ALMA MATER STUDIORUM · UNIVERSTÀ DI BOLOGNA

FACOLTÀ DI SCIENZE MATEMATICHE, FISICHE E NATURALI

Corso di Laurea Specialistica in Informatica

Cloud Computing: Accountability

Relatore:

Chiar.mo Prof.

Fabio Panzieri

Presentata da:

BRAHIM Kamel

III Sessione

Anno Accademico 2010/2011

Table of Contents

Abstract

For many companies cloud computing offers a new era of IT opportunity as it provides a powerful and flexible computing environment without any upfront commitment. However like any other new outsourcing IT environment the problem of trust for customer and provider, acts as a barrier to a wide adoption of the cloud paradigm. From a customer perspective, placing his computation and data on machines that he cannot directly control may be risky. From a provider perspective, he can't be liable for accepting to run a service whose details he does not know. If something goes wrong for example data leaks to a competitor or the computation return incorrect results no one of them is in good position to hold the other responsible.

The purpose of this thesis is to examine accountability in cloud computing environment as an urgent needed mechanism that can address the problem of trust in cloud computing. In the case when a dispute arises, accountability provides the means through which the cause of this dispute can be clearly and unambiguously established.

This thesis starts by focusing on service level agreement first as it define the quality of service and second as it represents the reference if any dispute arises before assigning responsibility. This thesis examines accountability and describes several challenges that are not yet addressed by current accountability techniques. Trust is a barrier that holding back the adoption of cloud as an alternative. Preventive controls are not enough to address challenges such as security and privacy. This examination of the mechanism will be achieved by a literature review of previous solution in the distributed system and proposed works in the cloud computing.

Sommario(Italiano)

Per molte aziende cloud computing offre una nuova opportunità di IT in quanto fornisce un ambiente di elaborazione potente e flessibile, senza alcun impegno in anticipo. Per anni, l'outsourcing nell' IT è crescuito e il cloud è simile al classico IT outsourcing, in cui il cliente trasferisce il controllo di una parte del suo sistema d' informazioni a un fornitore di servizi. Quando un'azienda o organizzazione si affida a servizi forniti da un *cloud provider* per l'implementazione dei propri processi di business, spesso vengono richieste garanzie contrattuali sulla qualità del servizio, come allo stesso modo i fornitori del servizio richiedono garanzie affinché i clienti non abusino del servizio. Queste qualità e i vincoli di utilizzo sono spesso definiti in accordi bilaterali chiamati Service Level Agreement (SLA), che specificano la qualità del servizio richiesta e le penalità associate alle violazioni. Queste penalità sono tradotte in un rimborso per il costo del servizio e possono essere viste come un'assicurazione contro la fornitura di un servizio che non funziona come previsto. Attualmente il cloud offre due tipi di SLA :

- Predefinito o non-negoziabile: i termini del servizio vengono prescritti completamente dal cloud provider con rischi in materia di privacy e di sicurezza. Inoltre, il provider può apportare modifiche unilaterali ai termini contrattuali.
- Negoziabile: paragonabili ai tradizionali contratti di outsourcing per il servizio IT. Possono includere i requisiti dell'organizzazione sulla sicurezza, privacy policy, procedure e controlli tecnici. Questo tipo può includere rischi quando i termini di servizio non soddisfano le esigenze dell'azienda o dell'organizzazione se il SLA è stato firmato senza ottenere un'adeguata consulenza tecnica e legale.

Le offerte cloud computing soffrono ancora la mancanza di supporto per il Business Service Management (BSM), in particolare per la gestione del SLA. Questa mancanza di sostegno deriva dal fatto che i cloud service provider trovano poco pratico creare offerte

personalizzate con conseguenze di tradurre i requisiti business in manifestazioni tecniche e di ottimizzare le implementazioni interne per ogni SLA. Ultimamente molti progetti tipo RESERVOIR o SLA@SOI hanno cercato di indirizzare questo problema attraverso il modello del federated cloud computing. Ma anche se SLA è un legame giuridico che deve fornire qualche garanzia sui termini già predefiniti, non è sufficiente per superare l'ostacolo di diffidenza.

Come qualsiasi altro nuovo ambiente di outsourcing IT il problema di fiducia per il cliente e il fornitore, agisce come una barriera ad una ampia adozione del paradigma cloud. Quando un'azienda o organizzazione vuole migrare i sui processi business verso un ambiente cloud sono pronte a perdere il controllo dei dati ma mantenere l'accountability anche se la responsabilità potrebbe essere assegnata a più parti. Per l'IT manager, l'economia o il fattore di costo offerto dalla migrazione verso il cloud è importante. Allo stesso tempo sono di una certa importanza le relazioni con i clienti, l'immagine pubblica, la flessibilità, la *business continuity* e la conformità. Inoltre, utilizzare cloud è un po' rischioso a causa di una perdita di controllo e mancanza di regole. I clienti hanno bisogno di garanzie da parte del *cloud service provider*. Tali garanzie devono impedire che i clienti subiscano qualsiasi perdita di dati o un attacco durante il calcolo in grado di incidere direttamente o indirettamente sulla loro business continuity. Anche se i rischi possono essere mitigati attraverso controlli preventivi per la privacy e la sicurezza (ad esempio SSL, la crittografia, controllo di accesso basato sulla definizione di profili ID, ecc.) questi non sono sufficienti poiché i clienti hanno bisogno di misure supplementari che possano assicurare responsabilità e garanzia. Dal punto di vista del provider, esso non può essere ritenuto responsabile per aver accettato di eseguire un servizio senza conoscerne i dettagli. Se si verifica un problema ad esempio, che i dati finiscono nelle mani di un concorrente o la computazione restituisce risultati non corretti nessuno di loro è in buona posizione per tenere l'altro responsabile. Come l'adozione del cloud computing come soluzione alternativa ai paradigmi tradizionali presenta molti rischi, possiamo capire l'importanza dell' accountability per l'esecuzione di un controllo dettagliato e che può istituire un certo livello di fiducia tra *cloud provider* e i clienti. Uno tra i maggiori

ostacoli a l'adozione del paradigma cloud è il problema della sicurezza. i clienti prima di affidare i loro processi business ad un *cloud provider* hanno delle grandi aspettative. Tuttavia, questo non è il caso, secondo uno studio pubblicato da Ponemon un istituto indipendente di ricerca nella sicurezza e nella privacy, cloud provider sono più focalizzati sulla fornitura di vantaggi come il basso costo e la velocità di implementazione associati al cloud computing più che la sicurezza. In una ricerca condotta da CA technologies mostra che (79%) dei cloud provider offrono solo il (10 %) per la sicurezza le attività relative di controllo. Accountability può giocare un ruolo importante per migliorare la sicurezza e ridurre i rischi.

Un altra questione riguardo al cloud è il problema della privacy anche se i controlli preventivi possono fornire qualche grado di sicurezza ma non sono sufficiente a garantire privacy. Ad esempio DropBox nel 2011 dopo una modifica nel codice ha disattivato i controlli di sicurezza e ha messo a rischio 25 milione di utenti. Accountability per la privacy può essere ottenuta tramite l'implementazione di meccanismi che possono garantire che le politiche siano rispettati dalle parti che usano o condividono i dati. In più accountability può rispondere a delle domande del tipo :

- Il cloud provider ha applicato le misure corrette per proteggere la privacy ?
- Il cloud provider ha rispettato i principi sulla privacy e i requisiti descritti dalla legislazione, ad esempio la direttiva UE 95/46/CE?

Tuttavia, privacy e accountability possono avere dei conflitti in quanto quest'ultima produce un log dettagliato che può essere controllato da un terzo.

I problemi relativi alla sicurezza e alla privacy possono incidere direttamente o indirettamente sul problema del trust. Mentre molti ostacoli per il trust che derivano dai problemi di privacy e di sicurezza possono essere affrontate attraverso controlli preventivi non sono sufficienti, soprattutto quando ad esempio un attacco utilizzando *impersonation* o il *social engineering* è stata effettuato per offuscare il rilevamento, sarà difficile se non impossibile assegnare responsabilità. Accountability è considerata come una proprietà che può garantire un *trustworthy computing system*. Recentemente un riconoscimento per la

necessità di accountability in ambiente cloud è cresciuto costantemente. Questo requisito è stato ulteriormente accelerato dalla alta sfiducia e i limiti dei controlli preventivi.

Spesso cloud computing viene considerato come un sistema distribuito su larga scala. L'adozione del on-demand computing service (ad esempio il grid computing o il utility computing) è cresciuto molto negli ultimi anni. La necessità dell'accountability è diventato molto evidente in quanto i servizi sono molti complessi e interdipendenti. Tra le soluzioni discusse in questa tesi ci sono le seguenti:

- Il sistema PeerReview, per esempio, registra informazioni aggiuntive affinché l'esecuzione di ciascun nodo possa essere riprodotta in modo deterministico. La riproduzione consente di rilevare eventuali guasti, attacco e disordini che comportano una modifica del nodo software o della sua configurazione. Inoltre, si può mostrare che un tale guasto in un sistema distribuito può essere rilevato, purché i log di ciascun nodo vengono controllati ed eventualmente riprodotti da un nodo attendibile. Questa soluzione non può essere migrata verso cloud. Ad esempio PeerReview dovrebbe essere integrato nella applicazione che richiede la modifica del codice sorgente. In più, aggiunge degli *overheads* come rallentamento nel tempo di risposta e CPU *overhead*.
- Accountable Virtual Machine (AVM) è una generalizzazione che funziona per arbitrare immagini binarie senza alcuna modificha. In questa soluzione l'immagine del software di ciascun nodo di un sistema distribuito viene eseguita all'interno di una macchina virtuale. Questa soluzione anche se viene addottata per l'ambiente cloud può includere rischi sulla privacy. In quanto AVM registra molta informazione che può essere visibile durante un operazione di controllo.

Ulteriori sforzi per implementare accountability nel ambiente cloud includano il framework TrustCloud proposta da HP e *Accountability service for the cloud* proposta in [64]. Questa tesi esamina queste soluzioni e ne discute criticamente i meriti ed i limiti.

Chapter **1**

Introduction

Recently an emerging technology called cloud computing has reshaped the way that IT could be designed and delivered. There's no standard definition for cloud computing until now. According to Zhang et al. [1] the most appropriate definition is provided by The National Institute of Standards and Technology(NIST), as it covers all the essential aspects of cloud computing : ***"****NIST definition of cloud computing Cloud computing is 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"* [2]*(See appendix 1 for more cloud computing definitions).*

Cloud computing has recently emerged as a new model for hosting and delivering services over the Internet in a pay-per-use fashion. These services are divided in three categories Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS) and Software-as-a-Service (SaaS).

The term of cloud computing is just now gaining attention on the marketplace due to enormous capacity with comparable low cost. Berkeley Report [3] released in Feb 2009 notes - "*Cloud computing, the long-held dream of computing as a utility, has the potential to transform a large part of the IT industry, making software even more attractive as a service and shaping the way IT hardware is designed and purchased*."

The possibility to dynamically lease and release resources without upfront commitment makes cloud more promising. The users who need to perform data intensive applications can find in cloud computing an alternative to traditional computing paradigms (like grid computing, utility computing, virtualization, etc…).

However these applications may exhibit strict Quality of Service (QoS) requirements such as Response time, throughput, availability, reliability and security that can be specified in so called SLA Service Level Agreements (SLAs) [4].

1.1 Motivation

Significant innovations in virtualization and distributed computing, as well as improved access to high-speed Internet and a weak economy, have accelerated interest in cloud computing. The market and research and analysis firm IDC suggests that the market for cloud computing services was $16 billion in 2009 and will rise to $55.5 billion/year by 2014 with an annual growth rate of 27.4 % compared with 5% of annual growth for traditional IT products [5].

For years, the IT industry has grown in outsourcing and cloud computing is similar to classic IT outsourcing, where client transfers the control of parts of its information system to a service provider. The cloud provider assumes responsibility for the client's Information system and operates it in accordance to the contractual terms which are defined in SLAs.

As cloud computing is still at its infant stage and like any other area in which there is a rapid innovation, both providers and customers face some challenges to address.

From a customer perspective, before adopting cloud there are many questions that need to be answered:

What is cloud computing? "*Is it a general term for anything that involves delivering hosted services over Internet* " [6]?

What does cloud computing have to offer more than traditional technologies? Is it only the economics of scale or a new computing paradigm which involves traditional technologies paradigms (like grid computing, utility computing, virtualization, etc.)?

For IT manager economics or cost factor is important but at the same time customer relationships, public image, flexibility, business continuity and compliance are of some importance.

Furthermore, using cloud is somewhat risky due to a loss of control and lack of regulations. Customers need guarantees from providers on service delivery. Such guarantees must prevent customers' data from any leakage or attack during computation that can affect directly or indirectly their business continuity. While risks can be mitigated via preventive controls for privacy and security (e.g. SSL, encryption, access control based on ID profiling, etc.) they are not enough as customers need additional measures that can assign responsibility and guarantee liability.

