challenges and Issues

Study [S12] emphasized that, at the moment, services offered by various IaaS providers do not provide services such as auto-scaling and advanced monitoring. Study [S2] enhanced service oriented performance modelling method and tool to model and to predict elasticity characteristics of IaaS cloud applications. For the study [S58], main motivation is investigation how do owners manage the cost of their applications (deployed in IaaS clouds) while maintaining the Quality of Service they provide. Study [S66] addressed many issues for the IaaS cloud architecture such as network management, cloud data federation, performance and complexity of virtualization etc. Future challenges, for the study [S12] will be enhancing flexibility into the proposed solution.

Proposed Solutions

Some of the public IaaS providers are: Amazon *Web Services, CloudSigma, GoGrid, SliceHost, Storm* etc. Study [S12] took into a consideration 17 public IaaS providers and their characteristics (Figure 4.19). They identified seven main features that characterize services offered by IaaS providers [S12]:

- Customization model – full model (the end user can specify and modify at runtime the amount of processing power, memory and storage of every virtual machine), and partial model (limited set of predefined virtual machine types. A virtual machine instance cannot be modified);
- Billing model – how virtual machines usage is billed by IaaS providers;
- Interface type;
- Load balancing – “the capability of distributing the incoming load among various virtual machines following a balancing policy [S12]”;
- Service Level Agreement;
- Monitoring services – mechanisms that allow users to monitor system performance;
- Auto scaling services – the capacity to automatically add or remove virtual machines on the basis of monitored performance.



Provider	Customization model	Billing model	Interface type	Load Balancing	SLA (availability)
Amazon Web Services (aws.amazon.com)	Partial	1 hour	SSH + GUI + API	yes (LL)	99.95%
AT&T Synaptic (synaptic.att.com)	Full	1 hour	SSH + GUI + API	yes	99.9%
CloudSigma (cloudsigma.com)	Full	5 minutes	SSH + GUI + API	no	100%
ElasticHosts (elastichosts.com)	Full	1 hour	SSH +GUI + API	no	100%
FlexiScale (flexiant.com)	Full	1 hour	SSH + GUI + API	no	100%
GoGrid (gogrid.com)	Partial	1 hour	SSH + GUI + API	yes (RR, LL)	100%
JoyentCloud (joyentcloud.com)	Partial	1 month	SSH + GUI + API	yes	100%
Layeredtech (layeredtech.com)	Full	1 month	SSH + GUI + API	no	100%
Locaweb (locaweb.com.br)	Partial	1 month	SSH	no	99.9%
Opsource (opsource.net)	Full	1 hour	SSH + GUI + API	yes	100%
Rackspace (rackspacecloud.com)	Partial	1 hour	SSH + GUI + API	no	100%
ReliaCloud (reliacloud.com)	Partial	1 hour	SSH + GUI + API	yes (RR, LL, SI)	100%
RSAWEB (rsaweb.co.za)	Partial	1 month	SSH + GUI	no	ND
SliceHost (slicehost.com)	Partial	1 month	SSH+GUI + API	no	ND
Storm (stormondemand.com)	Partial	1 hour	SSH + GUI + API	yes (RR, LL, HI)	100%
Terremark (vcloudexpress.terremark.com)	Partial	1 hour	SSH + GUI + API	yes (LL)	100%
VPSNET (vps.net)	Partial	1 minute	SSH + GUI + API	no	100%

Figure 4.19, IaaS provider features [S12]

The identification resulted in a way that “no IaaS provider delivers all the services required to implement an autonomic service management solution, no IaaS provider, except Amazon Web Services, delivers integrated autonomic service management solutions, when available, services that enable autonomic resource management have a low degree of customization [S12]“.

IaaS makes possible accessing compute, storage, and networking infrastructure maintained in the data centre of the IaaS provider, it also makes unnecessary for customers to maintain servers [S66]. “Enterprises use the IaaS model to build private clouds [S25] “.

According to the study [S19], key benefits of IaaS are:

- Reduced cost of purchase since the resources already exist, and end users pays only for the used resources;
- Pay for usage and any other combination of the used volume;
- Reduce environmental impact of the resources used on local computing centre.

Study [S2] introduced elasticity mechanism (Figure 4.4) on a typical IaaS cloud platform (Amazon EC2). Study [S45] proposed a provider independent classification framework for IaaS which can be used in e-Government (Figure 4.20), due to the lack of possibilities to compare and classify Cloud providers.

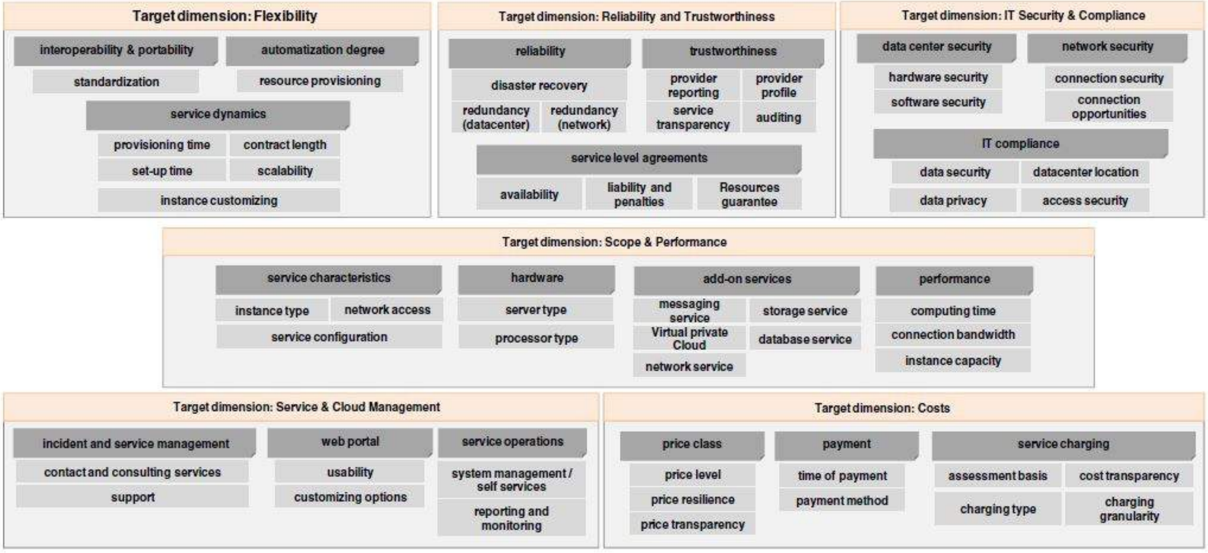


Figure 4.20, Classification framework for IaaS [S45]

The proposed framework has a purpose to help enterprises in their selection process and creates a bigger transparency in the Cloud market. It is divided into six target dimensions as follows: flexibility, reliability and trustworthiness, IT security and compliance, scope and performance, service and cloud management, and costs.

PaaS

Challenges and Issues

Study [S52] defines PaaS issues concerning fault-tolerance, configuration, accounting, performance, security and the interaction with the IaaS layer. Study [S54] addresses PaaS issues such as security and scalability. Future challenges for the study [S52] will be optimizing the checkpoint system and improving performance and security. Study [S54] defines future challenges such as supporting more authentication technologies in order to improve security.

Proposed Solutions

Customers can develop and test their applications more rapidly because PaaS provides an application development and runtime tools [S66].

According to the study [S19], key benefits of PaaS are:

- Lower total cost of ownership since there is no need to own and manage all lower abstraction resources like hardware;
- Minimize management and maintenance since most of the management is part of the vendor data centre;
- Scalable and flexible system capacity, scaling all resources dynamically as needed and scaling them to higher abstraction level (platform environment, development environment);

"The lack of guidelines and computing models makes the designing of PaaS rather difficult. Organization, government etc., are free to conceive its own PaaS [S52]“. Study [S52] proposed a Distributed Resilient Adaptable Cloud Oriented (DRACO) PaaS (Figure 4.21) in order to provide a platform for the development of complex algorithms in the Cloud, to design a middleware for the development of PaaS able to bind the IaaS and the PaaS service layers, and to conceive a new computing model for the development of PaaS in the Cloud [S52].

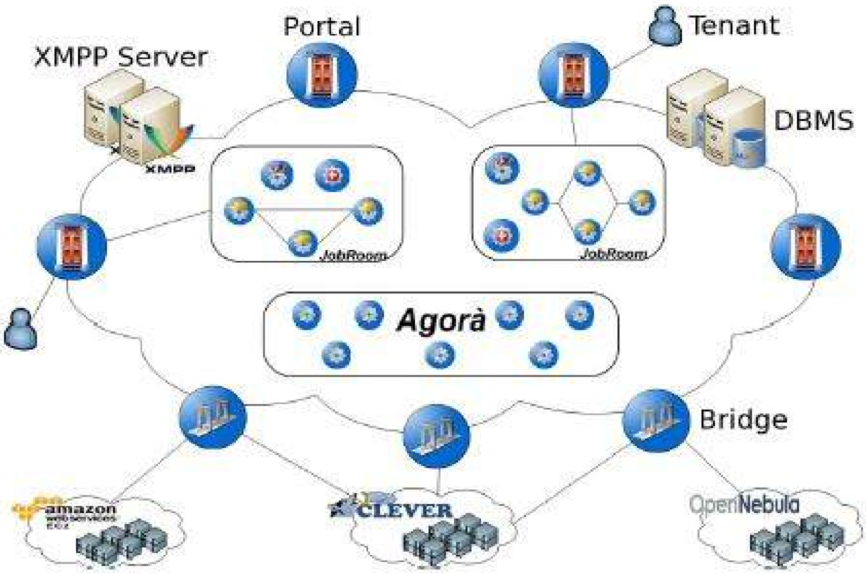


Figure 4.21, Example of DRACO PaaS environment [S52]

The XMPP Server signalling system (Figure 4.21) has the task of allowing DRACO PaaS to quickly recover from workflow failures. Furthermore, there are three specific chat rooms: cloud management (where the Portal and Bridge nodes interact), Agora (the chat room selects Jobnodes to satisfy the requirements of the given workflow), and JobRoom (the chat room assigned to a workflow and the Portal node to which the workflow has been submitted).

Although some Cloud providers offer secure cloud solutions, users are forced to redesign and adapt their applications in order to conform to the provided environment. Some PaaS solutions fail to meet the security requirements of developers and end users [S54]. Study [S54] presented a *MagosCloud secure* (Figure 4.22), made on the basis of MagosCloud (PaaS Cloud solution), designed to meet the needs of traditional and e-science application developers.

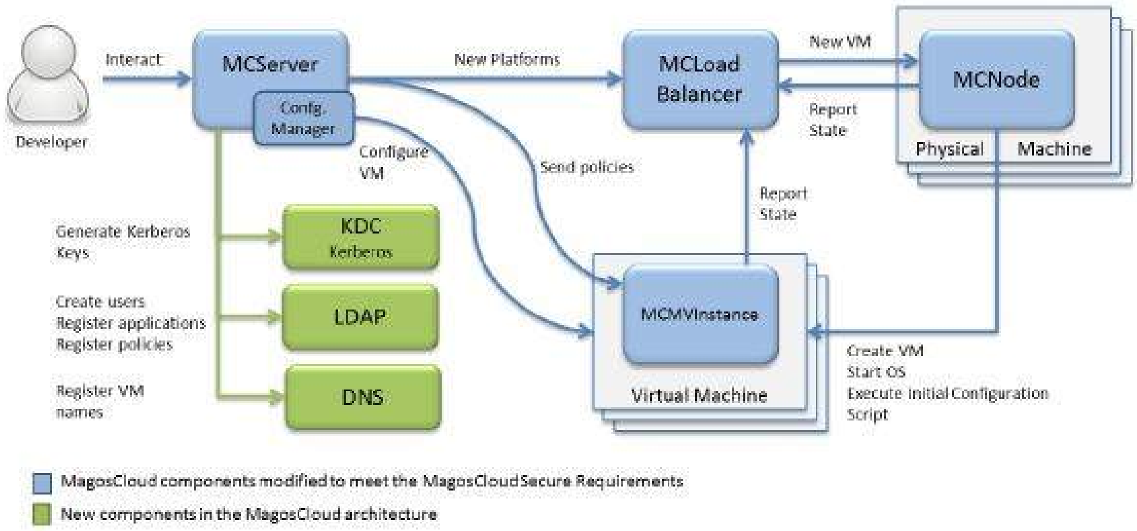


Figure 4.22, MagosCloud architecture after including MagosCloud Secure components [S54]

The system provides mechanisms for authenticating users and applications. Resource allocation is based on a virtualized scheme which guarantees availability and failure recovery depending on the load of the physical machines hosting the infrastructure. MagosCloud Secure gives a model that defines how resources can be shared among users.

In the case study of CCRA (Cloud Computing Reference Architecture) [S15], implementation of PaaS for a traditional software vender (giving them the ability to migrate their products to the cloud as quickly as possible) is done.

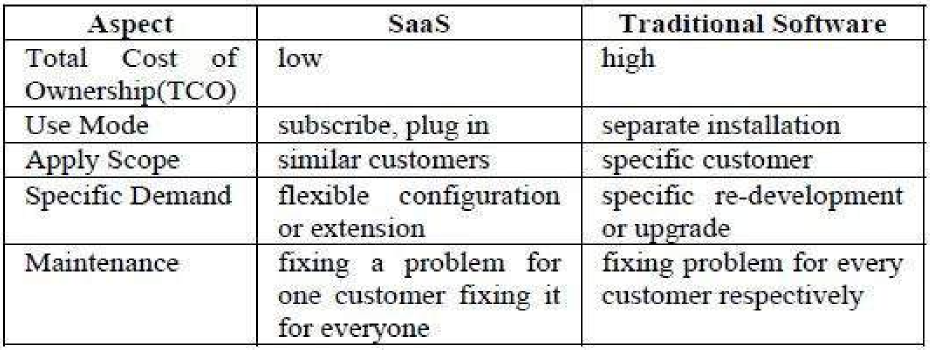
SaaS

Challenges and Issues

Study [S35] address scalability as the main issue of SaaS applications. Both studies [S35, S43] define tenant awareness as one of the SaaS issues. Future challenge for the study [S43] will be improving proposed SaaS architecture to develop customization functions and to extend their two-tier architecture into multi-tier architecture (explained in the sub-chapter 4.3.12).