For a cloud service provider it is critical to correctly monitor and control thousands of devices like servers, switches and routers. This technical limitations, like any other computing services can affect the cloud QoS and consequently make services suffer slowdowns and failures. For instance, a partial failure at the Amazon web services has caused the outage that derailed many web sites like Reddit[1] and Quora[2] [7].

Some potential problems that may occur due to a loss of control are listed below [8]:

- Misconfiguration in cloud machines can cause unexpected results.
- SLAs violation due to an inadequate allocation of resources.
- An attacker may be able to exploit a bug in the customer's software allowing him to steal valuable data or to launch a DDOS attack.
- The data may be unavailable or inaccessible at an inconvenient time

In order to reach the goal of the fifth utility (along with water, electricity, gas, and telephone) and as an alternative to traditional paradigms, cloud provider and customer must

[1] http://www.reddit.com/

[2] http://www.quora.com/

be conscious that neither of them is in good position to address solely these problems due to the natural of outsourcing computing. In addition, when something goes wrong, the customer and the provider, even detecting the presence of a problem, face the potentially difficult task of deciding which of them is responsible for it: on the one hand, the provider does not know what to look for, since he does not know what the computation is supposed to do; on the other hand, customer can only access the cloud machines remotely, so he has only a very limited information.

Moreover, as cloud computing is a new paradigm, the design of a fault-detection system that will meet the reliability requirements can be one of the most difficult tasks due to the nature of computing in outsourcing environments which include asynchronous operations, unreliable communication and is highly scalable.

As cloud computing exhibits all the drawbacks above, we can understand the Importance of Accountability for performing due diligence and establishing a certain level of confidence between cloud providers and customers.

1.2 Goal of this thesis

The goal of this thesis is to analyze the efficacy of accountability in cloud computing as a mean which can play an important role in addressing the current mistrust. In other words, the goal is to analyze why accountability should be taken into consideration as the most important “missing piece” in the cloud computing environment which can help towards the fully adoption of cloud as an economies of scale.

In order to do that the first part of this thesis present the Service Level Agreement (SLA) which is legal means that specifies a certain level of Quality of Service (QoS). However, even if this legal binding which should provide some guarantee to the cloud customer, it is not sufficient to overcome the mistrust barrier.

With this in mind the second part of this thesis defines accountability and describes several challenges that are not yet met by current accountability techniques. In addition, this thesis outlines the technical requirements for an accountable cloud by analyzing different solutions

proposed for the distributed system given that cloud computing is considered a large-distributed systems which introduce several complexities and constraints. Finally this thesis discusses some proposed solutions with the purpose to make cloud accountable.

1.3 Structure of this thesis

Chapter 1: Introduction

This chapter introduces the reader to the topic and gives a brief overview of the entire thesis.

Chapter 2: State-of-the-art

This chapter gives theoretical background information about cloud computing

Chapter 3: Quality of Service

Since everything in the cloud is delivered a service this chapter introduces the concept of QoS and SLA as a contract which determines the service level requirements and sets out responsibilities and priorities. Finally this chapter concludes by highlighting the needs for federated cloud computing as an urgent need in order to overcome many QoS issues in the cloud computing environment.

Chapter 4: Accountability

This chapter defines accountability and describes several challenges that are not yet met by current accountability techniques. In addition, this chapter outlines the technical requirements for an accountable cloud by analyzing different solutions proposed for the distributed system. Finally this chapter discusses some proposed solutions with the purpose to make cloud accountable.

Chapter 5: Conclusion

This chapter concludes the thesis by summarizing the results and giving an outlook.

Chapter **2**

State-of-the-Art

This chapter introduces and discusses the state-of-the-art concepts that are relevant to this thesis. In addition, I will discuss some related technologies to cloud computing as many people still confuse it with others traditional paradigms.

2.1 Cloud Computing

The main idea of cloud computing is not a new one, it was envisioned by John McCarthy since 1960 [9]. For instance, in 1969, Leonard Kleinrock one of the chief scientists of the original Advanced Research Agency Network (ARPANET) who created the first packet switched network, was quoted to have said the following : *"As of now, computer networks are still in their infancy, but as they grow up and become sophisticated, we will probably see the spread of 'computer utilities', which, like present electric and telephone utilities, will service individual homes and offices across the country"* [10].

Cloud computing has recently become a buzzword in both academia and business world. Figure 2.1 shows, how often the term Cloud Computing was entered in Google search relative to the total search-volume across various regions of the world, and in various languages.

Even though the term is widely used, the cloud computing is still suffering from the lack of standard definition. As an example, a work called "*A break in the clouds: towards a cloud definition "* [3] compared over 20 definitions from a variety of sources to confirm a standard definition. All major research and consultancy firms have also rushed to publish their own definitions of cloud computing. For example, Gartner, the leading IT research and advisory firm, has published the following definition:

"[Cloud computing is] a style of computing in which scalable and elastic IT-enabled capabilities are delivered as a service to external customers using Internet technologies."

Given the fact that there is still an ongoing debate regarding cloud definition, it's important to introduce in the followings sections common definitions that can be relevant for this thesis.

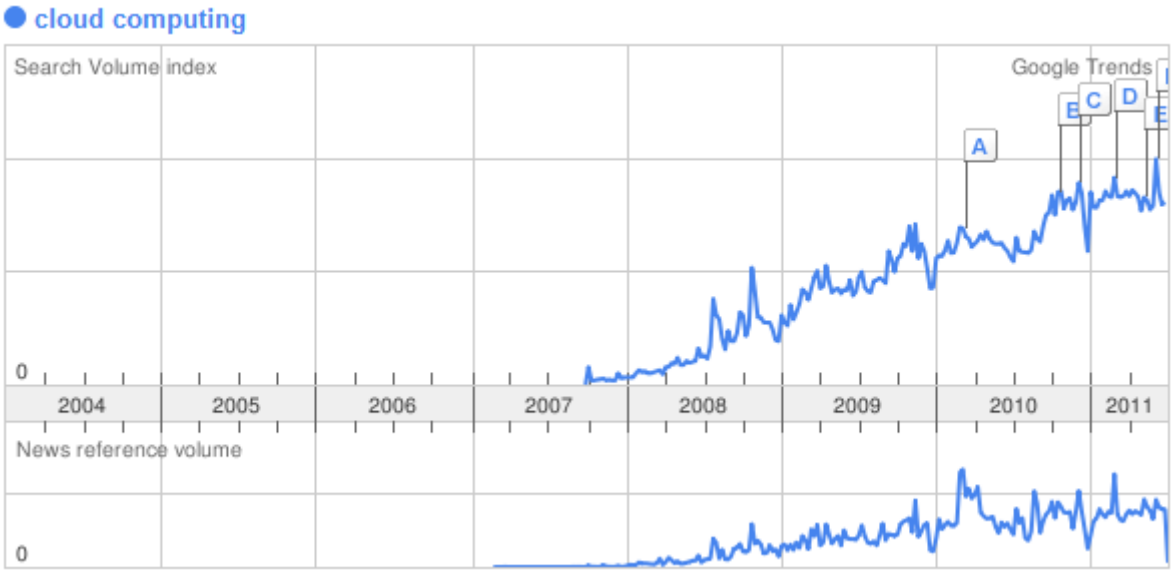

Figure 2.1: *Cloud Computing on Google trends* [11]

2.1.1 Layers of cloud computing

A cloud computing environment is composed of several layers, all of which can be accessed by users connected to it. By structuring the cloud computing architecture into layers, it is easier to define the roles and skills needed within the overall structure to see where your business fits into the model.

According to Zhang et al. [1] cloud computing architecture can be divided into 4 layers:

- *The Hardware Layer:* represents the physical hardware that provides actual resources that make up the cloud**.** In practice the hardware layer is typically implemented in data centers with thousands of servers that are organized in racks

and interconnected through switches, routers or others fabrics. Some issues in this layer include hardware configuration, fault tolerance, traffic management, and power cooling resource management.

- *The infrastructure layer:* also known as the virtualization layer which is an essential component of cloud computing. This layer provides much of the scalability and flexibility of the cloud computing model by inheriting ability of virtual machines to be created and deleted at will. Some virtualization technologies used are KVM, Xen and VMware.
- *The platform layer:* is based upon the infrastructure layer and offers operating systems and application frameworks. The purpose of this layer is to hide the complexity (like scalability and locality) of the infrastructure layer by providing custom programming APIs. Well Known examples are Google's App Engine, Microsoft's Azure, and Salesforce.com's Force.com.
- *The application layer:* Consists of the actual cloud applications which are different from traditional applications, by leveraging the automatic-scaling feature to achieve better performance, availability and lower operating cost.

Figure 2.2 illustrates the 4 layers of cloud computing environment.

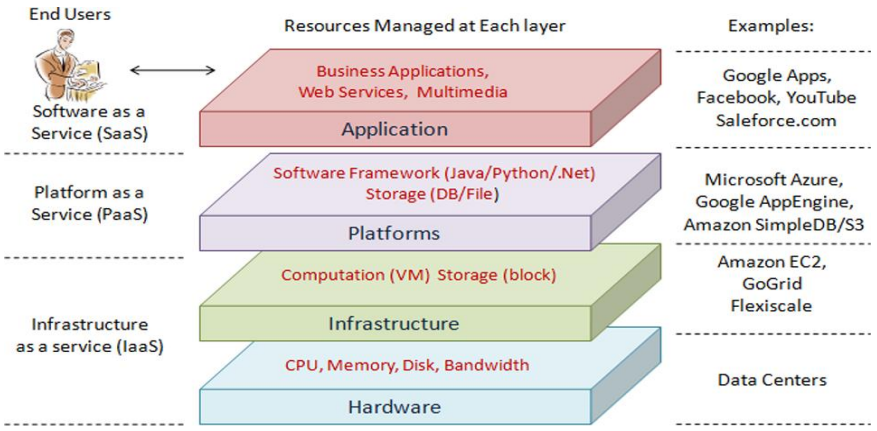

Figure 2.2*: Cloud computing architecture* [1]

2.1.2 Characteristics

As in all IT services, features and characteristics are important to take in consideration from all actors[3]. In fact, every institution that published a definition for cloud computing also published characteristics.

According to Gartner [12] and many others institutions like NIST [2] the most important five essential characteristics are:

- *Service-based:* implementation details of a service are hidden from the customer by a service interface that is well-defined. The service can be considered ready-to-use or off-the-shelf because it is designed to serve the needs of a set of customers [12]. The provision or un-provision of the service can be done automatically without human interaction. The description of the service and how to use the service is defined in the service levels such as availability, response time, performance versus price, etc. These are defined in a non-technical manner and describe IT outcomes rather than the technology itself and its capabilities.
- *Shared Pool of Resources:* In order to build economies of scale, the provider's computing resources are pooled to serve multiple consumers. This is done by assigning physical and virtual resources dynamically using a multi-tenant model. Thus, resources are shared by multiple customers in order to support a cost-effective pricing structure.
- *Scalable and Elastic:* This characteristic deals with the provisioning and the release of resources allowing cloud computing to offer the ability to quickly scale up and scale down and giving the illusion of infinite resources. The ability to perform this kind of scaling is located in the underlying infrastructure and software platform [12] which can allocate or deallocate resources for the particular service from a resource pool. Cloud computing can deal with two types of scalability vertical and horizontal.

[3] Actors : consumer, provider, auditor, broker and carrier

Vertical scalability is the ability is to increase the capacity of existing hardware or software by adding resources [13] which allow an application to continue to perform consistently as load increases. Horizontal scalability means that the ability to add more servers in order to meet growing demand typically through the traditional clustering load balanced model [14]. Elasticity is associated with the economic model; this of course has an impact on the service cost [12].

- *Metered by Use:* by monitoring the usage of the services, cloud provider can control and optimize resource usage [2]. Tracking of service usage is also important for business models of cloud services since it is the foundation of providing pay-per-use plans, subscriptions or fixed plans. These plans are based on usage by a customer, which might be in terms of hours, data transfers and/or other use-based attributes [12]. This characteristic makes this service very well defined and its cost is predictable.
- *Broad Network Access:* Cloud computing provides the users with a variety of services on the network that can be accessed through standard mechanism. These characteristics promote the use by heterogeneous thin or thick client platforms (e.g. mobile phones, laptops, and PDAs).

2.1.2 Service Models

A service model is a scenario in which a Cloud Provider offers a certain amount of capability to service users. The service models describe the degree of control the Cloud Service Provider (CSP) offers, and the degree of freedom a customer has. Figure 2.3 gives an overview on service models and their components.

Cloud providers typically focus on provisioning just one service model. However, there is no restriction in offering multiple types at the same time. For instance Google offers App Engine which is PaaS in combination with Google Docs which a SaaS [15].

Cloud computing services can be grouped into three typical service models:

- *Infrastructure as a Service:* (IaaS) the cloud provider delivers computing infrastructure, typically through a platform virtualization environment. This is a low level of abstraction that allows cloud customers to access the underlying infrastructure through the use of virtual machines. The infrastructure cloud provider manages a large pool of computing resources, such as storing and processing capacity. Customers can scale dynamically the configuration to meet changing needs, and are charged based on their usage. Well-known examples for this IaaS are Amazon Web services EC2 and S3.

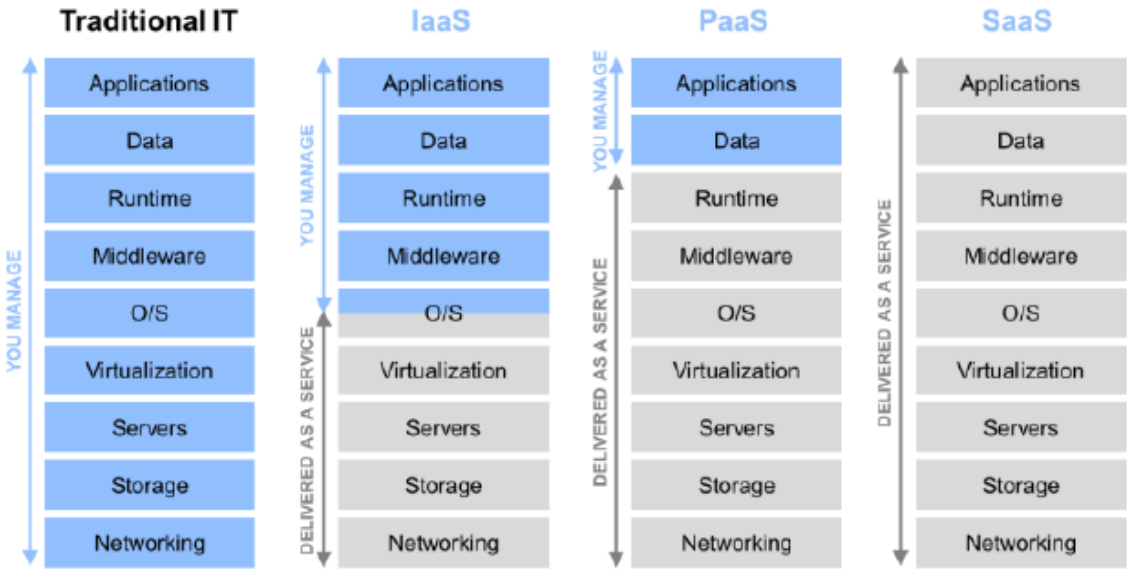

Figure 2.3: *Service models* [16]

- *Platform as a Service (PaaS):* This service model adds more value to the capabilities of IaaS by adding one additional abstraction layer. Developers can focus on developing their applications and not worry about the underlying infrastructure. The sizing of the hardware resources of the PaaS platform is done in an elastic scalable way in order to generally provide as much computing resources as the

customer demands. A well-known example is Google's App Engine [17], Microsoft's Windows Azure [18] and Force.com [19].

- *Software as a Service (SaaS):* this service provides on demand applications over the Internet [1]. These applications are commonly web-based and users access them by making use of a standard web-browser. SaaS providers usually offer these applications based on a pay-per-use fee. Customers, on the other hand, do not have to be concerned with the installation, operation, update etc. of the application or the infrastructure being used to host the application.

According to these aspects, Cloud Computing service models can be structured as displayed in figure 2.4.

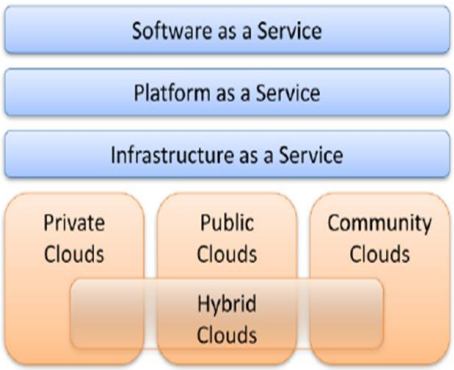

Figure 2.4: *Cloud computing deployment and service models*

Examples of SaaS providers include Salesforce.com [20], Rackspace Dedicated Server, Managed Hosting, and SAP Business ByDesign [21].

2.1.4 Deployment Models

Similar to the service models discussed in the previous section, there is common agreement that cloud computing can be deployed in different ways [22]. Mainly, deployment models are about who owns, manages and is responsible for the services. The following section discusses different kinds of deployment models.

- *Private Cloud:* this deployment model provides an exclusive cloud for every organization. The cloud may be operated by the organization itself or a third party and may exist on-premise or off-premise [22]. If the private cloud is operated by a

third party off-premise, it is similar to classic outsourcing. However, cost advantages of private clouds are mainly driven by virtualization technologies. This model is targeted at customers with high data and security requirements, higher SLA requirements and/or specific legal requirements.

- *Public Cloud:* using this deployment model a cloud infrastructure is made available to the general public and is owned by a provider selling cloud services such as Microsoft, Google or Amazon [22].This model is suitable for customers with a need to get benefit of economies of scale and highly standardized production environment. Therefore, customers must accept low data security and data privacy levels. Examples of public clouds include Amazon Elastic Compute Cloud (EC2), Google AppEngine and Windows Azure Services Platform.
- *Community Cloud:* This model allows sharing the infrastructure within a distinct group of organizations. This cloud is usually set up to the specific needs and requirements of this group that has shared concerns (e.g., mission, security requirements, policy, and compliance considerations) [2]. It could be managed by the organizations themselves or by a third party. For example the Open Cirrus[4] cloud computing Research Testbed that aims to support research in cloud computing could be regarded as community cloud [22].
- *Hybrid Cloud:* This model is a mixture of the above mentioned three deployment models. A hybrid cloud is independently managed but its data and applications can have the ability to burst outside of one model into another. An example of this is a cloud deployed as a private cloud that uses resources from a public cloud in cases of very high demand [2]. This model is suitable for customers looking for a scalable, standardized and cost efficient solution that also cover dedicated security requirements. The combination of both approaches leads to complex implementation architectures.

[4] www.opencirrus.org

2.1.5 Economic Aspects

The reason why cloud computing is so widely discussed at the moment is the fact that it bears a lot of economic benefits for companies. This section describes and discusses various benefits that make cloud economies of scale [22].

- *Cost Reduction:* the desire to reduce costs is omnipresent in companies today. Thus it is one of the primary drivers for companies to rely on cloud infrastructures. Cloud infrastructures enable companies to pay only for the IT they actually need and when they need it, instead of continuously supporting an infrastructure that is sufficient to handle their worst case computing needs. Cloud computing promises to reduce cost for procurement and infrastructure maintenance for customer companies. In addition, data center offer many advantages such as innovation, security and human capital amortization.
- *Pay per Use:* charging customers only for resources they actually consume is a very important aspect. This aspect is one of the primary economic drivers of cloud computing since it enables companies to decrease their upfront investments and shift it to operational expenditures.
- *Improved Time to Market:* this aspect is especially interesting for small and medium enterprises (SMEs) since they usually do not have the required cash to quickly acquire and setup an IT-infrastructure. Cloud computing, however, provides SMEs the ability to deploy their product on an infrastructure they can hire very fast and use it according to their needs. In this way, cloud computing can help SMEs bridge the gap of financial liquidity, enabling them to be more competitive with large enterprises. However, even larger enterprises can benefit from using this new paradigm in order to accelerate their time to market since even they need to be able to publish new capabilities with little overhead to remain competitive.

- *Converting CAPEX[5] into OPEX[6]:* For a company acquiring and setting up a local IT-Infrastructure CAPital EXpenditure (CAPEX) is required. Using a Cloud infrastructure transforms this CAPEX into OPerational EXpenditure (OPEX). OPEX is an on-going cost for running a product, business, or system. Transforming CAPEX to OPEX delays the upfront outlay of funds. From a business perspective there is a discussion regarding whether this transformation is financially beneficial or not [23].
- *Green ICT:* Green ICT or Green IT has been a hype topic for quite some time and its goal is to reduce the energy consumption of ICT systems as well as the carbon footprint. Of course for most companies the most appealing aspect of Green ICT is to reduce cost by reducing energy consumption. However, since cloud computing deals a lot with trying to use resources in the most efficiently way (e.g. through virtualization), cloud computing also provides the benefit of Green ICT. This can also be seen more and more with the operating of computing services from geographical regions that have more abundant non-carbon source energy available such as geothermal or wind.

2.2 Cloud Computing vs. Related Technologies

Despite of the high expectations, cloud computing has also many critics; the exaggerated hype, lack of clear definition, and general novelty of the concept makes it a very controversial topic. For example, in his famous address Oracle's chief executive officer Larry Ellison stated that "*the interesting thing about cloud computing is that we've redefined cloud computing to include everything that we [the IT industry] already do*" [24].

Gartner says," *firms should clearly separate the consideration of cloud computing and cloud computing services from the use of cloud computing-related concepts and technologies for the creation of internal systems*" [25]

[5] CAPital EXpenditures (CAPEX or capex) are expenditures creating future benefits [116]

[6] OPerational EXpenditure or OPEX is an ongoing cost for running a product, business, or system [117]

The following sections of this chapter show how different is cloud computing from related technologies and how cloud computing leverages these existing technologies to meet the technological and economic benefits requirements.

2.2.1 Virtualization

Cloud computing and virtualization are often used interchangeably, but they are very different concepts. A TechTarget's definition of virtualization is **"***the creation of a virtual (rather than actual) version of something, such as an operating system, a server, a storage device or network resources*." [26]

Virtualization has provided IT with incredible flexibility and an ability that would allow dynamic scaling, automatic reaction to performance and correctness problems.

Virtualization enables a more efficient utilization of existing computing resources by increasing the utilization of machines and reduces the number of physical devices.

Cloud providers are now looking for ways to improve the average utilization of their servers, reduce maintenance cost and also retain the Quality of Service (QoS). Server consolidation can solve issues related to lower utilization of physical server, security, quality of service, fault tolerance and application incompatibility issues. Actually the only technology available that makes a logical server consolidation *(see appendix 2)* possible without compromising on any of the desired features is the virtualization.

Virtualization is a key enabling for cloud computing environments, as it provides the capability of pooling computing resources from clusters of servers and dynamically assigning or reassigning virtual resources to applications on-demand [1].

Cloud computing is inclusive of virtualization and a way to implement it. However, cloud can be implemented without virtualization as well such as NoHype [27] architecture which aims at creating cloud without hypervisor.

Cloud computing and virtualization share many features, but the areas of differentiation may be the areas of self-provisioning, granular billing and APIs. In other words, virtualization is a fundamental element of cloud, but it doesn't address the business and economic benefits.

Some virtualization technologies examples are ESX Server from VMware, HyperV from Microsoft, Xen and KVM as an alternative open source solution.

2.2.2 Utility computing

Often, the cloud computing infrastructure resides in a large data center and is managed by a third party, who provides computing resources as if it were a utility such as electricity. This business model of computing was not new, but it was defined in a so called Utility Computing.

Utility Computing defines a "pay-per-use" model for using computing services. In utility computing, billing model of computing resources is similar to how utilities like electricity, gas, telephone, water are traditionally billed.

The idea was first propounded by American computer scientist John McCarthy of MIT as early as 1961, when he had said, "*If computers of the kind I have advocated become the computers of the future, then computing may someday be organized as a public utility just as the telephone system is a public utility… The computer utility could become the basis of a new and important industry*" [28].

Cloud computing can be perceived as a realization of utility computing. In addition cloud offer to developers and IT operations the opportunity to develop, deploy and run applications that can benefits from scalability performance and reliability offered by cloud computing, all without any concern as to the nature and location of the underlying infrastructure.

2.2.3 Grid computing

Grid computing is a form of distributed computing, where more than one computer from multiple administrative domains coordinate to reach a common goal. In 2002, Ian Foster [29] proposed a definition of the Grid as "*a system that coordinates resources which are not subject to centralized control, using standard, open, general-purpose protocols and interfaces to deliver nontrivial qualities of service*". Grid computing initially was driven by scientific applications which are usually computing-intensive. A well-known example of this

the Search for Extraterrestrial Intelligence (SETI)@Home[7]. In this project, people all over the world allow the SETI project to share the unused cycles of their computers to search for signs of intelligence in thousands of hours radio data.

Due to a lack of standard definition of cloud computing, grid computing is often confused with cloud computing. Cloud computing can be considered as an evolution of grid computing concept. Grid computing and cloud computing share many distributed system perspectives, both offer scalability through load balancing, and both involves multitenancy *(see appendix 2)* and multitask. Figure 2.5 shows, how often the term cloud computing vs. Grid Computing was entered in Google search relative to the total search-volume across various regions of the world, and in various languages.