Proposed Solutions

"SaaS makes it possible for customers to access software maintained in the data center of the SaaS provider. SaaS makes it unnecessary for customers to manage software upgrades, since they are done in the data centre of the SaaS provider. SaaS is provided on IaaS [S66]". Study [S43] gives a comparison between SaaS and traditional software (Figure 4.23).



Aspect	SaaS	Traditional Software
Total Cost of Ownership(TCO)	low	high
Use Mode	subscribe, plug in	separate installation
Apply Scope	similar customers	specific customer
Specific Demand	flexible configuration or extension	specific re-development or upgrade
Maintenance	fixing a problem for one customer fixing it for everyone	fixing problem for every customer respectively

Figure 4.23, Comparison between SaaS and Traditional software [S43]

Historically, the first successful SaaS was the early version of Salesforce.com "where a custom relationship manager application with a set of functionalities was exposed as a paid service [S47] ".

According to the study [S19], key benefits of SaaS are:

- Improved internal resource productivity since there is no need to manage its own resources;
- Rapid delivery of new applications and functionality (it immediately reaches all users connected to a network);
- Better government services, with standardization in services for all users;

The main concern for SaaS applications is scalability. Handling a large number of user requests effectively is critical for SaaS applications [S35]. Study [S35] identifies key factors that affect SaaS scalability and those are: levels of scalability mechanisms, workload support, recovery and fault-tolerance, tenant awareness, automated migration, software architecture, database access. "Current SaaS providers treat customers as independent entities and therefore ignore business interaction and integrated relationship amongst software customers. As a result, the implementation of SaaS in small and medium organizations slows down [S43] ".

Study [S43] proposed two-tier SaaS architecture based on group-tenant (Figure 4.24) which describes business interaction between each tenants of a group and achieves two-tier configuration. A master tenant is in the first tier and can extend and configure common information to meet demands of its slave tenants and its own specific part. Slave tenants whose part of configuration is not based on its master tenant, are also in the first tier. Slave tenants whose part of configuration is based on its master tenant are in the second tier and can configure its own specific part. Tenants of the first tier are known as SaaS tenants, and tenants of the second tier are known as SaaS providers.

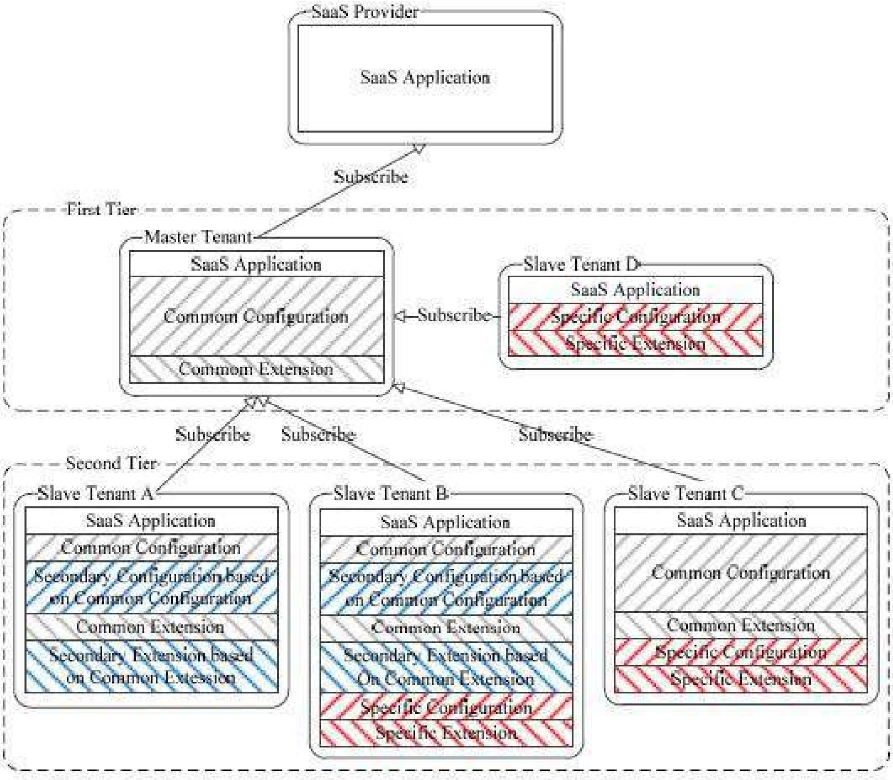


Figure 4.24, Proposed Two-tier SaaS architecture [S43]

In the study [S15], researches used a SaaS application as an example of how their proposed CCRA is applied and showed its architecture.

XaaS

Challenges and Issues

“Cloud storage architecture is a major topic nowadays because the data usage and the storage capacity are increased double year by year [S23]”. Issues that the studies [S6, S11, S8] are solving, regarding the storage, are connected to the security, and time consuming issues. For the proposed architectural solution [S11] to work, it is essential to have “a flexible and elastic scale-up-and-down model where additional resources are made available either on demand or automatically to satisfy a pre-agreed Quality of Service in a Service Level Agreement”.

Proposed Solutions

Cloud computing provides many services (XaaS). Anything can be a XaaS, which means anything can be a cloud service [S17]. Study [S6] proposes *Database-as-a-service (DbaaS)* which provides to large enterprise organizations: higher availability, cost savings, better service through centralized management, reduced risk (Figure 4.25); "an alternate solution to the problem of providing database functionality to application developers offering the database as a common, shared service to the enterprise as a whole [S6]".

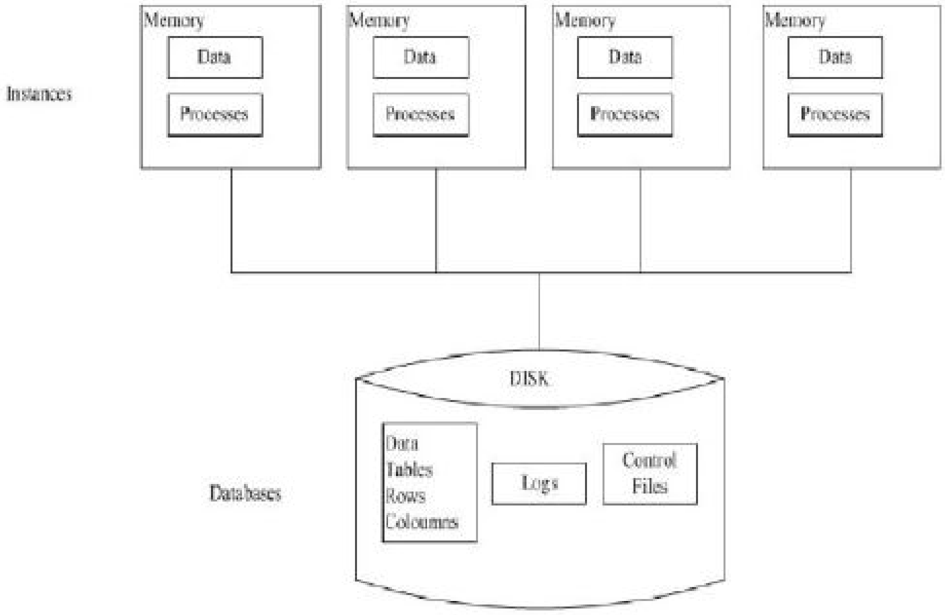


Figure 4.25, Shared disk architecture [S6]

"Shared-disk databases allow clusters of low-cost servers to use a single collection of data, typically served up by a Storage Area Network (SAN) or Network Attached Storage (NAS). All of the data is available to all of the servers, which means there is no partitioning of data [S6]". Shared disk architecture databases support elastic scalability.

Storage-as-a-service and its architecture (Figure 4.26) are defined by the study [S23] as a business model in which a large company rents space in their storage infrastructure to a smaller company or individual. It is a "good alternative for a small or mid-sized business that lacks the capital budget and/or technical personnel to implement and maintain their own storage infrastructure [S23]".

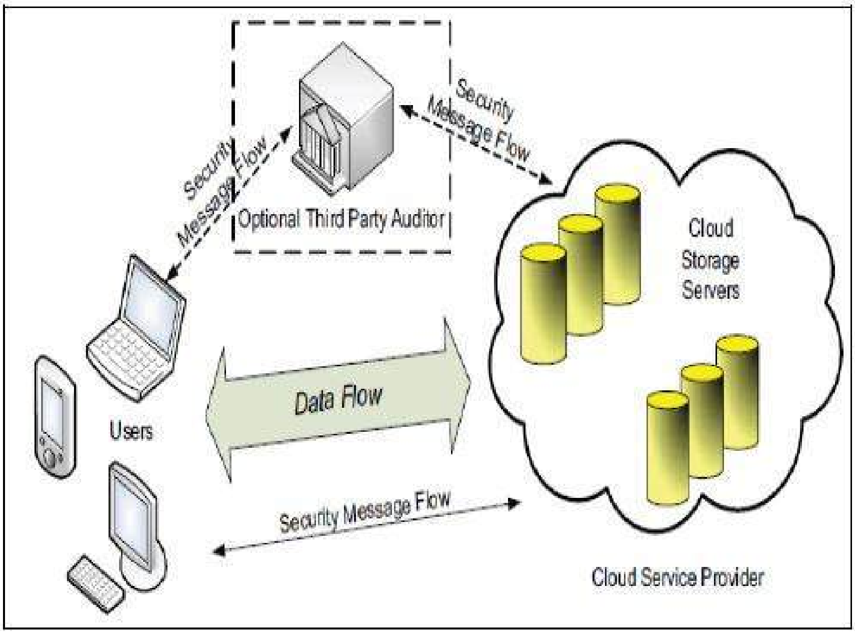


Figure 4.26, Cloud storage security and access method

As shown on the above figure, there are three ways to access cloud computing storage space: block-based, file-based, and through Web services. Block and file-based access improve greater performance, availability, and security. This model enhances security, availability, data protection, storage agility, performance, and scalability.

Researches in the study [S8] gave an architecture (BETaaS) for a platform providing *Building the Environment for the things-as-a-service* over a local cloud allowing in-site real-time data storage and processing while (energy efficiency, scalability, security, reliability) are improved.

Study [S11] deployed *Technology-as-a-service* that gives users access only to the resources they need for a particular task, which prevents them from paying for idle computing resources. Study [S53] provides *Elasticity-as-a-service,* an automated and optimized cloud resource allocation in federated cloud platform environments. Analyzing cloud computing security risks, study [S36] designs a multi-hierarchy, multi-level, elasticity and unified user interface secure cloud architecture on mobile internet (Figure 4.27) according to the principle of SeaaS (*Security as a Service*).

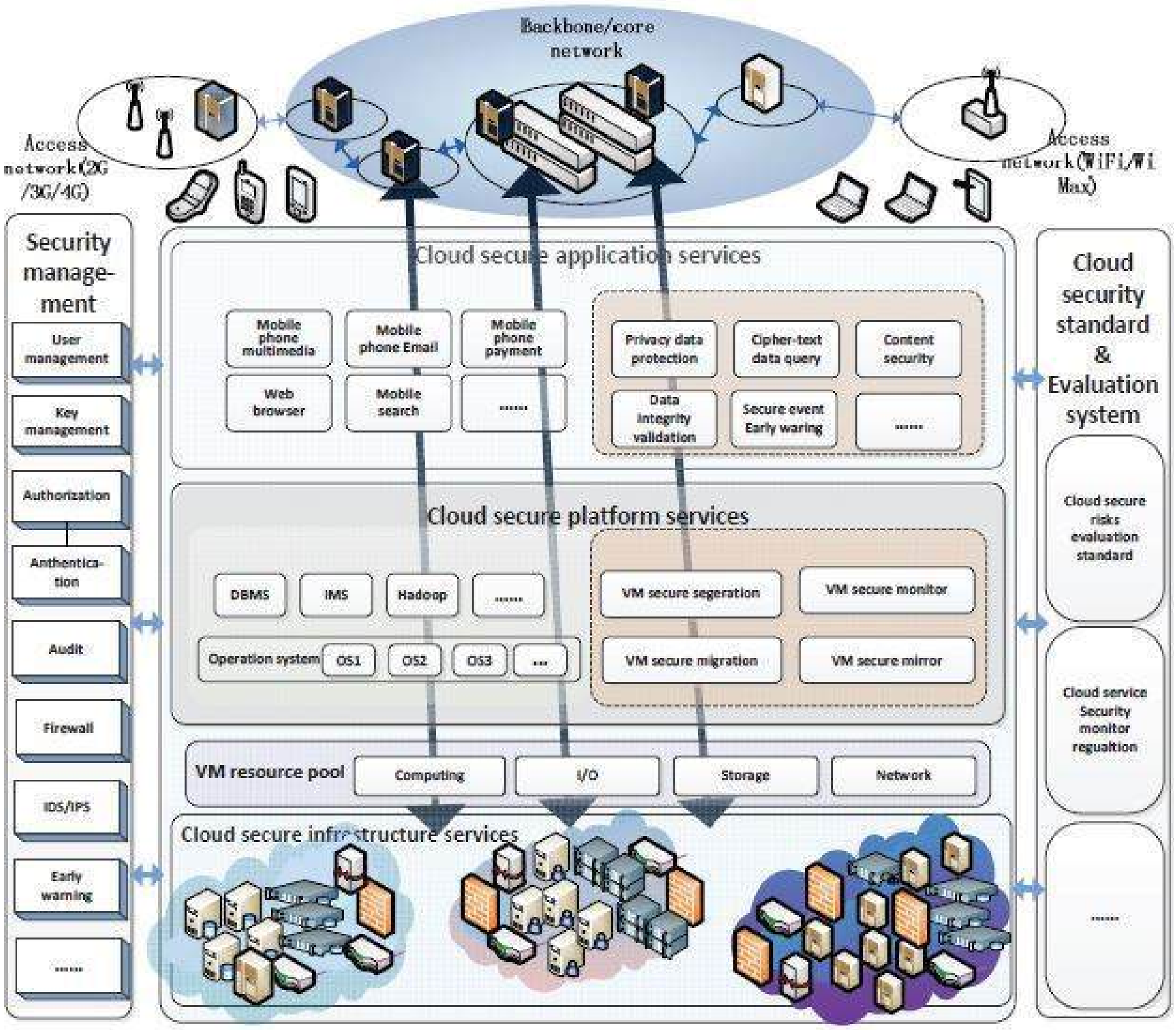


Figure 4.27, Cloud Computing secure architecture on mobile internet [S36]

The security level of cloud service for customers provided by the architecture can adapt to the various user requirements. Implementing this architecture is flexible to different scale system with different requirements. The idea of SeaaS is to [S36]: ensure the virtualization operation environment security of cloud computing platform, provide customized security services according to different requirements, implement risk assessment and security monitor on the running cloud computing platform, ensure cloud computing infrastructure security and construct trusted cloud services, preserve integrity and confidentiality for user private data.