The only differentiating factor between the two is the method it adopts for computing the tasks within their individual environments. Grid computing requires the use of software that can divide and farm out pieces of a program as one large system image to several thousand computers. Major disadvantage with grid computing is, if one piece of software on a node fails, other pieces of the software on the other nodes may fail [30].

The most adopted business model for grids is project-oriented in which the users or community represented by that proposal have certain number of service units (i.e. CPU hours) they can spend. For instance the TeraGrid operates in this fashion, and requires increasingly complex proposals be written for increasing number of computational power [31].

Cloud computing is the delivery of computing as a service through the Internet. Cloud users access services based on their requirements without regard to where the services are hosted or how they are delivered while grid computing involves sharing of tasks over multiple computers. Some applications like spreadsheets, presentations, email and word processors are part of cloud computing whereas grid computing mainly is designed for data-intensive

[7] http://setiathome.berkeley.edu

applications. Table 1 compares different features of grids and clouds. Vaquero et al. [3] has identified the following differences:

- *Resource Sharing:* grids allow resources sharing across organizations, whereas cloud computing does not allow sharing of resources due to the isolation through virtualization.
- *Virtualization:* both cloud and grids offer resource pooling through virtualization (data, computing resources) with the difference that cloud computing adds hardware virtualization.
- *Security:* cloud ensures security through virtual machine isolation and unique access to individual virtualized environment. Grid offer security services and delegation to access all the available shared resources.
- *High Level Services:* grid offers many services such as metadata search, and data transfer. A cloud does not offer this capability due to the problem of maturity. In the cloud, this issue can be treated at the application level.
- *Software Workflow:* grids are essentially service and job oriented they imply the need to perform the coordination of the services workflow and location which is not necessary in on-demand deployments such as those in the clouds
- *Scalability:* having a single owner, cloud offers better scalability and dynamic reconfiguration. In addition cloud offers hardware scalability while grid scalability is mainly enabled by increasing the number of working nodes.
- *Usability:* grid is hard to manage because are complex and needs a previous understanding of the architecture. While cloud hide the deployment details through a simple interface.
- *Standardization:* grids have reached a certain degree of standardization both in the user interface and the inner interfaces (accessing resources). Interoperability in cloud is a major issue due to the lack of standardization. Cloud is always managed by his owner, who keeps hidden the details implementation.

- *Payment Model:* grids services are billed using a fixed rate per service or different organizations sharing idle resources. While cloud computing uses a pay-per-use model.
- *Quality of Service:* the quality of service is addressed at the application level in the grid, while in the cloud environment is up to service provider to guarantee a certain level of quality of service.

2.2.4 Service Oriented Computing

Another topic that is considered as one of key technologies of cloud computing is Service Oriented Computing (SOC) due to the fact that everything in cloud computing is delivered as a service rather than a product and the origin of the SOC which is derived from the Service Oriented Architecture (SOA) which is considered the umbrella that describes any kind of services [33]. However, the basic SOA does not address well-known concerns such as management, service orchestration, service transaction management and coordination, security, and other concerns that apply to all components in services architecture.

SOC is defined in as [34] *"The Service-Oriented Computing (SOC) paradigm refers to the set of concepts, principles, and methods that represent computing in Service-Oriented Architecture (SOA) in which software applications are constructed based on independent component services with standard interfaces ".* SOA does not address key aspects in cloud computing. Cloud computing emphasizes significantly about its high-performance server aspects. In addition, cloud environment often has hundreds of thousands processors with numerous disks interconnected by dedicated high-speed networks.

Feature	Grid	Cloud
Resource Sharing	Collaboration (VOs, fair share).	Assigned resources are not shared.
Virtualization	Virtualization of data and computing resources.	Virtualization of hardware and software platforms.
Security	Security through credential delegations.	Security through isolation.
High Level Services	Plenty of high level services.	No high level services defined yet.
Software Workflow	Applications require a predefined workflow of services.	Workflow is not essential for most applications
Scalability	Nodes and sites scalability.	Nodes, sites, and hardware scalability.
Usability	Hard to manage.	User friendliness.
Standardization	Standardization and interoperability.	Lack of standards for Clouds interoperability.
Payment Model	Rigid	Flexible
QoS Guarantees	Limited support, often best-efforts only.	Limited support, focused on availability and uptime.

Table 1: *Grid vs. Cloud Characteristics* [3]

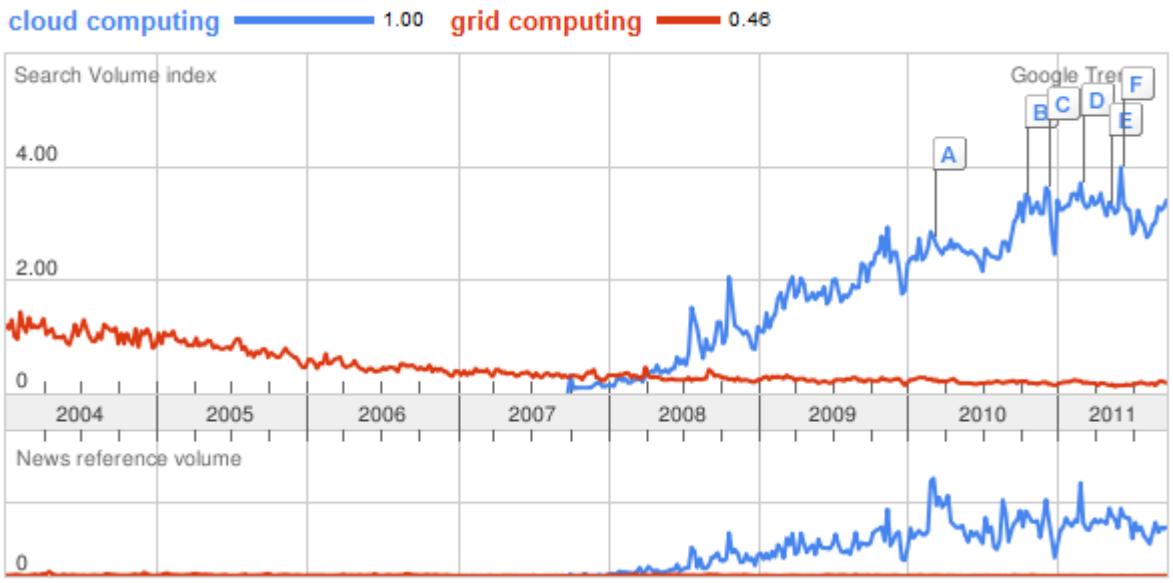

Figure 2.5: *cloud computing vs. Grid Computing on Google trends* [32]

2.3 Conclusion

This chapter discussed the state-of-the-art of cloud computing. The combination of service models and delivery models leads to a lot of possible cost-effective cloud solutions. This chapter provides a literature review for cloud computing-related technologies. Cloud computing is not a new technology; it is rather a new way to provide computing services combining well-known distributed systems paradigms and technologies. For accountability, even if cloud computing is considered a large-scale distributed system it was interesting to highlight some differences with those paradigms to let us understand what are the requirements that must be addressed.

Chapter 3

Quality of Service

Since Internet has become indispensable in our daily life and work, the interest for Internet applications is in a constant growth, and Quality of Service(QoS) playing a key role in any network services deal. This chapter introduces the concept of QoS and SLA as a contract which determines the service level requirements and sets out responsibilities and priorities.

3.1 QoS

Everything in cloud computing is delivered as a service. In order to increase the level of confidence between all cloud actors and to make a commercial success of cloud computing the ability to deliver QoS guaranteed services is crucial. This means any lack of predictable performance is an issue that must be addressed.

There is no unique definition in the literature for QoS, but different definitions can be found depending on the field to which it is applied. QoS has its origin in the networking community where it is defined by Crawley et al. [35] as "*a set of service requirements to be met by the network while transporting a flow* [8] ". QoS in the field of telephony was defined in the ITU-T Recommendation E.800 as "*The collective effect of service performance, which determines the degree of satisfaction of a user of the service*" [36].

As defined in [37] in the field of telecommunication the QoS has at least three distinct but interrelated notions:

- *Intrinsic:* this service notion can be obtained through technical aspects which include the transport network design and provisioning of network access, terminations, and connections [37]. Intrinsic QoS is evaluated by the comparison of

[8] flow represents a stream of IP packets from source to destination

measured and expected performance characteristics and verified by demonstration that those scores compare favorably with analogous score of competing services.

- *Perceived:* perceived QoS is influenced by customer's expectations compared to an observed service which reflects the customer's experience of using a particular service.
- *Assessed:* this level of quality can be seen when the customer decides whether to continue using the service or not [37].

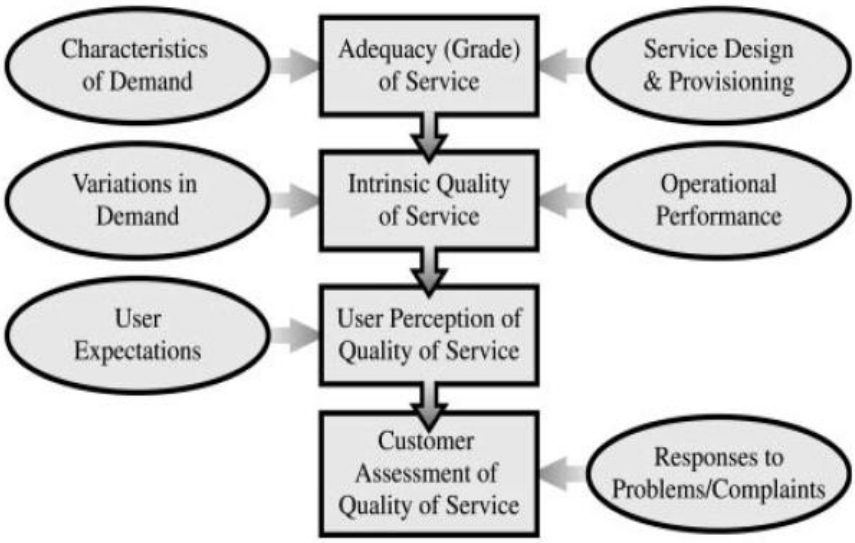

Figure 3.1: *Simplified model of factors that shape perception of quality of service* [37]

Figure 3.1 illustrates the essential distinctions of the three notions above. As suggested in figure 3.1, the perceived service may be influenced by representations by the service vendor as how the service will compare to others with which a user may be familiar as the QoS with the same intrinsic features may be perceived differently by various customers [37]. The assessed notion of QoS is affected by the perceived quality, cost of service, and responses of the provider to submitted complaints and problems.

Certain types of network traffic that QoS may be required are streaming media (such as IP television, audio over IP), VoIP, videoconferencing, circuit emulation service etc.

3.2 Service Level Agreement

The enterprise applications, unlike the normal applications, have a quite strict set of quality of service requirements. As IT decision makers move towards adopting cloud computing in their business, the quality and reliability of the services become crucial aspects. However the demands of the service consumer vary significantly. Cloud providers will need to consider and meet different QoS parameters of each individual consumer as negotiated in the SLAs.

Buco et al. [38] define SLA as "*an IT service contract that specifies the minimum expectations and obligations that exist between the provider and the customer of a utility computing service. It includes one or more service level components, each of which specifies the measurement, evaluation, and reporting criteria for an agreed service-quality standard*". This means a series of qualities and characteristics, which are typically non-functional requirements such as availability, scalability, reliability and timing of response that an application require in order to be transformed into functional requirements such as calculations, technical details, data manipulation and processing and other specific functionality.

SLAs are not a new concept for service providers or the customers that purchase them; it was firstly used since late 1980s by fixed line telecom operators as part of their corporate customer [39]. However, there is no standard SLAs as customer's requirements may vary from domain to domain and from service to service. In addition, standardization may introduce risk by failing to address and accommodate the specific security requirements of individual customers [40].

For instance, the closest SLA to the cloud computing paradigm is the Web Service Level Agreement (WSLA) [41]with the difference of the technology added and most often a third party management or monitoring provider. In order to make the SLA monitoring and reporting mechanisms applicable in the cloud computing environment, actors should take into considerations several complex aspects:

- *Dynamic environment:* due to the fact that cloud environment is dynamic and the resource usage changes dynamically, a system that tries to enforce SLA need to take into account this aspect.
- *Performance interference:* in order to achieve the goal of the economy of scale cloud providers must employ virtualization to encapsulate workloads in virtual machines (VMs) and consolidate them on multicore server. However, as illustrated by Ripal et al in [42] virtualization does not guarantee performance isolation between VMs. In addition Ripal et al. suggest the following *"We argue that interference should instead be addressed proactively with performance SLAs".*
- *Data location:* unlike traditional paradigms which offer a single dedicated server, cloud providers use geographically distributed servers in the cloud and this may conduct to a non-compliance with data protection directive such as the EU directive 95/46/EC [43] which regulates the processing of personal data within the European Union.