However, according to the study [S49], today’s “as-a-service“ characterization of cloud providers fails to adequately categorize the architecture components. “The implementation of the stack is proprietary. Responsibilities of the lower layers are of no concern for the enterprise user, but the trade off is that the functionality cannot be customized or controlled [S49]”. Researches of the study [S49] bring the Cloud Reference Model which divides cloud-based application architecture into seven layers: application, transformation, control, instantiation, appliance, virtual and physical. Each of mentioned layers focuses IT functionality on supporting a specific concern and abstract details of other layers.

4.3.10. Autonomic Cloud Management Architecture

Challenges and Issues

Study [S31] addressed issues for implementing a dynamic cloud computing infrastructure such as: latency sensitivity, performance, scalability, reliability and security. Study [S13] defined some of the challenges in designing autonomic cloud computing architecture, and those are: quality of service, energy efficiency, security which includes confidentiality, availability and reliability. Study [S7] solved reliability and availability issue, in a way of enabling auto recovery (component level recovery and node level recovery), and dynamic component pool sizing. Challenges they will deal with in the future are: automatic software updates, automatic hardware updates, and time based profiles. Future work for the study [S13] is implementing more dynamic provisioning algorithms for improving security, and testing it in the real applications.

Proposed solutions

“Cloud service management is processes, activities and methods that are generated by cloud providers by taking cloud consumer perspective as a measure of service assurance [S40] “. Cloud service management is supported by three support services: architecture service, business support and operational support [S40]. “Current hypervisors do not provide adequate separation between application management and physical resource management [S31] “. As a solution to that concern, study [S31] proposed a reference model (Figure 4.28) for a network-centric data centre infrastructure management stack that enables dynamism, scalability, reliability and security in the industry.

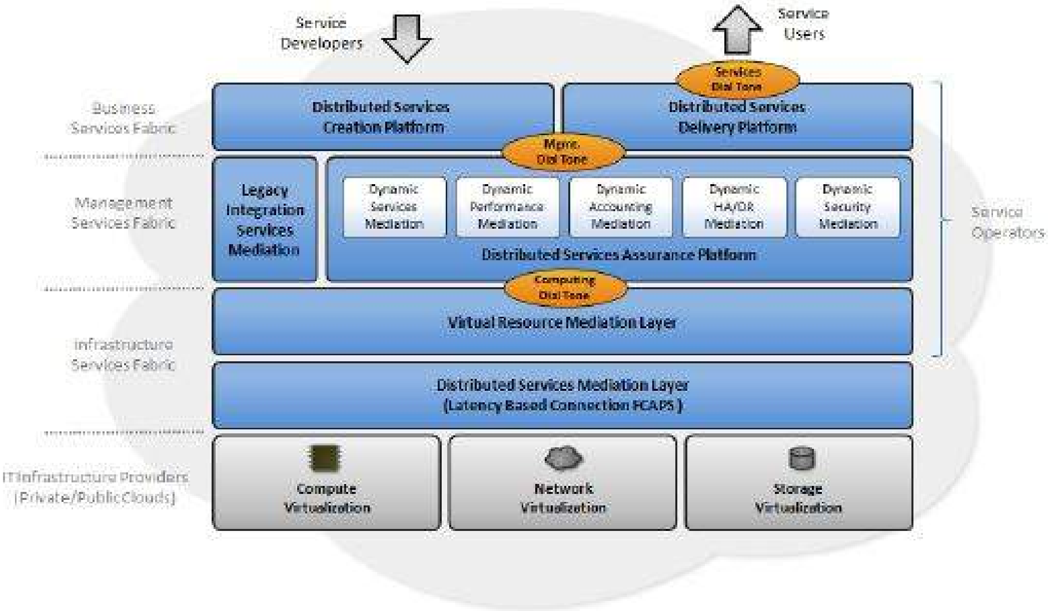


Figure 4.28, Proposed reference architecture model [S31]

The proposed model consists of five different levels:

- Infrastructure service fabric – it includes two pieces which together provide the basis for provisioning resources to all applications in the cloud:
  - Distributed services mediation – fault, configuration, accounting, performance and security (FCAPS) abstraction layer. It enables autonomous self-management of every resource in a network;
  - Virtual resource mediation layer – provides the ability to compose logical virtual servers;
- Distributed services assurance platform – it allows creation of FCAPS-managed virtual servers to allow the execution of applications;
- Distributed services delivery platform – it executes the application and defines the services dial tone in proposed model;
- Distributed services creation platform – it provides the tools in order for the developers to use for creating applications which can be composed, decomposed and distributed on virtual servers that are automatically created by the distributed services assurance platform;

- Legacy integration services mediation – it provides integration and support for existing applications.

Cloud Computing Energy-efficient Management

Proposed reference architecture of the study [S15] has the capability of cloud ecosystem management which provides the ability to manage lifecycle, evolution, interaction, relationship between the involved participants. Study [S13] proposed an autonomic Cloud manager (figure 4.29) with specific roles: application scheduler, energy-efficient scheduler, dynamics resource provisioning algorithms, and security and attack detection. Application scheduler assigns tasks in the application to resources for the execution based on user quality of service parameters and the cost for the service provider. Energy-efficient scheduler is ensuring energy utilization without compromising service level agreement and cost; applications need to be scheduled in resources in the way that their total energy consumption is minimized. Dynamic resource provisioning algorithms implement the logic for provisioning and managing virtualized resources in cloud environments based on the application scheduler. Security and attack detection implements all the checks to be performed in order to evaluate legitimacy of received requests.

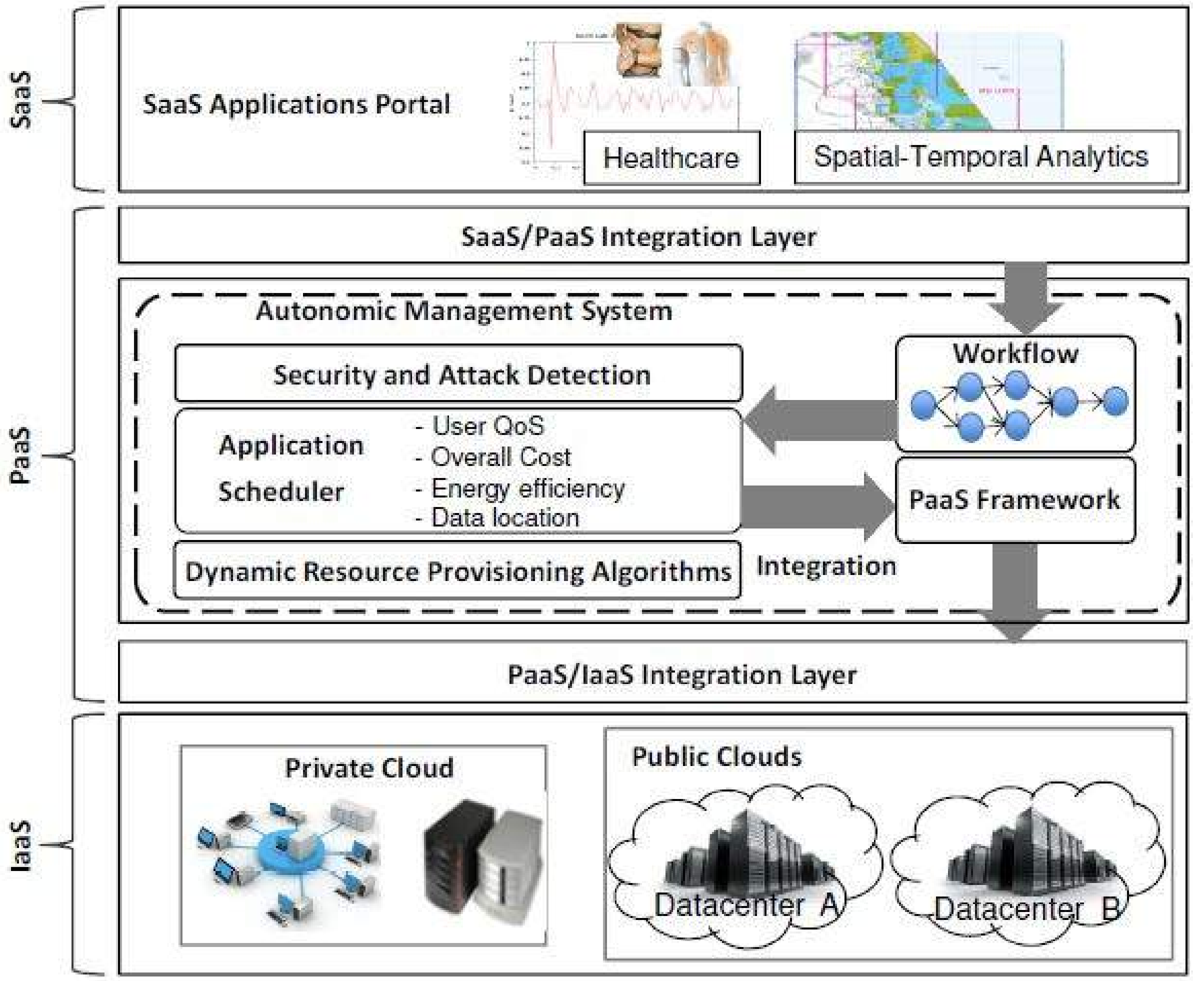


Figure 4.29, System architecture for autonomic Cloud management [S13]

“As Clouds are complex, large-scale, and heterogeneous distributed systems, management of their resources is a challenging task. They need automated and integrated strategies for provisioning of resources to offer services that are secure, reliable and cost-efficient [S13]“. Autonomic Cloud systems are self-regulating, self-healing, self-protecting, and self-improving [S13, S51].

Cloud Computing Autonomic Service Provisioning

Study [S7] proposed an *Always On* architecture that has the ability of auto recovery and dynamic component pool sizing. In order to design self-adaptable solution capable to react to unpredictable workload situations, study [S12] proposed an architectural model (Figure 4.30) for the autonomic service provisioning system. It also identified an autonomic lifecycle which contains of monitor phase, analyze phase, plan phase, execute phase and knowledge phase.

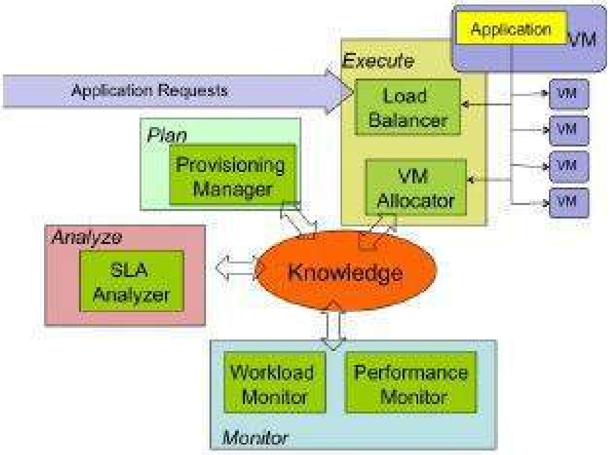


Figure 4.30, The autonomic service provisioning architecture [S12]

The proposed architecture can "be implemented in different ways depending on which components are deployed on the IaaS provider side and which on the autonomic service provisioning (ASP) side [S12]". They have proposed four different implementations:

- Extreme ASP control – the ASP has total control over all the components, included physical and virtual machine management;
- Full ASP control – the ASP has total control on the analyze and plan phases of the autonomic cycle, it avoids the higher cost of running a data centre, and it improves scalability and availability features;
- Partial ASP control – "this implementation is in between the need of the ASP to have total control over the components that perform the adaptation, and the opportunity to exploit the scalable and robust load balancing and monitoring functionality offered by IaaS providers [S12]";

- Limited ASO control – it can be implemented only with IaaS providers that offer auto scaling functionalities. The ASP is responsible to set of the scaling rules on the IaaS provider auto scaling component.

In the study [S51] an autonomic management system is presented. It is defined by a few control points: control parameter identification, system model, system input identification, model identification, model update, system decision type, prediction creation, coordinator creation, data measurement, managed system control, autonomic system control [S51]. Study [S68] gives a lifecycle for architecting cloud solutions in which the last phase is overall management of the cloud environment (managing the cloud services and managing all aspects of the customer-related account). Study [S4] used principles from security management and software engineering to design the distributed access control architecture.

4.3.11. 2-tiered vs. 3-tiered vs. Multi-tier Cloud Computing Architecture

Challenges and Issues

According to the study [S32], most of the existing cloud architectures are based on 2-tiered Cloud architectures. Study [S32] emphasized disadvantages of 2-tiered architectures connected with reliability and interoperability issue; and proposed 3-tiered cloud architecture. Study [S58] defined disadvantages of 3-tiered cloud architecture, and proposed multi-tier cloud application in order to improve cloud elasticity. Challenges of the study [S58] were enabling cost-aware scaling and workload-adaptive scaling. Study [S62] proposed multi-tier cloud application for improving portability comparing to the 3-tiered cloud architectures.