Creating a good SLA is not a trivial task. However, there are methods of making it more manageable such as using template or toolkit. According to SLA Information Zone [44] a well-defined SLA should usually include the following:

- *Services to be delivered:* describes the services and the manner in which those services are to be delivered. This agreement should be very detailed and accurate. In this section it is a preferable to separate standard services from customized services.
- *Performance:* deals with how monitoring and measuring the service level performance. The service performance level must be reviewed regularly by the two parties. Performance metrics of services are often measures up to a percentage level. For example a P-Percentile means that p% less than a pre-defined time value in the SLA must be guaranteed for a given customer otherwise legal penalties should be applied. An example of this is in Amazon EC2 SLA [45] where they state the following: *"AWS will use commercially reasonable efforts to make Amazon EC2*

available with an Annual Uptime Percentage (defined below) of at least 99.95% during the Service Year. In the event Amazon EC2 does not meet the Annual Uptime Percentage commitment, you will be eligible to receive a Service Credit as described below".

- *Problem management:* how to deal with unplanned incidents and how to solve them, also including how to actively prevent such events.
- *Customer duties and responsibilities:* explains what relationship the customer and the provider has and the responsibilities that the customer has regarding the service delivery process.
- *Warranties and remedies:* covers topics such as service quality, third part claims, exclusions and force majeure.
- *Security:* the most critical feature of any SLA where which security approaches must be followed and respected.
- *Disaster recovery:* usually included in the security section and sometimes also in the problem management area.
- *Termination:* covers topics as for example termination at end of initial term, for convenience, for cause and payments regarding termination.

Actually there are two types of service agreements [46]:

- *Predefined non-negotiable agreements:* the terms of service are prescribed completely by the cloud provider with risks regarding federal privacy and security requirements. In addition the provider can make modifications to the terms unilaterally.
- *Negotiated service agreements:* comparable to traditional outsourcing contracts for information technology service. They can include organization's requirements about security, privacy policy, procedures and technical controls.

3.2.1 SLA Lifecycle

A SLA lifecycle is a mean by which we can govern a service level definition from being initially identified, through to being retired when it is no longer in use. As cloud deals with several

issues such as scalability, heterogeneity latency distribution and segmentation [47], a well SLA life cycle may influence the level of the cloud business success. According to Buya et al. [48] SLA life cycle goes through a sequence of steps and consists of the following phases:

- *Contract definition:* Through standard templates service providers define a set of service offerings and corresponding SLAs.
- *Publication and Discovery:* In this phase the service provider advertises these base service offerings through a standard publication media.
- *Negotiation:* SLA terms and conditions needs to be mutually agreed after the selection of the service provider who can meet their application hosting need.
- *Operationalization:* After that the SLA has been signed off, cloud provider should start the service and assure that there is no SLA violation by involving SLA monitoring, SLA accounting and SLA enforcement.
- *De-commissioning:* This phase deals with the end of the relationship between the service provider and the service customer. SLA specifies the terms and the contract termination.

3.2.3 SLA Management

One of the most important challenges after the definition of the SLA is how to manage the deployment, distribution and configuration of required resources in order to offer a service that honor SLA. According to Buya et al. [48] there are five management phases which are feasibility analysis, on-boarding, pre-production, production and termination. Different activities performed under each of these phases are shown in Figure 3.2.

These activities are briefly explained in the following:

- *Feasibility Analysis:* Provider must evaluate his capabilities in terms of technical feasibility, infrastructure feasibility and financial feasibility.
- *On-boarding of Application:* After the agreement was established, application will be hosted. During this phase provider will determine the possible SLAs that can be

offered to the customer application and necessary policies will be created in order to guarantee the Service Level Objectives.

- *Preproduction:* After the agreements on terms and conditions, the customers sign-off the SLA. The application is hosted in a simulated environment and the service is started.
- *Production:* During this phase the application is made accessible to its end users and the behavior of the application monitored under the agreed SLA.
- *Termination:* This phase of the SLA deals with the termination of the relationship between the service consumer and the service provider for the application hosted. During this phase some legal aspects must be reviewed and resolve legal issues before the customer sign-off.

3.2.2 Benefits of SLA

Hence, cloud computing fit well into the definition of utility computing, Sun Internet Data Center Group's report [49] a good SLA sets boundaries and expectations of service provisioning and provide the following benefits:

- *Enhancing customer satisfaction level:* Clearly defined commitments can reduce the possibility of a customer disappointment as it helps to stay focused on customer requirements and assure that the effort is put in the right direction.
- *Improved Service Quality:* Each item in an SLA corresponds to a Key Performance Indicator (KPI). This indicators allow to internal objectives to become clearer and easier to measure.
- *Improved relationship between two parties:* SLA can play important role in making the relationship clear and positive by having penalties clearly defined which increase the customer level of trust and resolve dispute when is verified.

3.3 Federated Cloud Computing

As with any new IT that is new to the market, cloud computing is facing many deficiencies in current offerings such as [50]:

- *Inherently limited scalability of single-provider clouds:* cloud providers serve an increasing number of on-line services which may raise scalability problems.
- *Lack of interoperability among cloud providers:* the absence of interoperability in the current cloud system may results in vendor lock-in problems.
- *No built-in Business Service Management support:* Customers looking at transforming their IT operations to cloud-based technologies face disruptive step because current cloud computing solutions are not designed to support the BSM[9] practices.

With all this deficiencies the need for federated cloud computing has become a necessity more than an option. One definition of cloud federation as proposed by Reuven Cohen of Enomaly follows: [51]*"Cloud federation manages consistency and access controls when two or more independent geographically distributed clouds share either authentication, files, computing resources, command and control, or access to storage resources".* Cloud federation offers advantages for both customer and provider.

From a customer perspective federation offer the opportunity to take advantage of all capabilities available in the cloud without complexity or risk such as lock-in and run workloads in the environment that best matches their needs, based on cost, performance, security, compliance geography latency, etc.

From a provider perspective federation allows a provider to fulfill their customers' requirements when problem of restricted amount of resources is faced by outsourcing resources to other providers in response to demand variations [52]. Federation allows a provider that has underused resources to rent part of them to other providers.

In order to fulfill the promise of cloud computing as the 5th utility service there should be the technological capabilities to federate data centers belonging to different organizations. Only through the federation and interoperability, the cloud provider may take advantage of their aggregated capabilities and provide a seemingly infinite pool of available resources.

[9] Business Service Management (BSM) is a management strategy that allows businesses to align their IT management with their high level business goals

Cloud computing offerings still suffers the lack of support for the Business Service Management (BSM), in particular for the management of Service Level Agreement (SLA).This lack of support come from the fact that the service providers find it impractical to create personalized offerings with consequences such as translate business requirements into technical manifestations and optimize internal deployments for each SLAs [53].

In the next section RESERVOIR[10] will be taken as a reference model that addresses some of this deficiencies and serves as a potential foundation for delivering IT services as utilities over the Internet. In addition, an overview on SLA@SOI which is a project similar to RESERVOIR but with different architecture and use.

3.3.1 RESERVOIR

Resources and Services Virtualization without Barriers (RESERVOIR) is an EU-funded research project within the FP7[11] which aims to support the emergence of Service-Oriented Computing (SOC) as a new computing paradigm [54] (*see appendix 2 SOC*). In addition, the vision proposed by this project aims to create an architecture that enables a massive scale deployment and the management of heterogeneity across cloud environment (e.g. EC2 and Rackspace) giving the illusion of endless resources, while at the same time preserving the technological autonomy, and the business goals of each partners.

Approach

According to the project factsheet [55], their goal is "*to increase the competitiveness of the European Union economy by introducing a powerful Information and Communication Technologies infrastructure for the reliable and effective delivery of services utilities. This infrastructure will support the setup and deployment of services on demand, at competitive costs, across disparate administrative domains, while assuring quality of service*."

In order to reach the goal of cloud computing as 5th utility RESERVOIR approach is extending combining and integrating three core technologies: Virtualization, Grid computing and

[10] RESOURCES AND SERVICES VIRTUALIZATION WITHOUT BARRIERS

[11] Seventh Framework Program for research and technological development

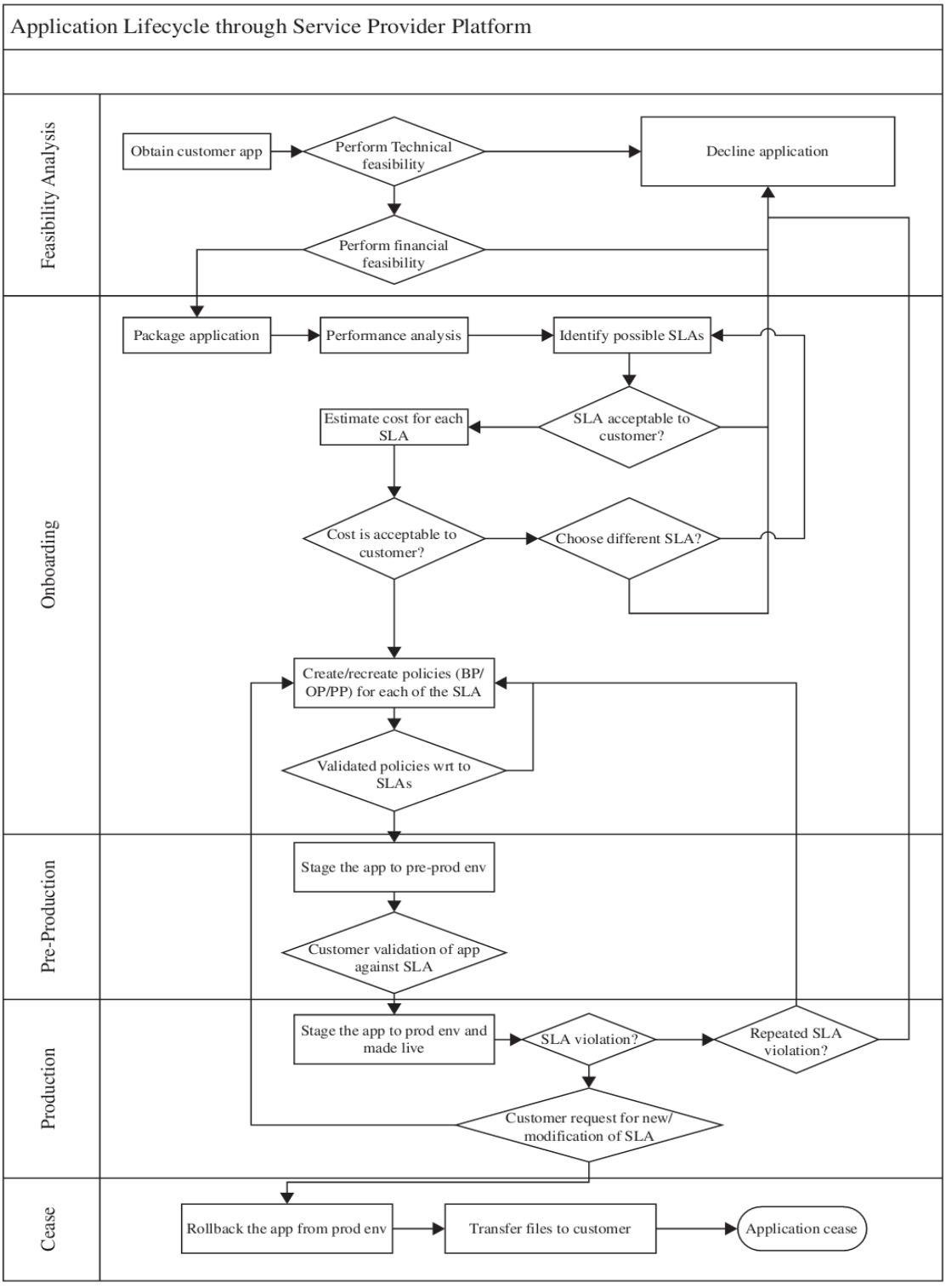

Figure 3.2: *Flowchart of the SLA management in cloud* [48].

Business Service Management (BSM) [54]. Figure 3.4 shows a high-level description of RESERVOIR architecture.

The RESERVOIR approach combines the paradigm of cloud computing with the paradigm of Service-Oriented Computing. Service-Oriented Computing (SOC) is a paradigm for programming distributed applications by means of the composition of services to support business processes and users. SOC is built on the Service Oriented Architecture (SOA).

Services in SOC are reusable, autonomous, loosely coupled, platform independent and can be published and discovered. RESERVOIR address lacking SOC functionalities such as end – to-end security, service deployment, management and orchestration, service billing and interpretation and monitoring of Service Level Agreement (SLA) conditions.

The main actors in the RESERVOIR cloud are service providers entities that understand the needs of a particular business and infrastructure providers which them provide with a seemingly infinite pool of computational, network and storage resources. RESERVOIR deploys a Service Application [12] using a service manifest that formally defines the contract and SLA between the service provider and the infrastructure provider.