Proposed Solutions

Study [S43] proposed the two-tier SaaS architecture for group-tenant model which represents “a set of customers who have relative business and shared data and subscribe the SaaS application as the whole [S43] “. Study [32] showed the major disadvantages in 2-tiered cloud architecture and proposed 3-tiered cloud architecture (Figure 4.31) with improved reliability and interoperability. According to the study [S32] disadvantages of 2-tiered cloud architecture are:

- Non-availability of service - outages or lack of reliability;

- Vendor lock-in - Cloud should ensure data and application interoperability with data and application portability;
- Limited consumer choice - lack of interoperability. In 2-tiered cloud architecture user can not have different services from different providers.

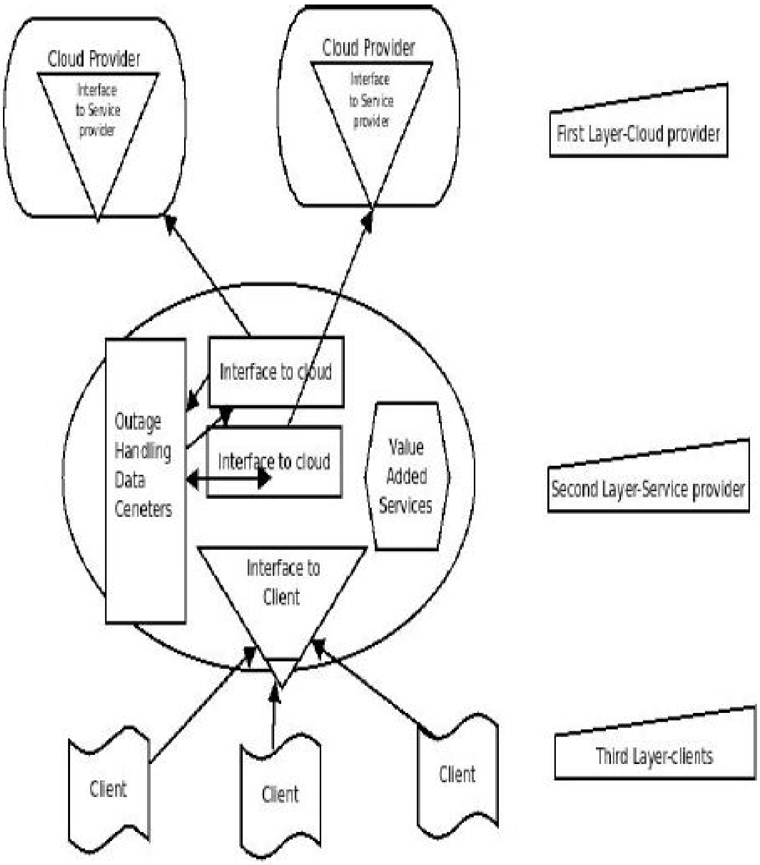

Cloud Provider
Interface to Service provider
Cloud Provider
Interface to Service provider
First Layer-Cloud provider
Interface to cloud
Outage Handling Data Ceneters
Interface to cloud
Value Added Services
Second Layer-Service provider
Interface to Client
Client
Client
Client
Third Layer-clients

Figure 4.31, Proposed architecture [S32]

Study [S32] improved reliability through outage handling data centres. When an outage occurs in the main servers, this data centres will take charge over it without user knowing about it. Standard interface ensures application portability, since the data is stored in the backup outage servers. Furthermore, the proposed architecture allows a user to subscribe to multiple cloud offerings through multiple cloud providers.

Study [S6] proposed 3-tiered Shared Disk Architecture for Cloud database (Figure 4.25) in order to improve availability and scalability.

However, “moving architecture from legacy or client-server to modern 3-tier architecture or web services will also mean that some services will be inherently slower. Cloud architecture developers must involve new and different techniques to handle that approach, for example asynchronous messaging techniques, which brings additional scalability to the system [S19]. “Traditional web 2.0 applications rely on a 3-tiered architecture. It is suitable mostly for applications with a predictable number of users, following a small number of usage patterns and reduces number of load spikes. It runs into problems with the need for high scalability and elasticity in modern web applications [S62] “. In a multi-tier application, servers are categorized into different categories according to their functionalities [S58]. Study [S62] proposes a new set of API for Cloud application development (multi-tiered architecture) which is integrated in an open-source *Cloudware, mOSAIC* to improve portability, availability, fault tolerance, maintainability (Figure 4.32).

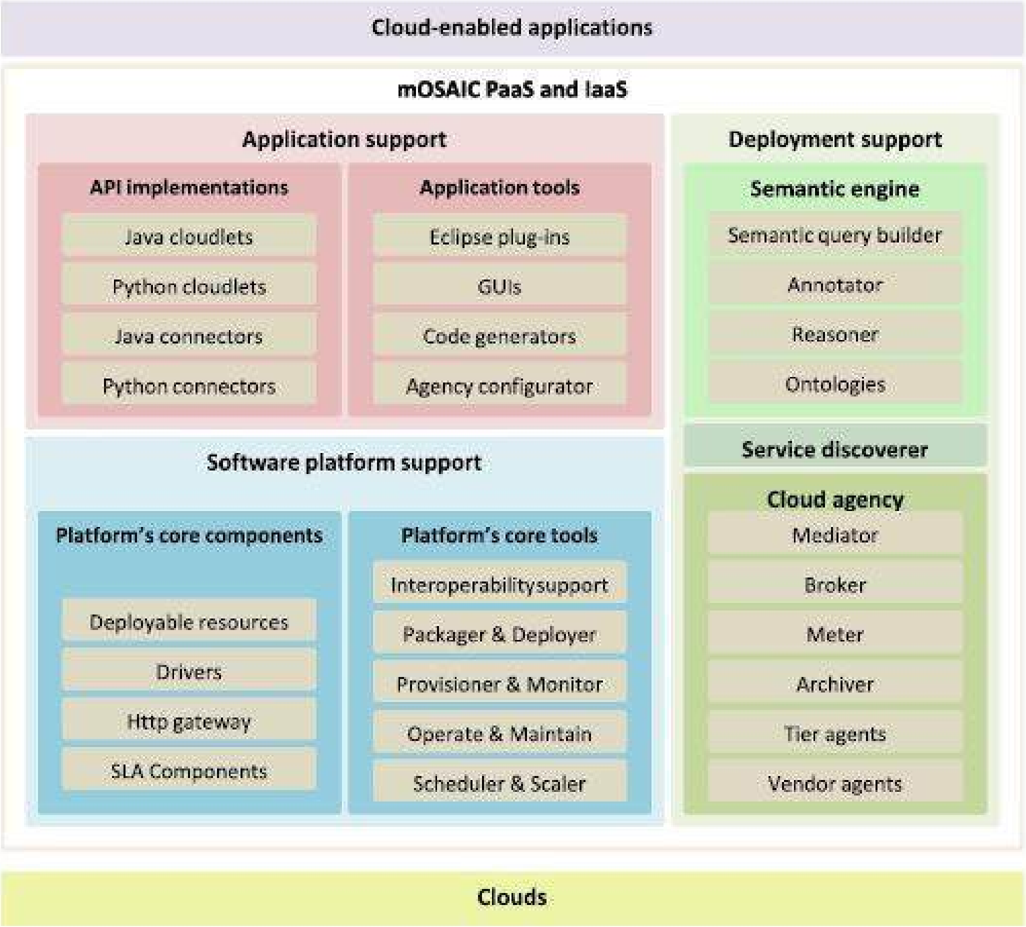


Figure 4.32, Components of mOSAIC’s architecture [S62]

Components which are grouped under the application and software platform support are responsible with the cloud portability. The other components are responsible to ensure elasticity at the applications component level. Proposed mOSAIC architecture gives the freedom for the developer to choose the programming language while architecting cloud applications, and it allows building own software for an application. While using mOSAIC architecture, Cloud Computing developer does not need to worry about issues specific to a certain Cloud provider [S62]. As mentioned in the third chapter, this research project was funded by the EU (European commission) [S1]. Proposed mOSAIC's APIs are improving the efficiency on the application side by removing restrictions like the requirement for synchronous communications [S1]. Study [S66] defines network architecture in which servers are organized in multi tier client-server architecture. "An intelligent storage management solution should be based on a multi-tier storage architecture [S23]".

Study [S58] proposed a cost-sensitive elastic scaling approach in multi-tier cloud applications (Figure 4.33).

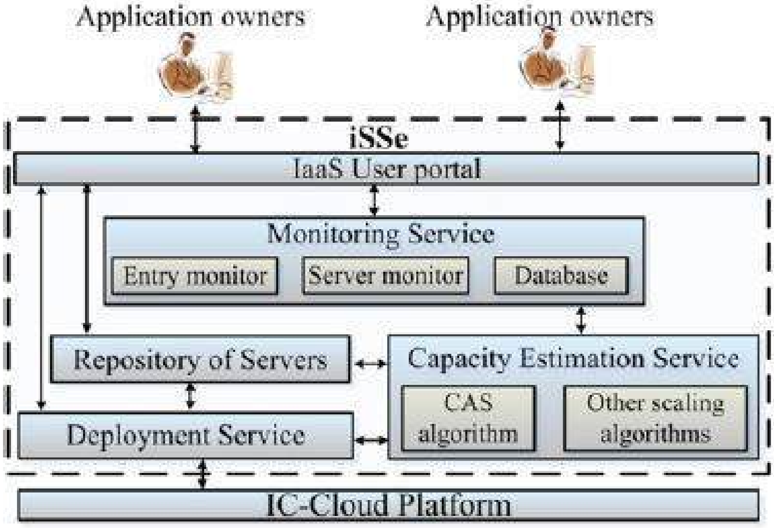


Figure 4.33, Proposed architecture of Imperial Smart Scaling engine (iSSe) [S58]

Proposed iSSe is a middleware between cloud providers and application owners. The IaaS user portal helps applications owners to provide services to application end users. All components working together help improving elastic scaling. Monitoring service monitors running applications using two types of monitor (entry monitor, and server monitor). Capacity estimation service estimates the number of servers to be scaled using the information in the database. The deployment service automatically implements the scaling of the servers. When one server is added, iSSe can automatically choose the cloud provider that offers the cheapest price [S58].

4.3.12. Other Architectural Paradigms and Solutions for Cloud Computing

In this section we will discuss about other architectural paradigms considering specific application domains in Cloud Computing, such as: cloud manufacturing, mobile cloud computing, Internet of Things, cloud-of-clouds, government Cloud Computing architectures, cloud governance, and scientific Cloud Computing architectures.

Cloud Manufacturing

Challenges and Issues

According to the study [S21], problems of networked manufacturing are: closed heterogeneous settlement mechanisms, lack of suitable resource supply mechanism, lack of unified resource management mechanism, lack of adaptability to external environment change, and fuzzy business mode.

Proposed Solutions

"In contrast to the conventional networked manufacturing approach, lessons learned from cloud computing can greatly improve the scale, agility and security of applications, and reduce costs for delivering new services. Introducing cloud computing to solve the problems of networked manufacturing can provide new ideas and opportunities [17]". Proposed cloud manufacturing vision *GetCM* (Figure 4.34) has the responsibility of providing reliability and flexibility based on cloud computing [S21]. It also defines benefits of GET cloud manufacturing paradigm in different perspectives:

- Provider perspective - publish and share manufacturing resources and businesses;
- Requester perspective – find and use the most appropriate manufacturing resources and businesses, improve efficiency and pay only for used;
- Business perspective – value chain positioning, get more market opportunities;

- Industry perspective – resource allocation, establishment of competition in the supply chain ecosystem;

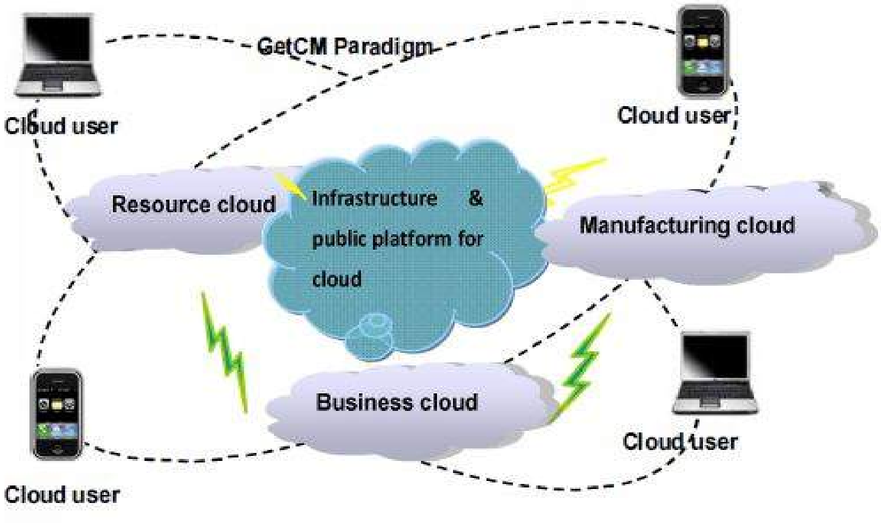


Figure 4.34, GetCM paradigm [S21]

GetCM paradigm includes five parts which are: resources cloud, business cloud, manufacturing cloud, infrastructure and public platform, and cloud user. Cloud users care categorized into three types: cloud manufacturing service provider, service requester and infrastructure and public platform provider. Cloud service requester can fix resources flexibility based on granularity synthesis [S21].

Mobile Cloud Computing

Challenges and Issues

"Introducing cloud computing into mobile internet leads to the changing of mobile internet's architecture, and arises many new security problems such as cross-domain data security and privacy protection; virtual running environment security and cross-domain security monitor [S36]". Study [S5] defined mobile cloud computing challenges such as:

- Network latency – mobile cloud computing face challenges from the network latency, due to the limited bandwidths of wireless networks;
- Various access mechanism – mobile cloud computing requires on-demand available wireless links with scalable bandwidth and energy-efficient devices;

- Handover latency – this is caused because of moving between different access technologies, e.g. from 3G to WiFi. It also happens due to the transition from one network domain to another;
- Elastic application model – limited resources on the mobile device do not allow offline updating;
- Security and privacy – mobile cloud computing could experience different attacks because of low power and slow CPU execution. Security can also be compromised due to authentication to different networks during motion. "A mobile user may need to re-authenticate with different providers or send authentication credentials to the home server which could create vulnerability to password attacks [S5]".

Study [S5] also defines challenges faced by the mobile cloud computing to enterprises:

- Identity security – how to secure the identity of the users if the mobile device is sold or given to another person;
- Administrative tasks - how to maintain the user privileges locally and on the cloud since different users may need a different type of the privileges on the application and the services;
- Auditing and Monitoring – how to track activities when users are changing different network boundaries. There is still no solution to capture, audit or monitor data used by the mobile users;
- Personal or official data – how to secure personnel of official data on the device or cloud when the device is lost or stolen;
- Service redundancy and Load balancing – if there is any failure in the hardware, the service must work and be delivered to the users. Many cloud service providers may not have their own data centres. In case of a large number of mobile users accessing the video, or data, it is very important to provide load balancing in order to provide services efficiently;
- Disaster recovery – there could be a possibility of losing data or service in case of failure since many cloud providers do not implement redundancy in the data centre.

Proposed Solutions

The growth in the wireless networking has led to the invention of smart mobile devices such as cell phones, notebooks and tablet. According to the ICT statistics from Cisco visual networking index, mobile data is expected to experience growth by 66% each year, because of that a mobile network and cloud computing formed a new computing model , Mobile Cloud Computing [S5].

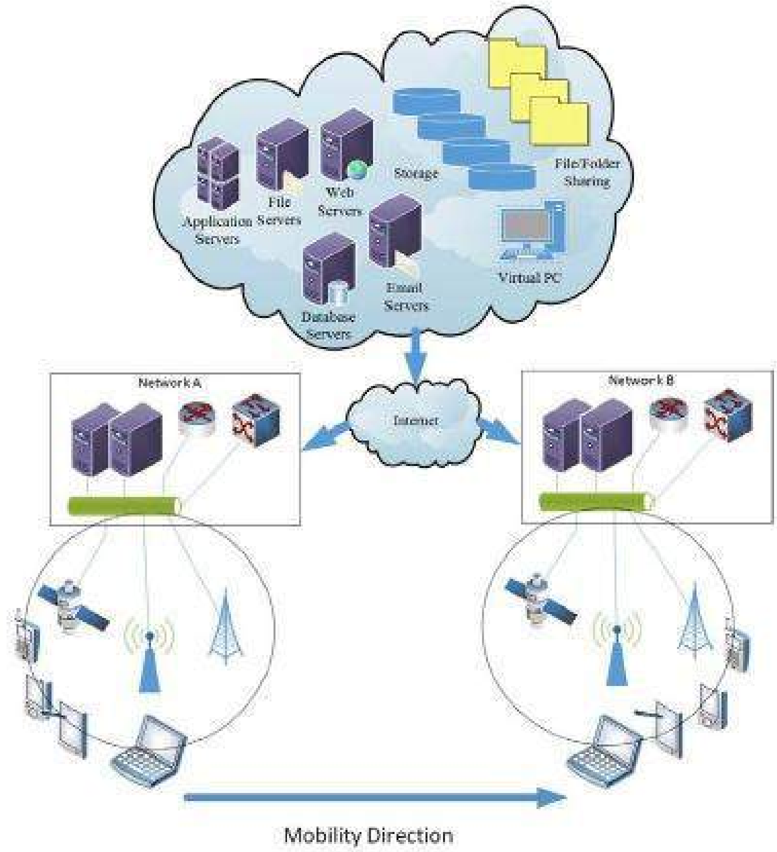


Figure 4.35, Mobile Cloud Computing [S5]

Architecture of mobile Cloud Computing (Figure 4.35) is based on four layers to ensure reliability, security and validity to support mobile users [S5]: access, management, virtual and physical layer. “Many different applications based on mobile cloud computing have been developed such as Google Gmail, Calendar, Groups, Maps, iCloud, Live Mesh etc. [S5]”. Five most important characteristics of mobile cloud computing [S5]: on-demand self service, broadband network access, resource pooling, rapid elasticity, and measured service. Study [S36] proposed multi-layer, multi-level, elastic, across platform cloud computing secure architecture based on mobile internet (explained in the sub-chapter 4.3.9).

Challenges and Issues

According to the studies [S69, S8], challenges of integrating cloud computing with Internet of Things are: computational infrastructure, data management, and networking. Study [S8] proposed BETaaS platform in order to improve flexibility, scalability, security, reliability. In their future work, they will allow connecting external platforms which would be able to provide capabilities as well as some sensors and communication hardware such as GPS, camera, 3G, Bluetooth etc.

Proposed Solutions

"The integration of Cloud computing into the Internet of Things presents a viable approach to facilitate Things (all computer-embedded objects which operate on the Internet) application development [S24]". The computation cloud is tightly coupled to the Internet of Things [S69]. The difference between Cloud-based Internet of Things from conventional Internet of Things is the ability to develop, deploy, run, and manage Things applications online by Cloud [S24], study proposed CloudThings architecture. Study [S8] proposed BETaaS (Building the Environment for the Things as a Service), in order to improve flexibility, energy efficiency, scalability, security, and reliability (Figure 4.36).

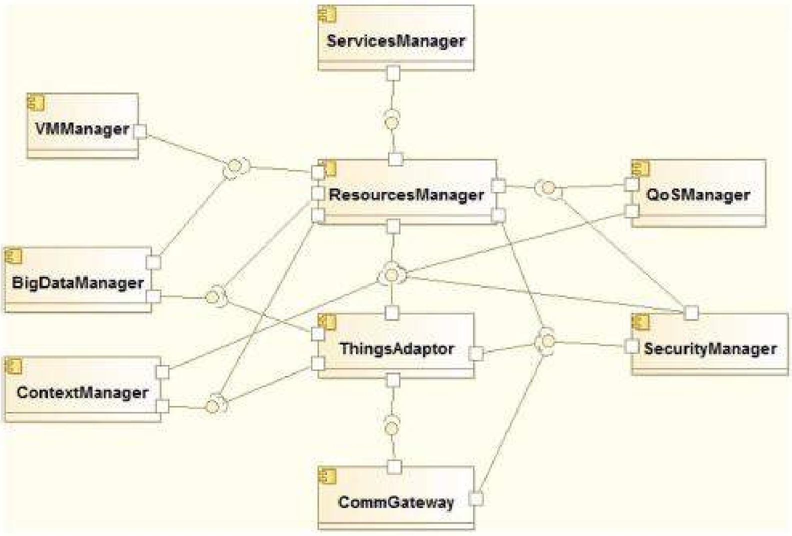


Figure 4.36, BETaaS proposed architecture [S8]

- The service manager is used for managing services and applications in the BETaaS Platform. It is interacting with the resource manager for allocating the resources as well as other components which have impact in aspects that can affect application execution.
- The resource manager decides how to manage the available resources. It is responsible of creating the local cloud of things in the BETaaS Instance and it guarantees scalability.
- The Things adaptor represents the bridge between the things and the BETaaS capabilities related to the things. It provides data to several components which depend on these data in order to run properly.
- The Big data manager provides advanced storage capabilities for large amounts of data coming from things and their monitoring.
- The VM manager creates and removes lightweight VMs which can be used for performing processing operations or providing isolated storage.
- The QoS manager retrieves information about the BETaaS instance and defines the QoS level. It maintains QoS policies which may influence the resources management performed in the BETaaS instance.
- The Context manager provides things context information.
- The Security manager provides trust calculation and other basic services used at different levels.
- The CommGateway communicates directly with things. It provides flexibility and evolution capability to the platform.

Study [S69] discussed about TransCloud infrastructure considerations which focuses on problems such as: ensuring that cloud users can run applications wherever they have access, ensuring that execution of remote queries is done safely for both remote user and data host, and designing a simple, network-aware architecture for geographically distributed heterogeneous data.

Cloud-of-clouds

Challenges and Issues

Study [S39] defined security and dependability concerns faced by existing Cloud Computing architectures such as:

- Dependability and security issues cannot be solved only by the application layer, currently they require security-specific solutions to be provided at lower layers (infrastructure and platform) of the cloud architecture;
- Some IaaS and PaaS approaches for achieving resilience can make migration or interoperation difficult and expensive, creating vendor-lock in;
- High-resilience objectives may be compromised in solutions of single points of failure (e.g., common or related management and trust domain).

Proposed Solutions

“Large cloud providers attempt to achieve resilience by developing several, differently located, cloud subsystems, but they still remain under a single management and trust domain, with regard to common-mode and malicious faults [S39]”. Cloud-of-clouds paradigm is proposed way of the study [S39] to achieve cloud computing resilience. “The cloud-of-clouds paradigm extends the cloud concept, by leveraging the availability of multiple or federated cloud environments to create diverse ecosystems, by letting users to self-organize the way they use multiple cloud computing offerings [S39]”. In order to solve mentioned challenges and issues, study [S39] proposed TCloud architecture (Figure 4.37). The idea of TCloud is to build data centres with trusted computing technology- enabled servers that can provide advanced security properties [S1]. The proposed architecture provides automated computing resilience against attacks and accidents, and avoids single points of failure. In order to ensure this, algorithms such as Byzantine fault tolerance[2] and proactive recovery are addressed in TClouds. These mechanisms are transparent and offer cloud-of-clouds abstractions to the higher-level users, while hiding the complexity of managing redundancy and diversity. The proposed architecture also preserves legacy needs while enabling a diverse ecosystem. “The architecture should accommodate multiple resilient Cloud Computing deployment alternatives, in order to be successful [S39]”. This research project was funded by the EU (European commission) [S1].

[2] Byzantine fault tolerance's objective is to be able to defend against Byzantine failures which are an arbitary failures that occur during the execution of an algorithm by a distributed system [24]

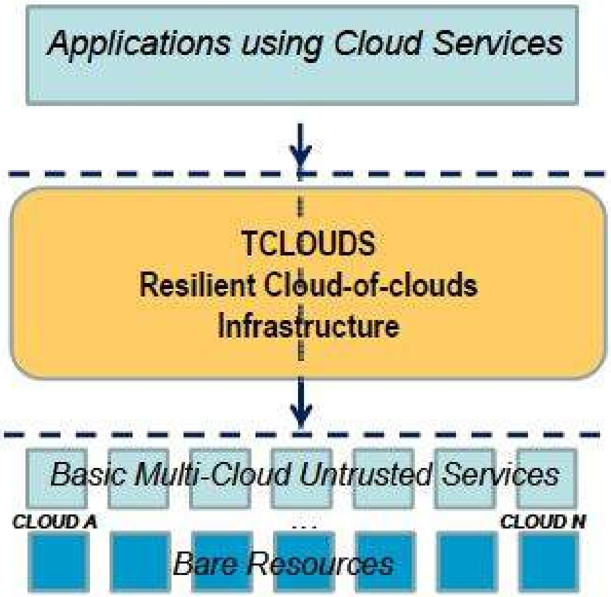

Applications using Cloud Services
TCLOUDS
Resilient Cloud-of-clouds
Infrastructure
Basic Multi-Cloud Untrusted Services
CLOUD A
...
CLOUD N
Bare Resources

Figure 4.37, Proposed TCloud architecture [S39]

Government Cloud Computing Architectures

Challenges and Issues

According to the study [S19] opportunity that governments around the world have with the Cloud Computing are achieving an optimized, cost-effective, government-wide information technology infrastructure that supports agency business, while providing reliability and security in service. “There is obvious need for the government to transform its Information Technology infrastructure by virtualizing data centres, consolidating data centres and operations, and adopting a Cloud Computing business model [S19]”.

Proposed Solutions

For achieving global competitiveness, governments in developing countries are implementing Cloud Computing technologies to enable their countries participate in the current ICT revolution [S61]. “There is a number of the process and technical challenges and issues that a government should carefully consider prior to designing or using a cloud computing based solution or considering a use of cloud computing resources and capacity for government need [S19]”.

Microsoft Government Private Clouds are dedicated cloud environments that address data security, privacy and regulatory requirements especially for the Government [S19]. The U.S. government estimates its IT spending on migration to cloud computing solutions at $20 billion for 2010 [S45]. Study [S19] defined four types of government Cloud Computing solutions, depending on the needs of the organization, and the level of risks caused by public and private cloud:

- Government Public Cloud (for Central/Federal Government ministries/agencies and LRG) – from technology viewpoint, this is the easiest solution for implementing Cloud Computing in the government, but has the highest risks of privacy, control and security attacks. It reduces total costs of ownership by removing the need for IT infrastructure management, and it provides fast, flexible and economical way to expand service availability to users;
- Government Private Cloud Dedicated (for Central/Federal Government ministries/agencies and LRG) – technologies and services are hosted in vendor data centres, but the data centre is located in country due to the security and privacy issues;
- Government Private Cloud Self-Hosted (for Central/Federal Governments that are providing shared services) – government can achieve all benefits of Cloud Computing while maintaining improved level of control and security. Only a government entity may host a government data centre. This solution enables governments to optimize their own data centres, and to operate and provide cloud services as a part of their own shared services;
- Government Private Cloud Hosted (for Central/Federal Government ministries/agencies and LRG that are outsourcing IT) - solution where data security requires that the private cloud must be hosted in country and international bandwidth is not sufficient to host the private cloud in another country.

According to the study [S45], the comparison of different Cloud services and their providers will become of high relevance to the government. Study [S61] offered the model for an e-education which gives the government responsibility for providing infrastructure and e-education services to educational institutions in the country. “Adopting Cloud Computing for education offers huge cost savings for governments through data centre consolidation, aggregation of demands, and multi-tenancy [S61]”.

Cloud Governance

Challenges and Issues

According to the study [S63], security issues with data availability issues are the domain of the IT governance. Future challenges will be as follows: “the possible relationships need to be developed from the point of view of every relevant perspective. The compliance paradigm needs to be more specified and extended with the capability of providing specific recommendations based on business or technological side, and the development of the supporting tools needs to be improved [S63]”.

Proposed Solutions

“The paradigm of extending traditional IT governance to cloud computing is denoted as cloud governance [S63]”. Study [S63] offered a governance model (Figure 4.38) for enterprise architecture governance. Both cloud user organizations and cloud provider organizations can benefit from this proposed model. Cloud user organizations can establish their own governance processes based on the tools of the model, and cloud providers can ensure better service between their offerings and the requirements of the organization of their potential and current customers.