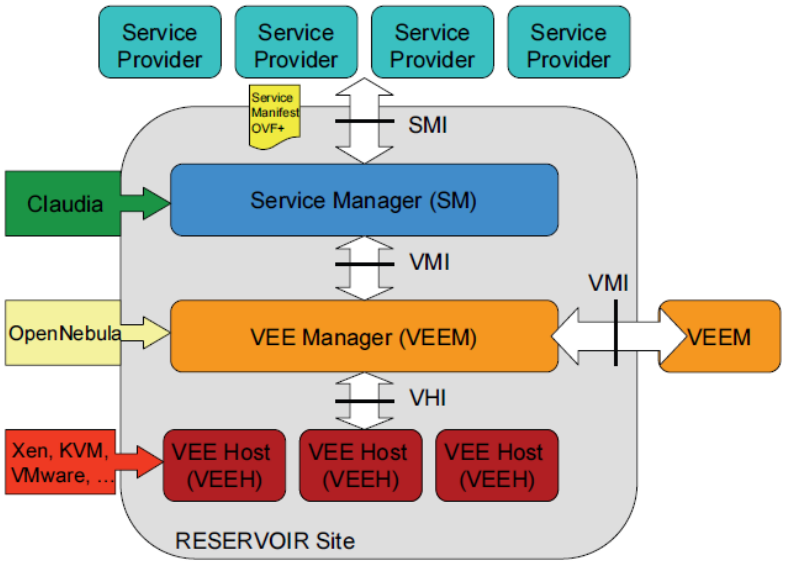

Figure 3.4: *RESERVOIR Architecture*

[12] Service Application is a set of software components that work collectively to achieve a common goal

RESERVOIR Components

One of the core principles of RESERVOIR is to keep clear separation of concerns and responsibilities and to hide low level infrastructure details and decisions from high-level management and service providers by using the following three components:

- *Service Manager(SM):* The highest level of abstraction. The role of Service Manager is to interact with the service providers to receive Service Manifests, negotiate pricing, and handle billing. In addition SM ensures SLA compliance and alignment with high-level business goals by monitoring the deployed services and adjusting their capacity (i.e. memory, CPU, etc.)
- *Virtual Execution Environment Manager (VEEM):* VEEM acts as a coordinator between the SM, the Virtual Execution Environment Host (VEEH) and the VEE Managers at other sites to enable federation. VEEM is responsible for the placement of VEEs into VEE hosts subject to constraints determined by the Service Manager.
- *Virtual Execution Environment Host (VEEH) :* The lowest level of abstraction and is responsible for the basic control and monitoring of VEEs and their resources (e.g., creating a VEE, allocating additional resources to a VEE, monitoring a VEE, migrating a VEE, creating a virtual network and storage pool, etc.).

3.3.4 SLA@SOI

In order to follow the trend of a rapidly growing service-oriented economy, cloud providers face many challenges and need to know what level of service they can guarantee to their customers. Furthermore from a service provider perspective creating customized service offerings, negotiating with individual customers, and translating from business requirements into specific internal provisioning manifestations consumes time and resources. The fulfillment of each individual consumer expectations from the service provider perspective impossible and hence a balance need to be made via a negotiation process.

The SLA@SOI's goal is "*to deliver and showcase an innovative open SLA Management Framework that provides holistic support for service level objectives - enabling an open, dynamic, SLA-aware market for European service providers* [56]

The SLA@SOI project vision is developing an exhaustive framework that automates the negotiation of SLAs in e.g. cloud computing and also taking into account the QoS. In addition, SLA@SOI goal is to harmonize the perspective of all stakeholders (CSPs and customers), develop a machine readable standard for SLA specification and give guaranteed quality of service according to the SLAs.

In order to achieve a good generalization and to provide the advantage of integration with other Resource managers, the SLA@SOI architecture mainly focuses in separation of concerns between Service management and SLA management.

Service Managers are responsible for all management activities directly related to service such as the management of information about available services [54] and communication with customers and providers.

SLA Managers are responsible for all actions that are related to service level agreements such as negotiation planning and optimization of new services. In addition, the monitoring function allows to service managers to react in case of a SLAs terms violations [54]

SLA@SOI has a vertical architecture for SLA management and starts from the highest level where some business entity goals are defined and consequently all management activities supported by the framework should refer to the needs of that business entity. Figure 3.5 give an idea on SLA@SOI architecture.

SLA@SOI provides three major benefits to the provisioning of services [57]firstly, service predictability and dependability means that the quality characteristics of service can be predicted and enforced at run-time. Secondly transparent SLA Management means that the exact conditions under which services are provided or consumed can be transparently managed across the whole business IT stack.

Third, automation means that the whole process of negotiating SLAs and provisioning, delivery and monitoring of services can be automated allowing highly dynamic and scalable service consumption.

SLA@SOI and RESERVOIR

SLA@SOI and RESERVOIR frameworks aim to guarantee the quality of service in cloud computing and both have automatic and elastic management solutions which control the life cycle, placement and SLAs of a Cloud services. Both projects contribute in the Open Cloud Computing Interface (OCCI) which is a set of open community-lead specifications providing protocol and API for all kinds of management tasks using REST-style(Representational State Transfer) [58] (*see appendix 2 OCCI/REST*). However, SLA@SOI and RESERVOIR have clearly different architectures. RESERVOIR follows a horizontal approach and focuses on federated clouds with focus upon interoperability and optimization of the management of virtual workloads across administrative domains.

SLA@SOI follows a vertical approach and focuses on SLA-driven management and monitoring the life-cycles of services such as SaaS, IaaS. SLA@SOI is not limited to cloud computing environments but can be applied to various other domains such as D.B6a use case specification e-government [59] where it's applied to human based services.

3.4 Conclusion

This chapter presented a detailed overview of SLA which defines different QoS parameters of each individual consumer. It described a brief history of how the SLA that evolved from application in the field of telecommunication to cloud computing. The chapter covered the main considerations to make SLA applicable in the cloud environment and the requirements for a well-defined SLA. In addition this chapter discussed the needs for a federated cloud computing as standardization in the cloud environment is needed in order to meet the expected QoS.

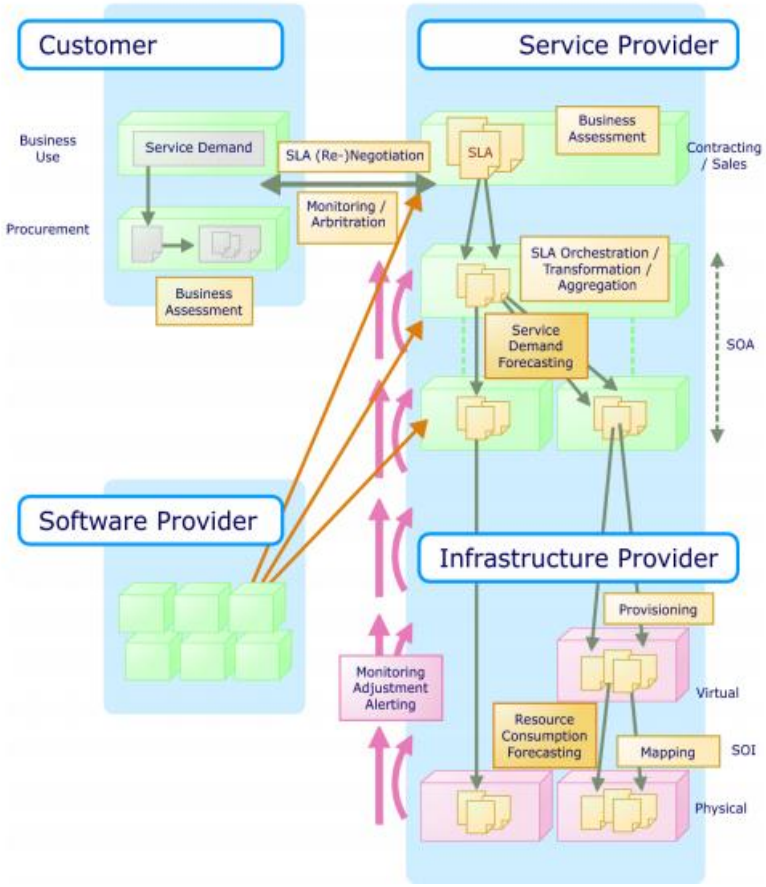

Figure 3.5: *The Highest Level of SLA@SOI* [60]

Chapter 4

Accountability

In the previous chapter we have introduced SLAs as a legal means that specifies the minimum expectations and obligations that exist between the provider and the customer. However, cloud SLAs are not enough to address the mistrust. Cloud computing platforms are effectively an outsourced computing environment and like any other new IT environment in which there is a rapid innovation, cloud computing facing skepticism due to many challenges such as performance, confidentiality or privacy, and security issues and data availability. This chapter defines accountability and describes several challenges that are not yet met by current accountability techniques. In addition, this chapter outlines the technical requirements for an accountable cloud by analyzing different solutions proposed for the distributed system. Finally this chapter discusses some proposed solutions with the purpose to make cloud accountable.

4.1 What's accountability?

Accountability is a concept susceptible to a variety of different meanings within and across disciplines. For example as stated in [61] there are 8 types of accountability, namely: moral, administrative, political, managerial, market, legal/judicial, constituency relation, and professional. Depending on the context in which the term accountability is used there are various refinements of this definition appropriate to the context in question but, it is often used synonymously with such concepts as responsibility and answerability [61]. Koppell in his work [62] argues that providing accountability single definition would render the concept meaningless and describes five dimension of accountability with five relative questions that needs to be answered when a conflict arise. The five dimensions and relative questions are given in the table 4.1.

Conception of accountability	Key determination
Transparency	Did the organization reveal the facts of its performance?
Liability	Did the organization face consequences for its performance?
Controllability	Did the organization do what the principal (e.g., Congress, president) desired?
Responsibility	Did the organization follow the rules?
Responsiveness	Did the organization fulfill the substantive expectation (demand/need)?

Table 4.1: *Conceptions of Accountability*

Koppell consider transparency and liability as the foundations of the accountability's concept. Transparency is the most important dimension as it provides an instrument for assessing organizational performance and a key requirement for all other dimensions of accountability. Liability means that individuals and organizations should be held liable for their actions.

A generic definition of accountability from Merriem-Webster Dictionary is "*The state of being accountable; liability to be called on to render an account; the obligation to bear the consequences for failure to perform as expected; accountableness.*" [63].

The purpose of this thesis is to examine the accountability in distributed computing systems and particularly in cloud computing which is an emerging field that still suffering from trust due to the outsourcing nature and the lack of mechanisms that guarantees transparency and liability. A useful definition of accountability in this context is given by Chen et al. [64] as "*Accountability is a concept to make the system accountable and trustworthy by binding each activity to the identity of its actor. Such binding should be achieved under the circumstance that all actors within the system are semi-trusted".*

According to ENISA [65] Accountability offers three capabilities:

- *Validation:* it allows users to verify in a later time if the system has performed data processing as expected.
- *Attribution:* in case of fault, users can assign responsibility.

- *Evidence:* it can produce evidence that can be used to convince a third party when a dispute arises.

Accountability is often confused with fault-tolerance or responsibility (see next section). Fault-tolerance is defined in [66] as ***"****The ability of a system to respond gracefully to an unexpected hardware or software failure".* Which makes accountability different from fault-tolerance is that it does not attempt to mask faults, but it provides evidence and it may detect arbitrary faults.

4.1.1 Accountability and Responsibility

A prerequisite for understanding the accountability in cloud computing is to understand the difference between responsibility and accountability because are often used interchangeably.

Responsibility defined in [67] *as "A duty or obligation to satisfactorily perform or complete a task (assigned by someone, or created by one's own promise or circumstances) that one must fulfill, and which has a consequent penalty for failure"*. Responsibility is the obligation to do something according to certain parameters. Accountability can be seen as the ultimate responsibility. In [68], Hickman suggests that *"Responsibility may be bestowed, but accountability must be taken. In other words, responsibility can be given or received, even assumed, but that doesn't automatically guarantee that personal accountability will be taken. Which means that it's possible to bear responsibility for something or someone but still lack accountability"*. For example, a manager who delegates task to a subordinate makes this subordinate responsible for the task, but remains accountable in case something goes wrong. In other words, the manager can delegate responsibility, but not accountability.