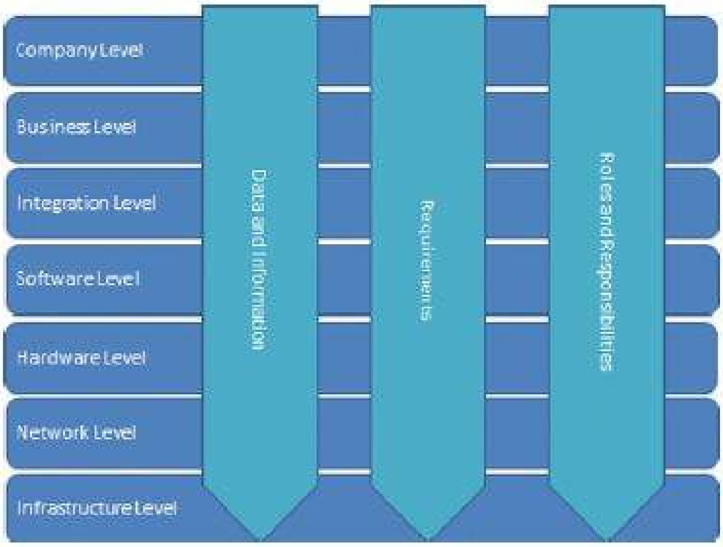


Figure 4.38, Proposed architectural model [S63]

Proposed model consists of seven horizontal (company, business, integration, software, hardware, network, and infrastructure) and three vertical (data and information, requirements, roles and responsibilities) levels.

Scientific Cloud Computing Architectures

Challenges and Issues

Studies [S64, S50] while architecting scientific Cloud Computing architectures, encountered these issues: scalability, flexibility, and modularity. Study [S48] addressed security as the main issue while architecting cloud application. Automatism, scalability and interoperable security are defined as main issues by the study [S29]. As future challenges, study [S48] plans to improve incorporate high-availability mechanisms into proposed architecture.

Proposed Solutions

By increasing the data traffic growth, science analysis and processing complexity is also growing exponentially [S50]. The benefits of running scientific applications on Cloud [S50]:

- The scale of scientific problems can be greatly increased with limited resource sharing, cloud platform can offer sufficient amount of computing resources as well as storage space;
- Application deployment can be made flexible and convenient;
- Cloud-based science applications can get resources allocated dynamically;
- Cloud Computing gives the opportunity for improving the performance/cost ratio of larger-scale scientific problems.

However, "scientists and medical researchers are still searching for a simple cloud based architecture that enables secure collaboration and sharing of distributed datasets [S48]". Study [S64] presented Cumulus project (Figure 4.39). It provides virtual computing platforms for scientific and engineering applications.

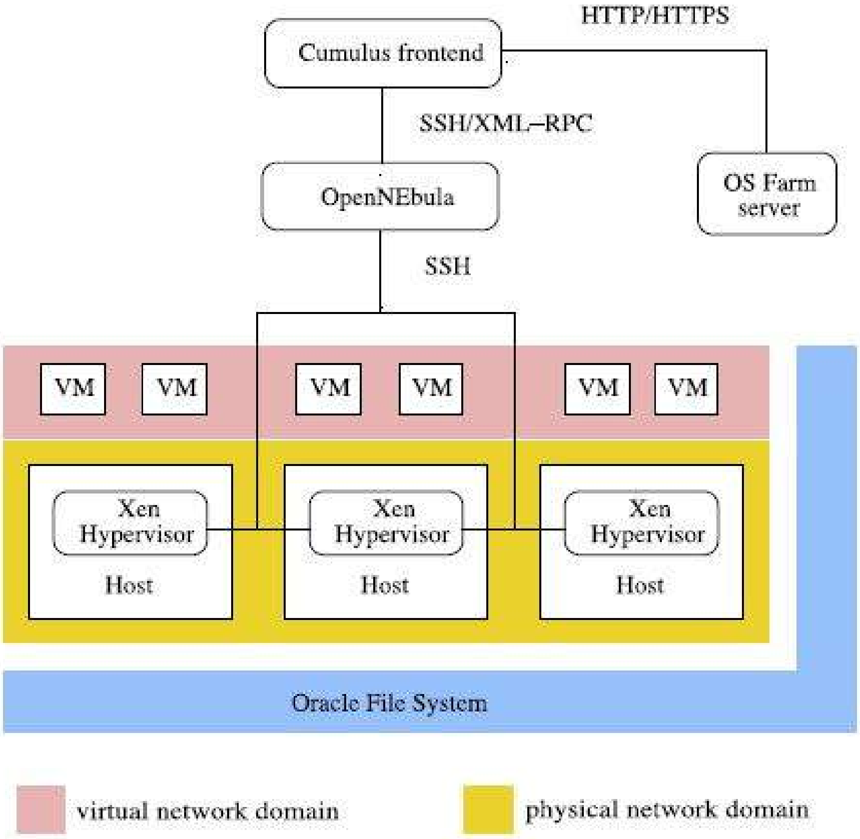


Figure 4.39, Proposed Cumulus architecture [S64]

Proposed architecture improves scalability and keeps the autonomy of compute centres:

- The Cumulus frontend does not depend on any specific local virtualization management system, it uses a third party software;
- Compute centres inside clouds define their own resource management policies;
- The Cumulus frontend can delegate user requirements to other clouds.

Study [S50] presented the CloudDragon scientific computing Cloud platform (Figure 4.40).

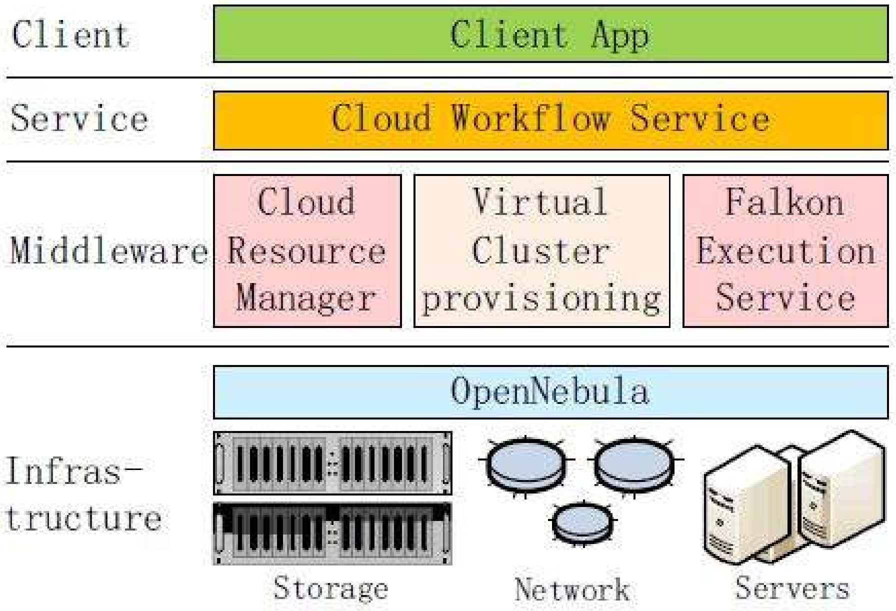


Figure 4.40, Proposed CloudDragon architecture [S50]

The architecture consists of four layers:

- Client layer – client-side development and submission tool for application specification is provided;
- Service layer – cloud workflow service based on swift management system is presented as a gateway to the cloud platform;
- Middleware layer – a few components (cloud resource manager, virtual cluster provisioner, and a task execution service) are integrated in order to bridge the gap between the service layer and the infrastructure layer;
- Infrastructure layer – cloud data centre resources such as servers, network and storage are managed.

Study [S48] designed CloudDRN (Figure 4.41) for securely sharing research data in a cloud, and allowing an enterprise to participate in CloudDRN via server running within its enterprise.

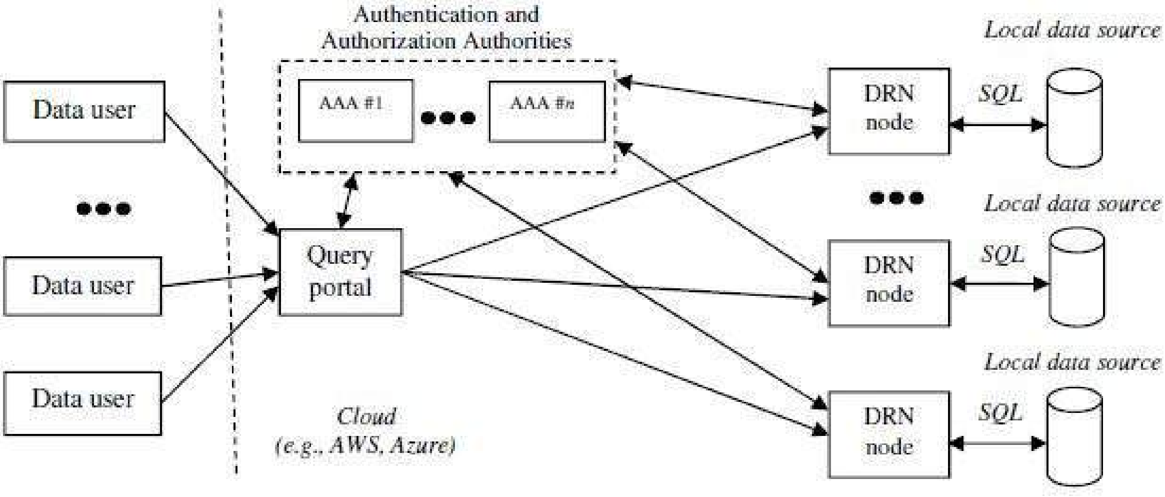


Figure 4.41, Proposed architecture of a Cloud Distributed Research Network [S48]

For the improvement of data efficiency, easy accessibility, strong reliability and always availability, study [S17] proposes a solution to make six local sub clouds on the basis of six sub-continents instead of one Global cloud. If one sub cloud experiences failure, any client from anywhere in the world has the right to use another sub cloud under the ‘failover’ techniques [S17]. Study [S46] introduced and proposed Cloud@home middleware for enhancing interoperability among clouds. Study [S29] defined how to make up an interoperable heterogeneous cloud environment with a proposed three-phase (discovery, match-making, and authentication) cross-cloud federation model.

Chapter 5 - Discussions

In this chapter, we will discuss about selected studies in a way of giving complete overview of them. Overview includes level of maturity, and which specific quality attribute concerns studies were focusing on. Later on, we will validate our done research and give a conclusion.

5.1. Level of Maturity of Selected Studies

While doing the research we were investigating which of the proposed architectures (if there was one) in the study has been implemented or is just an idea. In the table 5.1, is presented the complete overview of the level of maturity of our selected list of studies. It also shows which of the main quality attributes were studies focusing on and if they solved the issue or not.

		Quality attributes										Maturity level	
		Se	Sc	A	Pr	Po	E	I	CE	R	F	+/-	Explanation
ACM	S1	🟩	🟥	🟩	🟥	🟩	🟥	🟥	🟥	🟥	🟥	🟩	Horizon 2020
	S2	🟥	🟥	🟥	🟥	🟥	🟩	🟥	🟩	🟥	🟥	🟩	Experiments on real Cloud applications
	S3	🟥	🟥	🟥	🟩	🟥	🟥	🟥	🟥	🟥	🟥	🟥	Not implemented
IEEE	S4	🟩	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	Not implemented
	S5	🟩	🟩	🟥	🟥	🟥	🟥	🟥	🟥	🟩	🟥	🟥	Not implemented
	S6	🟩	🟩	🟩	🟥	🟥	🟥	🟥	🟩	🟥	🟥	🟥	Not implemented
	S7	🟥	🟥	🟩	🟥	🟥	🟥	🟥	🟥	🟩	🟥	🟥	Not implemented
	S8	🟩	🟩	🟥	🟥	🟥	🟥	🟥	🟥	🟩	🟩	🟥	Not implemented
	S9	🟥	🟩	🟩	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟩	Project ASMONIA
	S10	🟩	🟩	🟩	🟥	🟥	🟩	🟥	🟥	🟥	🟥	🟩	Implemented solution
	S11	🟩	🟥	🟥	🟥	🟥	🟩	🟥	🟥	🟥	🟩	🟥	Not implemented
	S12	🟥	🟩	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟩	🟩	Implemented solution
	S13	🟩	🟥	🟩	🟥	🟥	🟥	🟥	🟩	🟩	🟥	🟥	Not implemented
	S14	🟥	🟩	🟥	🟥	🟥	🟩	🟥	🟥	🟥	🟩	🟥	Not implemented
	S15	🟥	🟩	🟥	🟥	🟥	🟥	🟩	🟥	🟥	🟥	🟥	Not implemented
	S16	🟩	🟩	🟥	🟥	🟥	🟥	🟩	🟥	🟥	🟩	🟥	Not implemented

	Quality attributes											Maturity level	
		Se	Sc	A	Pr	Po	E	I	CE	R	F	+/-	Explanation
	S17	🟥	🟥	🟩	🟥	🟥	🟥	🟥	🟥	🟩	🟩	🟥	Not implemented
	S18	🟩	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	Not implemented
	S19	🟩	🟩	🟥	🟩	🟥	🟥	🟥	🟥	🟥	🟥	🟥	Not implemented
	S20	🟥	🟥	🟥	🟥	🟥	🟩	🟥	🟥	🟥	🟥	🟥	Not implemented
	S21	🟥	🟥	🟥	🟥	🟥	🟩	🟥	🟥	🟩	🟩	🟥	Not implemented
	S22	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	Not implemented
	S23	🟥	🟩	🟥	🟥	🟥	🟥	🟥	🟩	🟥	🟥	🟥	Not implemented
	S24	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟩	Implemented solution
	S25	🟩	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	Not implemented
	S26	🟩	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	Not implemented
	S27	🟩	🟩	🟥	🟥	🟥	🟥	🟥	🟩	🟥	🟥	🟥	Not implemented
	S28	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟩	🟥	🟩	🟥	Not implemented
	S29	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	Not implemented
	S30	🟥	🟥	🟥	🟥	🟥	🟥	🟩	🟥	🟥	🟥	🟩	Implemented solution in GEYSERS project
	S31	🟩	🟩	🟥	🟥	🟥	🟥	🟥	🟥	🟩	🟥	🟩	Implemented solution
	S32	🟥	🟥	🟥	🟥	🟥	🟥	🟩	🟥	🟩	🟥	🟥	Not implemented
	S33	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	Not implemented
	S34	🟩	🟩	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟩	🟥	Not implemented
	S35	🟥	🟩	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	Not implemented
	S36	🟩	🟥	🟥	🟥	🟥	🟩	🟥	🟥	🟥	🟥	🟥	Not implemented
	S37	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟩	Given initial prototype and experiment
	S38	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	Not implemented
	S39	🟩	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟩	Given prototype
	S40	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	Not implemented
	S41	🟥	🟩	🟥	🟥	🟥	🟥	🟩	🟥	🟥	🟥	🟥	Not implemented
	S42	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	Not implemented
	S43	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟩	🟥	🟥	🟩	Implemented solution, case study
	S44	🟩	🟥	🟥	🟩	🟥	🟥	🟥	🟥	🟥	🟥	🟩	Implemented solution
	S45	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	Not implemented

		Quality attributes										Maturity level	
		Se	Sc	A	Pr	Po	E	I	CE	R	F	+/-	Explanation
	S46	🟥	🟥	🟥	🟥	🟥	🟥	🟩	🟥	🟥	🟥	🟥	Not implemented
	S47	🟩	🟥	🟥	🟥	🟥	🟥	🟩	🟥	🟥	🟥	🟥	Not implemented
	S48	🟩	🟩	🟩	🟥	🟥	🟥	🟥	🟥	🟩	🟥	🟩	Evaluation of results in AWS: CloudCHORDS
	S49	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟩	Case study: media transcoding and wiki portal
	S50	🟥	🟩	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟩	Case study
	S51	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	Not implemented
	S52	🟩	🟩	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟩	Given experiment and results
	S53	🟥	🟥	🟥	🟥	🟥	🟩	🟥	🟥	🟥	🟥	🟥	Not implemented
	S54	🟩	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	Not implemented
	S55	🟩	🟩	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟩	🟩	Case study
	S56	🟩	🟥	🟥	🟥	🟥	🟥	🟥	🟩	🟩	🟩	🟥	Not implemented
SCIENCEDIRECT	S57	🟥	🟩	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟩	Given experiment
	S58	🟥	🟩	🟥	🟥	🟥	🟩	🟥	🟥	🟥	🟥	🟩	Example in an e-commerce application
	S59	🟥	🟩	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟩	🟩	Performance evaluation
	S60	🟩	🟥	🟥	🟩	🟥	🟥	🟥	🟥	🟥	🟥	🟥	Not implemented
	S61	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	Not implemented
	S62	🟥	🟩	🟩	🟥	🟩	🟥	🟥	🟥	🟩	🟥	🟩	Implemented solution
	S63	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟩	Given experiment
	S64	🟥	🟩	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟩	🟩	Given experiment
	S65	🟩	🟥	🟥	🟩	🟥	🟥	🟥	🟥	🟥	🟥	🟥	Not implemented
SCOPUS	S66	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	Not implemented
	S67	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	Not implemented
	S68	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	Not implemented
	S69	🟩	🟥	🟩	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟩	Implemented solution
SNOWBALL	S70	🟥	🟥	🟥	🟥	🟥	🟩	🟥	🟥	🟥	🟥	🟥	Not implemented
	S71	🟥	🟥	🟥	🟥	🟥	🟩	🟥	🟥	🟥	🟥	🟥	Not implemented
	S72	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	Not implemented

Table 5.1 Selected list of studies, maturity level, and quality attributes challenges

Quality attributes in the table 5.1: Se (security), Sc (scalability), A (availability), Pr (privacy), Po (portability), E (elasticity), I (interoperability), CE (cost efficiency), R (reliability), F (flexibility)

In the table 5.1, green colour indicates that the study and its proposed architecture solved some specific concern; if the field is red, it means that the concern was not solved. We also included maturity level column into the table which defines whether the proposed solution was implemented (e.g. experimented with, included in case study etc.); or not.

High level of maturity in selected list of studies, and a proof of doing the research method properly, presents the result in the table 5.1 which shows that 24 out of 72 studies have implemented their solution, and 59 out of 72 succeeded to solve one or more quality attribute issue. Those numbers are high taking into a consideration the novelty of the topic.

5.2. Evolution of the Cloud, What Is Next?

Study [S29] defines evolution of the cloud computing market in three stages:

- "Monolithic" – now;
- "Vertical Supply Chain" - some cloud providers will leverage cloud services from other providers;
- "Horizontal Federation" - smaller, medium and large providers will federate horizontally themselves to gain an efficient use of their assets;

OpenNebula, Nimbus and Eucalyptus frameworks are almost at the stage 2 [S29]. "Since, a standard guideline of how a cloud architecture should be made does not exist, each cloud middleware may be deeply different from each other which makes the transition at the stage three very difficult to achieve [S29]".

"Over the next decade, widespread availability of massive computation- the Computation Cloud- will give to everyone the ability not only to look up what someone knows, but to discover things that no one knows [S69]". Developers would have the knowledge to design their next generation system to be deployed into cloud computing, in general, the emphasis should be in the horizontal scalability to thousands of virtual machines on a single virtual machine [S70].

5.3. Validity

This master thesis was written by two students. The systematic literature review was done twice by each student separately (for reducing bias); the results of the research were compared and summed in the end. Through whole phases (both researching and writing) assigned mentor observed and helped. In the first phase of doing a systematic literature review, planning, we used the search terms ‘cloud architecture’ OR ‘cloud architecting’. While screening the articles we realized that the term 'cloud design' is also used very often in the same manner as the previous terms. Therefore, the term of 'cloud design' was also added to the main search terms and the additional reference scanning was done. Although doing the systematic literature review twice, there might be some missed articles. Explanation for this could be that researches used different terms from ‘architecting’ and/or ‘designing’ cloud applications in their studies. Our selected list of studies consists of studies found by different search libraries (defined in sub-chapter 3.1), summed with studies found by using snowball effect. Combining these two methods for getting to the final list of selected studies ensures us that we covered relevant articles.

Conclusion

How to architect for the cloud based applications concerns both academia and industry. In this Master thesis, we conducted a systematic literature review with the main research question: How to design an application for the cloud. The research method of systematic literature review gave us 72 studies which we analyzed and selected information in order to answer on the main research question. Based on the selected list of studies, security is the main concern for enterprises to adopt a cloud solution. Next to the security, we analyzed quality attributes such as availability, scalability, privacy, interoperability, elasticity etc. Each quality attribute concern was defined, and proposed solutions regarding the concern are given. Even though becoming a cloud provider is really expensive and takes a lot of resources, there are many advantages shown in this thesis to become one due to the rapid grow of interest in Cloud Computing. We have also shown architectural views and challenges from different Cloud stakeholders' perspectives in order to help Cloud developers for better understanding the stakeholder needs. As mentioned, security is on top of the list of concerns; but for the cloud consumers, information privacy is what worries them the most. Some current cloud solutions do not allow cloud consumers migrating to another cloud services, or retrieving their data once stored. Although the most existing cloud applications are based on the 2-tiered architecture, studies we presented show lack of its abilities and proposed new architectural solutions (3-tiered and multi-tiered architectures). Results and findings in this thesis will help industry and academia, as well as Cloud Computing architects since we have summed up issues connected to the existing cloud architectures such as: lack of portability, scalability, availability, security and privacy issues etc.; and selected guidelines for architecting a new cloud which is able to solve some specific quality attribute concerns. The main contribution of the thesis is the identification of main concerns when architecting for the cloud and existing architecture solutions for coping with these different concerns.

References or Bibliography

[1] Irena Bojanova, “Analysis of Cloud Computing Delivery Architecture Models“, 2011 Workshops of International Conference on Advanced Information Networking and Applications, 2011

[2] Mohammad Hamdaqa, Ladan Tahvildari, “Cloud computing uncovered – a research landscape“, Advances in Computers, Volume 86, 2012

[3] An Oracle White Paper, “Cloud Reference Architecture“, Oracle Enterprise Transformation Solutions Series, 2012

[4] www.techopedia.com/definition/133/cloud-provider

[5] Keele University, “Guidelines for performing Systematic Literature Reviews in Software Engineering“, EBSE Technical Report, Version 2.3, 2007

[6] http://www.nielsen.com/us/en/newswire/2009/twitters-tweet-smell-of-success.html

[7] https://www.ibm.com/developerworks/cloud/library/cl-cloudstorage/cl-cloudstorage-pdf.pdf

[8] Vivek Nallur, Ramu Bahsoon, Xin Yao, “Self-optimizing architecture for ensuring quality attributes in the cloud“, IEEE/IFIP WICSA/ECSA, 2009

[9] Xiaoni Wang, “The research of a resource-aware cloud computing architecture based on web security“, 2012

[10] Julie Bort, “Yahoo builds ultimate private cloud“, Network World, July 19, 2011

[11] http://www-935.ibm.com/services/us/gbs/thoughtleadership/ibv-power-of-cloud.html

[12] Yashpalsinh Jadeja, Kirit Modi, „Cloud computing-concepts, architectures and challenges“, 2012 International Conference on Computing, Electronics and Electrical Technologies (ICCEET), 2012

[13] Peeyush Mathur, Nikhil Nishchal, “Cloud Computing: New challenge to the entire computer industry, 1st International Conference on Parallel, Distributed and Grid Computing“ (PDGC 2010), 2010

[14] http://www.techopedia.com/definition/4927/web-collaboration

[15] http://www.yourdigitalspace.com/2010/10/the-amount-of-data-generated-and-consumed-on-the-internet/

[16] http://www.idc.com/getdoc.jsp?containerId=prUS24298013

[17] M. A. H. Masud and X. Huang, “An E-learning System Architecture based on Cloud Computing“, World Academy of Science, Engineering and Technology 62, 2012.

[18] http://blog.runtux.com/2009/08/04/106/

[19] http://www.thecloudcomputing.org/2014/history.html

[20] http://aws.amazon.com/message/65648/

[21] http://readwrite.com/2010/02/01/top-5-cloud-outages-of-the-pas#awesm=~oDZ80KtLjpt4Ss

[22] http://status.foursquare.com/

[23] http://www.huffingtonpost.com/2011/08/10/facebook-goes-down_n_924016.html

[24] http://en.wikipedia.org/wiki/Byzantine_fault_tolerance

[25] http://www.remics.eu/

[26] http://www.cloudmig.org

[27] http://www.canopy-cloud.com/news/atos-canopy-leads-artist-project-migrate-to-the-cloud

[28] http://www.cloudtm.com

[29] http://contrail-project.eu/

[30] http://www.visioncloud.eu/

[31] http://blog.evolveip.net/index.php/2012/05/24/cloud-elasticity-and-cloud-scalability-are-not-the-same-thing-2/

Appendix- List of Studies

[S1] Alysson Bessani, Paolo Romano, Rüdiger Kapitza, Spyridon V. Gogouvitis, Roberto G. Cascella, Dana Petcu, Dimosthenis Kyriazis, “A look to the old-world_sky EU-funded dependability cloud computing research”, SIGOPS Oper. Syst. Rev., Volume 46, Issue 2, July 2012.

[S2] Paul Brebner, “Is your cloud elastic enough performance modelling the elasticity of infrastructure as a service (IaaS) cloud applications”, Proceedings of the 3rd ACM/SPEC International Conference on Performance Engineering, 2012.

[S3] Siani Pearson, “Taking account of privacy when designing cloud computing services”, Proceedings of the 2009 ICSE Workshop on Software Engineering Challenges of Cloud Computing, 2009.

[S4] Abdulrahman A. Almutairi, Muhammad I. Sarfraz, “A Distributed Access Control Architecture for Cloud Computing”, Software IEEE, Volume 29, Issue 2, 2012.

[S5] Mohammed Arif Amin, Kamalrulnizam Bin Abu Bakar, Haider Al-Hashimi, “A review of mobile cloud computing architecture and challenges to enterprise users”, GCC Conference and Exhibition (GCC), 2013 7th IEEE, November 2013.

[S6] V.P.Krishna Anne, Vidya Sagar Ponnam, Gorantla Praveen, “A significant approach for cloud database using shared-disk architecture”, Software Engineering (CONSEG), 2012 CSI Sixth International Conference on, September 2012.

[S7] Manu Annd, “Always On Architecture for High Availability Cloud Applications”, Cloud Computing in Emerging Markets (CCEM), 2012 IEEE International Conference On, October 2012.

[S8] Francisco Javier Nieto, “An architecture for a platform providing things as a service”, Wireless Personal Multimedia Communications (WPMC), 2013 16th International Symposium on, June 2013.

[S9] Mark Gall, Angelika Schneider, Niels Fallenbeck, “An Architecture for Community Clouds Using Concepts of the Intercloud”, Advanced Information Networking and Applications (AINA), 2013 IEEE 27th International Conference on, March 2013.

[S10] Mark Shtern, Bradely Simmons, Michael Smit, Marin Litoiu, “An architecture for overlaying private clouds on public providers”, Network and service management

(CNSM), 2012 8th international conference and 2012 workshop on system virtualization management (SVM), October 2012.

[S11] Bassam S. Farroha, Deborah L. Farroha, "Architecting security into the clouds An enterprise security model", Systems Conference (SysCon), 2012 IEEE International, March 2012.

[S12] E. Casalicchio, L. Silvestri, "Architectures for autonomic service management in cloud-based systems", Computers and Communications (ISCC), 2011 IEEE Symposium on, July 2011.

[S13] Rajkumar Buyya, Rodrigo N. Calherios, Xiaoronog Li, "Autonomic Cloud computing Open challenges and architectural elements", Emerging Applications of Information Technology (EAIT), 2012 Third International Conference on, December 2012.