Though cloud computing is a major paradigm shift, it does not change the assignment of accountability. Cloud customers are accountable for their assets, including any assets that were outsourced to cloud providers. Cloud computing only transfers the responsibility to perform compute task according to certain parameters, but accountability remains. The responsibility for a given task in cloud computing depends on the delivery model chosen (SaaS, IaaS or PaaS). Figure 4.1 gives an idea on how this responsibility is shared. As

illustrated in figure 4.1 for the IaaS model customer can control the stack from the operating system to the application. Whereas for SaaS and PaaS models typically place much of the responsibility for data security and control in the hands of SaaS and the PaaS provider. In these two models the CSP owns most of the IT and security stack. IaaS offers a great degree of shared responsibility. IaaS providers typically provide some baseline level of security, such as firewalls and load balancing to reduce the risk of a distributed denial of service attacks. For example AWS customer agreement state that customer should be responsible [69] for the following things:

- *Content:* customer is solely responsible for the development, content, operation, maintenance, and use of own Content.
- *Other Security and Backup:* customer is responsible for properly configuring and using the Service Offerings and taking his own steps to maintain appropriate security, protection and backup of his Content, which may include the use of encryption technology to protect his content from unauthorized access and routine archiving his content. AWS log-in credentials and private keys generated by the Services are for his internal use only and he may not sell, transfer or sublicense them to any other entity or person, except that he may disclose his private key to his agents and subcontractors performing work on his behalf.
- *End User Violations:* customer is responsible for End Users' use of his Content and the Service Offerings.
- *End User Support:* customer is responsible for providing customer service to End Users. Additional services must be previously and separately agreed with AWS.

4.1.2 Accountability in Computing Systems

Accountability is not a new concept but it was largely discussed during the past two decades as an emergent requirement. This requirement has been further accelerated since 1990s by the widespread availability of broadband Internet access for small businesses and individual domestic users. An early example of this recognized need for accountability in computing systems is a 1994 article by Helen Nissenbaum [70]. Nissenbaum recognized that strong

culture of accountability may contribute to develop and encourage a sense of responsibility within community.

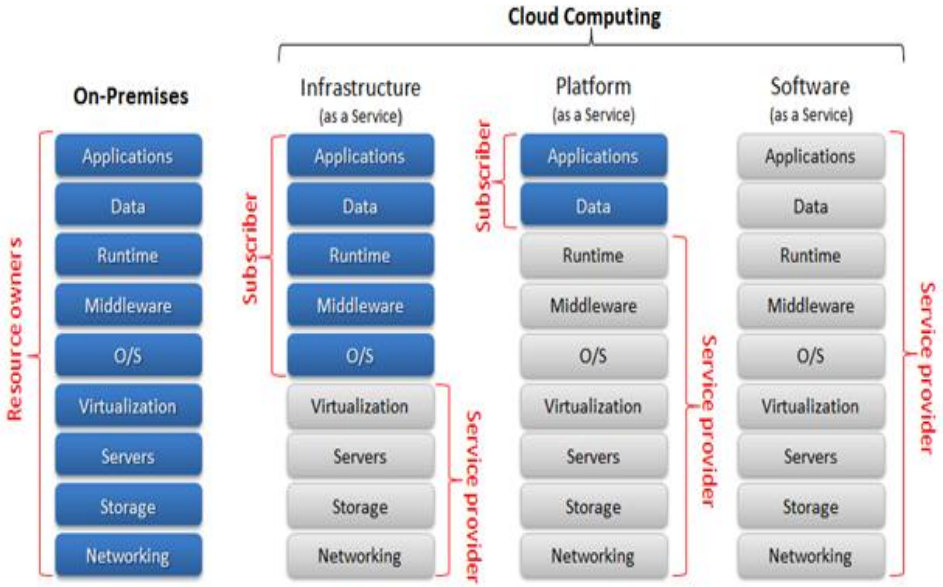

Figure 4.1: *Separation of responsibility in cloud computing* [71]

For Nissenbaum, accountability is particularly important for a technology still struggling with standards of reliability, because is a means through which the end user is assured of answerability in case of an event of malfunction and the absence of such accountability provides a situation where no individual is answerable for risks or harm. She identified four barriers for the adoption of rigorous accountability mechanisms:

- *The problem of many hands:* computer systems are a result of coaction of individuals with a diverse range of skills and varying degrees of expertise, such as designers, engineers, programmers, writers, psychologists, graphic artists, managers, and salespeople. The collective action may render difficult the task of assigning responsibility and obscure accountability.
- *Bugs:* bugs cause problems but are also commonly regarded as an inevitable element of computer systems which includes a range of acceptable error.
- *The Computer as scapegoat:* the problem of assigning responsibility to the computer which is a stopgap that may cause the loss of accountability.

- *Ownership without responsibility:* while systems providers were eager to assert rights of ownership, the responsibilities of ownership were being neglected.

Nissenbaum provides three recommendations to encourage accountability in the emerging computerized society:

- *An explicit Standard of Care:* offers a way to distinguish between malfunctions and the result of inadequate practices.
- *Distinguishing accountability from liability:* even if liability and accountability are frequently connected their conceptual meanings are sufficiently distinct.*" One key difference is that appraisals of liability are grounded in the plight of a victim, whereas appraisals of accountability are grounded in the relationship of an agent to an outcome"*.
- *Strict Liability and Producer Responsibility:* the compensation for harm even if not intentionally caused.

4.2 Security, Privacy and Trust challenges

When cloud's customers adopt cloud for their business, they are ready to lose control while maintaining accountability even if the operational responsibility falls upon one or more third parties. Currently, customers can at best monitor the virtual hardware performance metrics and the system event logs of their services models. In order to understand the need for accountability as an emergent concept that can motivate and encourage cloud computing's business, this section focuses on concerns when customers try to move to the cloud. The purpose of this thesis is not to cover all risks related to cloud but to highlight some interesting concerns where accountability mechanisms can play an important role to omit some skepticism and enhance the level of trust and confidence in the cloud computing environment.

4.2.1 Security

One of the most important barriers that holding back the adoption of the cloud is the security concern. Being based on the Internet does make cloud computing susceptible to cyber-attacks. According to CSA[13] [72] which make security different in cloud computing from the traditional IT solutions are the cloud service models employed, the operational models and the technologies used to enable cloud services. CSA in [73] identifies 7 critical areas for cloud security risk management:

- *Abuse and Nefarious Use of Cloud Computing:* many cloud computing service providers offer free trials or immediate access to cloud storage space. Through the different service layers, specifically PaaS and IaaS, malicious users could take the opportunity to learn the environment of the cloud to develop viruses and other malicious code. The relative anonymity in signing up to use the cloud is not well-regulated, leaving the systems open to vulnerabilities by malicious code writers. The power of computing offered by cloud can be used in the future for password and key cracking, DDOS, Launching dynamic attack points hosting malicious data, botnet command and control, building rainbow tables, and CAPTCHA solving farms. For example the Zeus botnet [74] used Amazon's Elastic Computing Cloud (EC2) to host the central server used to control a portion of the compromised machines.
- *Insecure Interfaces and APIs:* As this is an external link into the cloud services, there is a threat associated with the security of these APIs. As these APIs are serving as a gateway into the cloud, there is a higher risk for the organizations that must pass their credentials to this interface to gain access to the cloud.
- *Malicious Insiders:* If a provider is not revealing how its employees are granted access to both physical and virtual resources how it monitors these employees, or how it analyzes and reports on policy compliance there is certainly a cause for

[13] CSA = "Cloud Security Alliance (CSA) is a not-for-profit organization with a mission to promote the use of best practices for providing security assurance within Cloud Computing"

concern about who is handling the customer's data. Sensitive client's data could get into the wrong hands or adversary can gain complete control over the cloud services with little or no risk of detection.

- *Shared Technology Issues:* even if server consolidation increases the efficient use of server resources through sharing there is a threat that a client can either maliciously or accidentally gain control of the underlying infrastructure through the virtual system. Most of this concern, according to the CSA, is attributed to the fact that *"underlying components that make up this infrastructure were not designed to offer strong isolation properties for a multi-tenant architecture."*
- *Data Loss or Leakage:* this is one of the most critical threats to cloud computing because data loss to a business or organization could be catastrophic and cost the business financial and legal implications. For example BPOS: a data leak in Microsoft's cloud which allows non-authorized users to download data. [75]
- *Account or Service Hijacking:* in addition to traditional attack methods such as phishing, fraud, and exploitation of software vulnerabilities still achieve results. If the credentials are hijacked from an insecure session, the entire cloud environment and data may be vulnerable to attack. If credentials are compromised, confidentiality, integrity, and availability can all be compromised as well.
- *Unknown Risk Profile:* **c**loud computing is a new technology with rapid innovations and offers, and like any other new IT is anticipated by benefits and by groups who may lose track of the security ramifications. Customers before adopting the cloud as an alternative IT choice should evaluate factors such as code updates, security practices, vulnerability profiles, intrusion attempts and security design. Questions on the cloud service provider's compliance of the internal security procedures such as configuration hardening, patching, auditing, and logging are often not clearly answered or are overlooked leaving customers with unknown risk profile that may include serious threats.

A research conducted in [76] using Amazon EC2 service as a case study to illustrate that is possible to introduce others new vulnerabilities in cloud computing. Hiring VMs from Amazon to act as victims, researchers noted that if multiple VMs are bought in a small span of time, the VMs would have similar IP addresses, which indicated that they might be on the same physical server. Realizing this meant that they needed to hire the attack VMs at the same time as their victim would hire her VMs. One way they suggested to achieve this in a real world setting would be to bombard the intended victim's website with requests, encouraging them to hire additional VMs to handle the sudden spike. This is greatly eased if the hiring of such VMs for spikes is automated. The researchers found that the attack VMs ended up on the same server 40% of the time. The researchers then used the attack VMs that were on the same server as the victim VMs to monitor the victim VMs' use of computing resources. The researchers claim that though they did not steal any data, possibly for legal reasons, they could have. Amazon claims that the possibility of data theft was only theoretical, but they have since taken measures to prevent this kind of attack, though they did not state exactly how.

Customers will have a high expectation on security before moving their business services or business processes to the cloud. However, this is not the case, according to a study released by the Ponemon Institute [77], an independent research firm specializing in privacy and security policies, cloud providers are more focused on delivering cost and speed-of-deployment benefits associated with cloud computing than the underlying security. While enterprises are also keen to leverage those benefits, they would like to see providers accept more responsibility for data security.

"*Given the well-publicized concerns about the potential risks to organizations' sensitive and confidential information in the cloud, we believe it is only a matter of time until users of cloud computing solutions will demand enhanced security systems,*" Larry Ponemon, chairman of the Institute, said in a statement. *"However, until this happens, users of cloud computing should be aware of their responsibility to assess the risks before migrating to the cloud."*

The study, titled "Security of Cloud Computing Providers" was co-sponsored by CA Technologies[14]. It found that the majority of cloud providers (79%) allocate 10% or less of IT resources to security or control-related activities.

Additional findings included:

- Less than 20% of cloud providers in the U.S. and Europe view security as a competitive advantage and fewer than 30% of respondents consider it to be an important responsibility.
- A majority of cloud providers (69%) believe security is primarily the responsibility of cloud users. This contrasts sharply with cloud users, with only 35% believing security is their responsibility.

There is a wide gap between cloud providers and cloud users on the degree to which they see intellectual property as being too sensitive for the cloud. About 68% of enterprises felt their intellectual property was too risky for cloud use, compared to 42% for cloud providers.

Ponemon said cloud users and providers should consider the importance of accepting joint responsibility when structuring a cloud arrangement.

Accountability can play an important role in order to leverage security level as well as tool that helping resolving dispute when arise. Lombardi et al. [78] argue that "*even though cloud services are difficult to trace for accountability purposes, in some cases this is a mandatory application requirement*". They consider "*accountability can increase security and reduce risks for both the service user and the service provider. A trade-off between privacy and accountability exists, since the latter produces a record of actions that can be examined by a third party when something goes wrong"*. As stated in [79] Rose a principal analyst at Forrester Research[15] said that "*As security professionals, we remain accountable for resolving these issues, no matter how much responsibility has been pushed to 3rd parties and cloud vendors*".

[14] http://www.ca.com

[15] http://www.forrester.com

4.2.2 Privacy

Privacy is subject to different interpretations such as privacy of the person, privacy of personal behavior, privacy of personal communications and privacy of personal data [80]. It is beyond the scope of this thesis to examine each interpretation. The closest interpretation we adopt here is the privacy of personal data which is defined in Directive 95/46/EC [81] as follows "*Personal data shall mean any information relating to an identified or identifiable natural person ("data subject"); an identifiable person is one who can be identified, directly or indirectly, in particular by reference to an identification number or to one or more factors specific to his physical, physiological, mental, economic, cultural or social identity*".

Privacy often considered as subset of information security. There are two interrelated conception, but privacy brings a host of concerns all its own [82]. Privacy issues in IT are often addressed along with other security requirements, rather than as a separate design criterion in the system development process.

Privacy is another important concern of holding back the full adoption of the cloud in business, even if modern encryption technologies can guarantee a certain level of security.

The outsourced nature of the cloud means the loss of the immediate physical control in addition to the fact that customers must delegates some tasks to the cloud service provider such as the dynamic allocation to meet a spike in demand and the storage which in this case is not limited by space or geography. These properties from a customer perspective raise valid questions about privacy such as:

- Are hosted data and applications within the cloud protected by suitably robust privacy policies?
- Are the cloud computing provider's technical infrastructure, applications, and processes secure?
- Are processes in place to support appropriate action in the event of an incident that affects privacy or security?