[S14] Liang-Jie Zhang, Qun Zhou, "CCOA Cloud Computing Open Architecture", Web Services, 2009, ICWS 2009, IEEE International Conference on, July 2009.

[S15] Jing Liu, Liang-Jie Zhang, Bo Hu, Keqing He, "CCRA Cloud Computing Reference Architecture", Services Computing (SCC), 2012 IEEE Ninth International Conference on, June 2012.

[S16] Huimin Zhang, Xiaolong Yang, "Cloud Computing Architecture Based-On SOA", Computational Intelligence and Design (ISCID), 2012 Fifth International Symposium on, October 2012.

[S17] Nawsher Khan, A. Noraziah, Tutut Herawan, Elrasheed Ismail, Zakira Inayat, "Cloud Computing Architecture for Efficient Provision of Services", Network- Based Information System (NBiS), 2012 15th International Conference on, September 2012.

[S18] En NiariSaad, Khalil El Mahdi, Mostapha Zbakh, "Cloud computing architectures based IDS", Security Days (JNS3), 2013 National, November 2012.

[S19] Ratko Mutavdzic,"Cloud computing architectures for national, regional and local government", MIPRO, 2010 Proceedings of the 33rd International Convention, May 2010.

[S20] Huaglory Tianfield, "Cloud computing architectures", Systems, Man, and Cybernetics (SMC), 2011 IEEE International Conference on, October 2011.

[S21] Mingwei Wang, Jingtao Zhou, Shikai Jing, "Cloud manufacturing Needs, concept and architecture", Computer Supported Cooperative Work in Design (CSCWD), 2012 IEEE 16th International Conference on, May 2012.

[S22] Pooyan Jamshidi, Aakash Ahmad, Claus Pahl, “Cloud migration research a systematic review”, IEEE Transactions On Cloud Computing, Volume 1, Issue 2, December 2013.

[S23] Gurudatt Kulkarni, Rani Waghmare, Rajnikant Palwe, Vidya Waykule, Hemant Bankar, Kundlik Koli,“Cloud storage architecture”, Telecommunication Systems, Services, and Applications (TSSA), 2012 7th International Conference on, October 2012.

[S24] Jiehan Zhou, Teemu Leppänen, Erkki Harjula, Mika Ylianttila, Timo Ojala, Chen Yu, Hai Jin, Laurence Tianruo Yang, “CloudThings A common architecture for integrating the Internet of Things with Cloud Computing”, Computer Supported Cooperative Work in Design (CSCWD) 2013 IEEE 17th International Conference on, June 2013.

[S25] Kapil Bkshi, “Considerations for cloud data centers Framework, architecture and adoption”, IEEEAC paper, March 2011.

[S26] Peterson Gunnar, “Don't Trust. And Verify A Security Architecture Stack for the Cloud”, Security & Privacy, IEEE, Volume 8, Issue 5, October 2010.

[S27] Vajihe Lohmosavi, Akbar Farhoodi Nejad, Elham Morad Hosseini, “E-learning ecosystem based on service-oriented cloud computing architecture”, Information and Knowledge Technology (IKT), 2013 5th Conference on, May 2013.

[S28] Longji Tang, Jing Dong, Yajing Zhao, Liang-Jie Zhang, “Enterprise Cloud Service Architecture”, Cloud Computing (CLOUD), 2010 IEEE 3rd International Conference on, July 2010.

[S29] Antonio Celesti, Francesco Tusa, Massimo Villari, Antonio Puliafito, “How to Enhance Cloud Architectures to Enable Cross-Federation”, Cloud Computing (CLOUD), 2010 IEEE 3rd International Conference on, July 2010.

[S30] Yuri Demchenko, Marc X. Makkes, Rudolf Strijkers, Cees de Laat, “Intercloud Architecture for interoperability and integration”, Cloud Computing Technology and Science (CloudCom), 2012 IEEE 4th International Conference on, December 2012.

[S31] Vijay Sarathy, Purnendu Narayan, Rao Mikkilineni, “Next Generation Cloud Computing Architecture Enabling Real-Time Dynamism for Shared Distributed Physical Infrastructure”, Enabling Technologies, Infrastructures for Collaborative Enterprises (WETICE), 2010 19th IEEE International Workshop on, June 2010.

[S32] Madhukara Phatak, Kamalesh. V. N., “On cloud computing deployment architecture”, Advances in ICT for Emerging Regions (ICTer), 2010 International Conference on, October 2010.

[S33] Saira Begum, Muhhamad Khalid Khan, “Potential of cloud computing architecture”, Information and Communication Technologies (ICICT), 2011 International Conference on, July 2011.

[S34] Hongli Luo, Aaron Egbert, Timothy Stahlhut, “QoS architecture for cloud-based media computing”, Software Engineering and Service Science (ICSESS), 2012 IEEE 3rd International Conference on, June 2012.

[S35] Wei-Tek Tsai,Yu Huang, XiaoyingBai, Jerry Gao, “Scalable Architectures for SaaS”, Object/Component/Service-Oriented Real-Time Distributed Computing Workshops (ISORCW), 2012 15th IEEE International Symposium on, April 2012.

[S36] Qiu Xiu-feng, Liu Jian-wei, Zhao Peng-chuan, “Secure cloud computing architecture on mobile internet”, Artificial Intelligence, Management Science and Electronic Commerce (AIMSEC), 2011 2nd International Conference on, August 2011.

[S37] Wei-Tek Tsai, Xin Sun, Janaka Balasooriya, “Service-Oriented Cloud Computing Architecture”, Infromation Technology: New Generations (ITNG), 2010 Seventh International Conference on, April 2010.

[S38] Samah Ahmed Zaki Hassan, “STAR A proposed architecture for cloud computing applications”, Cloud Computing Technologies, Applications and Management (ICCCTAM), 2012 International Conference on, December 2012.

[S39] Paulo Verissimo, Alysson Bessani, Marcelo Pasin, “The TClouds architecture Open and resilient cloud-of-clouds computing”, Dependable Systems and Networks Workshops (DSN-W), 2012 IEEE/IFIP 42nd International Conference on, June 2012.

[S40] Yanuarizki Amanatullah, Charels Lim, Heru Purnomo Ipung, Arkav Juliandri, “Toward cloud computing reference architecture Cloud service management perspective”, ICT for Smart Society (ICISS), 2013 International Conference on, June 2013.

[S41] Nikolaos Loutas, Vassilios Peristeras, Thanassis Bouras, Eleni Kamateri, Dimitrios Zeginis, Konstantinos Tarabanis, “Towards a Reference Architecture for Semantically Interoperable Clouds”, Cloud Computing Technology and Science (CloudCom), 2010 IEEE Second International Conference on, December 2010.

[S42] Roger Clark, "User Requirements for Cloud Computing Architecture", Cluster, Cloud and Grid Computing (CCGrid), 2010 10th IEEE/ACM International Conference on, May 2010.

[S43] Hao Yuan, Xiaoping Liu, Chunhui Guo, "A design of two-tier SaaS architecture based on group-tenant", Computer Science and Network Technology (ICCSNT), 2012 2nd International Conference on, December 2012.

[S44] Gerd Breiter, Vijay K. Naik, "A Framework for Controlling and Managing Hybrid Cloud Service Integration", Cloud Engineering (IC2E), 2013 IEEE International Conference on, March 2013.

[S45] Jonas Repschlaeger, Stefan Wind, Ruediger Zarnekow, Klaus Turowski, "A Reference Guide to Cloud Computing Dimensions Infrastructure as a Service Classification Framework", System Science (HICSS), 2012 45th Hawaii International Conference on, January 2012.

[S46] V. D. Cunsolo, S. Distefano, A. Puliafito, M. Scarpa, "Applying Software Engineering Principles for Designing Cloud@Home", Cluster, Cloud and Grid Computing (CCGrid), 2010 10th IEEE/ACM International Conference on, May 2010.

[S47] Santonu Sarkar, Rajeshwari Ganesan, Manish Srivastava, Sharada Dharmasankar, "Cloud Based Next Generation Service and Key Challenges", Services in Emerging Markets (ICSEM), 2012 Third International Conference on, December 2012.

[S48] Marty Humphrey, Jacob Steele, In Kee Kim, Michael G. Kahn, Jessica Bondy, Michael Ames, "CloudDRN A Lightweight, End-to-End System for Sharing Distributed Research Data in the Cloud", eScience (eScience), 2013 IEEE 9th International Conference on, October 2013.

[S49] Teresa Tung, "Defining a Cloud Reference Model", Cluster, Cloud and Grid Computing (CCGrid), 2011 11th IEEE/ACM International Symposium on, May 2011.

[S50] Yong Zhao, Yanzhe Zhang, Wenhong Tian, Ruini Xue, Cui Lin, "Designing and Deploying a Scientific Computing Cloud Platform", Grid Computing (GRID), 2012 ACM/IEEE 13th International Conference on, September 2012.

[S51] Bogdan Solomon, Dan Ionescu, Marin Litoiu, Gabriel Iszlai, "Designing autonomic management systems for cloud computing", Computational Cybernetics and Technical Informatics (ICCC-CONTI), 2010 International Joint Conference on, May 2010.

[S52] Antonio Celesti, Nicola Peditto, Fabio Verboso, Massimo Villari, Antonio Puliafito, "DRACO PaaS A Distributed Resilient Adaptable Cloud Oriented Platform", Parallel and Distributed Processing Symposium Workshops & PhD Forum (IPDPSW), 2013 IEEE 27th International, May 2013.

[S53] Giuseppe Carella, Thomas Magedanz, Konrad Campowsky, Florian Schreiner, "Elasticity as a service for federated cloud testbeds", Communications Workshops (ICC), 2013 IEEE International Conference on, June 2013.

[S54] JhanYuler De la Pava Torres, Claudia Jimenenez-Guarin, "MagosCloud Secure A secure, highly scalable platform for services in an opportunistic environment", High Performance Computing and Simulation (HPCS), 2012 International Conference on, July 2012.

[S55] Ying Chen, QingniShen, Pengfei Sun, Yangwei Li, Zhong Chen, Sihan Qing, "Reliable Migration Module in Trusted Cloud Based on Security Level - Design and Implementation", Parallel and Distributed Processing Symposium Workshops & PhD Forum (IPDPSW), 2012 IEEE 26th International, May 2012.

[S56] Linping Su, Lin Li, Lu Zhang, Xiaoqian Nie, "Research and Design of Electric Power Private Cloud Data Storage Model", Computational and Information Sciences (ICCIS), 2012 Fourth International Conference on, August 2012.

[S57] Lawrence Chung, Tom Hill, Owolabi Legunsen, Zhenzhou Sun, Adip Dsouza, Sam Supakkul, "A goal-oriented simulation approach for obtaining good private cloud-based system architectures", Journal of Systems and Software, Volume 86, Issue 9, October 2012.

[S58] Rui Han, Moustafa M. Ghanem, Li Guo, YikeGuo, Michelle Osmond, "Enabling cost-aware and adaptive elasticity of multi-tier cloud applications", Future Generation Computer Systems, Volume 32, Issue 0, May 2012.

[S59] Bahman Javadi, Jemal Abawajy, Rajkumar Buyya, "Failure-aware resource provisioning for hybrid Cloud infrastructure", Journal of Parallel and Distributed Computing, Volume 72, Issue 10, June 2012.

[S60] Georg Lackermair, "Hybrid cloud architectures for the online commerce", Procedia Computer Science, Volume 3, Issue 0, 2010.

[S61] Azubuike Ezenwoke, Nicholas Omoregbe, Charles Korede Ayo, Misra Sanjay, "NIGEDU CLOUD Model of a National e-Education Cloud for Developing Countries", IERI Procedia, Volume 4, Issue 0, 2013.

[S62] Dan Petcu, Georgiana Macariu, Silviu Panica, Ciprian Craciun, "Portable Cloud applications—From theory to practice", Future Generation Computer Systems, Volume 29, Issue 6, January 2012.

[S63] Knud Brandis, Srdan Dzombeta, Knut Haufe, "Towards a framework for governance architecture management in cloud environments A semantic perspective", Future Generation Computer Systems, Volume 32, Issue 0, September 2013.

[S64] Lizhe Wang, Marcel Kunze, Jie Tao, Gregor von Laszewski, "Towards building a cloud for scientific applications", Advances in Engineering Software, Volume 42, Issue 9, March 2011.

[S65] Christos Kalloniatis, Haralambos Mouratidis, Manousakis Vassilis, Shareeful Islam, Stefanos Gritzalis, Evengelia Kavakli, "Towards the design of secure and privacy-oriented information systems in the cloud Identifying the major concepts", Computer Standards & Interfaces, Volume 36, Issue 4, December 2013.

[S66] Won Kim, "Cloud architecture a preliminary look", 9th International Conference on Advances in Mobile Computing and Multimedia, MoMM, 2011.

[S67] Gianmario Motta, Nicola Sfondrini, Daniele Sacco, "Cloud computing An architectural and technological overview", 2012 International Joint Conference on Service Sciences, Service Innovation in Emerging Economy: Cross-Disciplinary and Cross-Cultural Perspective, IJCSS 2012, May 2012.

[S68] Mahesh H. Dodani, "The practice of architecting cloud solutions", Journal of Object Technology, Volume 9, Issue 1, February 2010.

[S69] Rick McGeer, "Transcloud Design considerations for a high-performance cloud architecture across multiple administrative domains", 1st International Conference on Cloud Computing and Services Science, CLOSER 2011, 2011.

[S70] Michael Armbrust, Armando Fox, Rean Griffith, Anthony D. Joseph, Randy H. Katz, Andrew Konwinski, Gunho Lee, David A. Patterson, Ariel Rabkin, Ion Stoica, Matei Zaharia, "Above the Clouds A Berkeley View of Cloud Computing", Technical Report, February 2009.

[S71] Christoph Fehling, Frank Leymann, Ralph Retter, David Schumm, Walter Schupeck, "An architectural pattern language for cloud-based applications", October 2011.

[S72] Fang Liu, Jin Tong, Jian Mao, Robert Bohn, John Messina, Lee Badger, Dawn Leaf, "NIST Cloud computing reference Architecture", NIST Special Publication 500-292, July 2011.