Privacy defined in [83] as "*the ability of an individual or group to seclude themselves or information about themselves and thereby reveal themselves selectively* ". By default, all

data related to users should be private and not accessible by any other service provider unless the user decides to make them (partially or completely) public.

According to Tim et al. [83] privacy add several concerns to the security (even if are not specific to personal information only) such as:

- *Access:* even if cloud service provider can grant access to all individual's personal information and this later exercise the right to ask the organization to delete his data the main concern remain will it be possible to ensure that all his information has been deleted in the cloud?
- *Compliance:* there may be some issues regarding the applicable laws, regulations, standards, and contractual commitments that govern this information and who is responsible for maintaining the compliance. This can be complicated when the data can be stored in multiple countries which raise the following question: What is the relevant jurisdiction that governs an entity's data in the cloud and how is determined?
- *Storage:* there is a risk that data stored in the cloud may be moved across countries without the knowledge of the customer, resulting in a potential violation of the local law.
- *Retention:* this concern deals with the retention period, the retention policy that governs the data and who enforces this policy in the cloud.
- *Destruction:* this concern is about how to ensure that deleted or destroyed Personally Identifiable Information (PII) at the end of the retention period becomes unrecoverable.
- *Audit and monitoring:* how can organizations monitor their CSP and provide assurance to relevant stakeholders that privacy requirements are met when their PII is in the cloud?
- *Privacy breaches:* how do customer know that a breach has occurred, how do, and who is responsible for managing the breach notification process? If contracts

include liability for breaches resulting from negligence of the CSP, how is the contract enforced and how is it determined who is at fault?

Security is an essential component of strong privacy safeguards in online computing environments, but security alone is not sufficient. The fact that cloud computing is relatively different from traditional IT in the sense that there is no backup media to lose, laptop based database to steal, unencrypted or unauthenticated connections to sniff or hijack does not mean that cloud computing is totally immune from a privacy's risks.

For example Google use SAML Single Sign-On (SSO) (*see appendix 2 SAML*) for exchanging authentication and authorization data between Google Apps but a vulnerability was discovered in the Google's implementation that allow a malicious service provider to impersonate a user at other service providers [84].

Another important privacy threat case regarding the use of Dropbox which is a free cloud storage service online with over 25 millions of users. On June 20th 2011, Dropbox announced that after a code change, it discovered that it had accidentally turned off password authentication for its 25 million users for four hours. [85]

Accountability principle states that "*an organization is responsible for personal information under its control and should designate an individual or individuals who are accountable for the organization's compliance with the remaining principles."* [83].

Tim et al. [83] argue that accountability for privacy can be achieved by attaching policies to data and mechanisms to ensure that these policies are adhered to by the parties that use, store, or share the data without regard to the jurisdiction in which the information is processed.

As commonly agreed [8] [86] privacy and accountability may have conflicts, since the latter produces a detailed record of machine's actions that can be inspected by a third party. It is important to consider what is being recorded and who the record is made available. With logs available in the cloud, if attackers gain access to the system they can benefit from studying this logs. From another side accountability implementation can make evidence to questions such as:

- Has the cloud service provider applied the correct measures to protect privacy?
- Has the cloud service provider covered the whole scope of privacy principles and requirements, described by privacy legislation for example EU directive 95/46/EC?

4.2.3 Trust

The downside of cloud computing, relative to SLAs, is the difficultly in determining root cause for service interruptions due to the complex nature of the environment. Reviewing the security and privacy sections and the way how the service is provided along with SLAs (see chapter 3) that defines how the service should be supplied, I have concluded that trust is a vital issue in cloud computing's business.

Trust defined in [87] as *"A legal arrangement in which an individual (the trustor) gives fiduciary control of property to a person or institution (the trustee) for the benefit of beneficiaries."*

Trust is a word that can be daily used in different contexts and at different times. It is generally understood to mean an assured reliance on the character, ability or truth of someone or something. Trust plays an important role across many disciplines, including sociology, psychology, economics, political science, history, and philosophy and computer science. From a psychological point of view, trust could be understood when a person has a faith in the trustworthiness of another person.

The definition of trust by Gambetta is often quoted in the literature [88] "*… trust (or, symmetrically, distrust) is a particular level of the subjective probability with which an agent assesses that another agent or group of agents will perform a particular action, both before he can monitor such action (or independently of his capacity of ever to be able to monitor it) and in a context in which it affects [the agent's] own action"*

Coleman in his works on the foundations of social theory [89] distinguishes four elements that define a trust situation between a trustor and a trustee:

1. Placing trust by the trustor allows the trustee to honor or abuse trust. In other words anticipating trust from trustor.
2. The trustor regrets placing trust if trust is abused, but benefits from honored trust.

3. Trust is an action that involves a voluntary transfer of resources (physical, financial, intellectual, or temporal) from the trustor to the trustee with no real commitment from the trustee.
4. There is a time lag between placement of trust and the action of the trustee.

Trust and confidence are often used interchangeably but confidence is a shape of trust or as defined in [90] confidence "is *also true of trust*". Confidence in social science is considered to be easier to measure as trust itself is viewed as a mental state and confidence reflects actions around that trust. Since trust is considered a mental state, it is hard to evaluate trust, Mui [91]proposes a more mathematical approach to solve this. Mui's approach will not be discussed in detail other than comparing how a mathematical, or systematical, approach differs from a mental one. In his work he presents that trust is connected with reputation and reciprocity.

Reputation defined by Granovetter [92] as *"a social quantity calculated based on actions by a given agent a_i and observations made by others in a_n "embedded social network" that a_i resides".*

Reciprocity in [91] as *"mutual exchange of deeds (such as favor or revenge)"*. Figure 4.2 gives an idea on how reputation, trust and reciprocity works together and creates net benefit. In other words increasing reputation for an agent a_i should also increase the trust from the other agents. Increasing trust in another agent a_j will let a_j reciprocate positively to a_i's action. Increase in a_i's reciprocating actions to other agents in a particular domain A should also increase a_i's reputation in A.

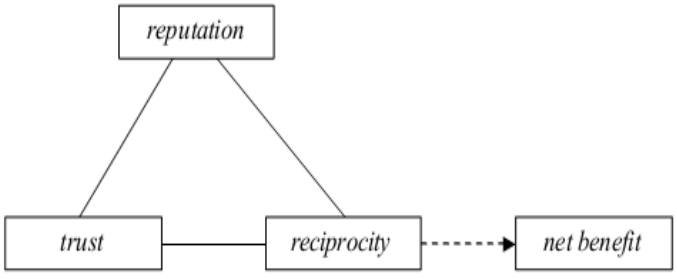

Figure 4.2: *Relationships among trust, reputation and reciprocity [91].*

The scope of this thesis is to focus on trust in computer science literature generally and in cloud computing particularly and how accountability can enhance the trust level. The reason for having introduced trust from social to psychological perspectives, because it is considered as the application of human's trust notion into the digital world [93].

In computing literature, Marsh [91] was the first person to introduce the concept of trust in distributed artificial intelligence. However, according to Mui [91] exhibits several limitations:

- The absence of reputation's model in his work
- Trust is represented in his model as a subjective real number between the arbitrary range -1 and +1. But, as mentioned before trust is a mental state.
- The model exhibits problems at extreme values and at 0
- Difficulties with the concept of "negative" trust and its propagation

According to Chang et al. [94] trust can be found in two distinct environments

- The physical trust environment: in which the trust is often established between two parties who know each other through personal interaction. In this environment trust can be established through face-to-face interactions or by introductions or recommendations.
- The virtual trust environment: where the trust relationship can be established through virtual communication mechanism. For example when an individual buy a book from Amazon he must use his credit card. In this environment, the communicating parties are referred to as agent.

Trust is important since it forms the basis upon which personal and commercial transactions take place. Commerce is based upon legal enforcement of agreements which can be very fast and effective. However, in e-commerce it is not always easy to identify individuals and because transactions may take place across geographical boundaries. An alternative approach is that adopted by eBay where each buyer and seller has a feedback rating. This is an example of a trust metric where participants in a transaction rate each other and these ratings are publicly visible. If an eBay seller consistently behaves in a trustworthy manner their rating increases, conversely if they do not it decreases. People can chose whether or

not to transact with another individual based on this rating. Trust between buyers and sellers can be inferred from the reputation that agents have in the system.

Trust is emerging as an important facet of relationships in e-business (see appendix 2). Whether it is for use in security, determining privacy or recommender systems, the definition of trust will help to improve the confidence level.

Cloud computing has opened up modes of interaction and dynamic organizational configurations that were previously inconceivable within wide array of human and business activities.

In the cloud no one knows who you are, a self-created user identity is no longer adequate. In addition from user's perspective, cloud computing means loss of physical control. In such wide distributed, open and complex environments, fraudulent or incomplete practice could occur where the cloud service provider or customer does not behave in a manner that is mutually agreed upon in the SLAs.

In a survey [95] conducted by Fujitsu Research Institute on potential cloud customers, it was found that 88% are worried about who has access to their data, 84% worry about where their data is stored and 91% want a system which enables them to control how there data is used.

For both academia and industry trust in cloud computing was an important research topic with focus on security as key concern. Security from the cloud actors' perspective is imperative in order to protect the network, the resources in order to improve the robustness and reliability of those resources. In his work [96] Firdhous et al. have identifying and categorizing 13 recent trust developments. In his summary comparison we can concludes the following:

- Over 60% of the models presented do not offer data security.
- Over 75% of the models analyzed do not support SLA.
- Over 75% of the models are not implemented or are tested using simulation.

While many obstacles to trust for privacy and security can be addressed through preventive controls (e.g. encryption, access control based on ID profiling, etc.) there are not enough,

particularly when for example an attack using impersonation or social engineering was performed to obfuscate detection it will be hard if not impossible to assign responsibility. While preventive controls are used to mitigate the occurrence of an action from taking place at all, detective controls (like accountability in our case) are used to identify the occurrence of privacy or security risk that goes against the privacy or security policies.

Accountability has been viewed as a property of trustworthy computer systems. In [97] accountability has been considered as *"The second basic control objective addresses one of the fundamental principles of security, i.e., individual accountability. Individual accountability is the key to securing and controlling any system that processes information on behalf of individuals or groups of individuals"*.

Recently an acknowledgement for the need of accountability in the cloud computing environment has steadily grown. As shown above, this requirement has been further accelerated by high distrust and the limits within preventive controls. For instance NIST [16] in his report [46] recommend the following "*Maintain accountability over the privacy and security of data and applications implemented and deployed in public cloud computing environments.*"

Neil et al. [98] recommend the following:" *Accountability – Improvement of rules enabling cloud users (especially consumers) to exercise their rights as well as improvement of models of Service Level Agreements (SLAs) as the principle vehicle to provide accountability in meeting security, privacy and trust obligations"*.

As a conclusion trust in cloud computing might be regarded as a consequence of progress towards security or privacy. But as discussed above a jump to a new level of trust is highly urgent towards full business adoption of the approach and requires a strong accountability implementation to overcome the deficiency of preventive controls.

[16] NIST = National Institute of Standards and Technology

4.3 Threats against accountability

Accountability in cloud computing depends on a strong identities. The fact that cloud computing is a different Internet-based services paradigm does not make it immune from any others well known distributed system threats against accountability. In [65] ENISA has identified some threats against identity management or against the integrity of the collection, storage and transmission of accountability. These threats can be replicated in the cloud computing environment:

- Sybil attacks: a customer can use multiple identities and access the service from many different points which make the task of accountability difficult if not impossible because there is no identified person to blame.
- Whitewashing: a customer changes identity to escape accountability, e.g. using stolen credit card.
- Attacks against authentication (e.g. impersonation): an attacker breaks the authentication system and acts as another individual's identity.
- Falsifying record collection: an attacker omits, falsifies or fabricates records.
- Tampering with records: an attacker tampers with accountability records during storage or transmission.
- Destroying or suppressing the transmission of records: an attacker destroys accountability records, or prevents the records from being transmitted to an authorized auditor.

4.4 Accountability in Distributed System

In recent years the adoption of distributed computing paradigms in on demand fashion such as grid computing and utility computing has been widely embraced by companies, governments and others organizations and hence a need for accountability as an additional mechanism towards trustworthy system has been widely discussed [99] [100] [101] and implemented [102] [103] [104]. Given that cloud computing is considered as large-scale distributed system this section will provide an overview of some accountability's solutions and why cannot be adopted in the cloud environment.

4.4.1 Accountability in network services

As mentioned above the traditional security mechanisms are not sufficient to guarantee a trustworthy system. The need for accountability is becoming more evident as services grow more complex and inter-dependent.

In [99] a framework was proposed by Yumerefendi et al. with the goal to promote accountability for dependable networked services. The main idea of this framework is preserve digitally signed records of actions and/or internal state snapshots of each service, and use to detect tampering, verify the consistency of actions and behavior, and prove responsibility for unexpected states or actions. They provide 3 fundamental properties of accountability in terms of service provision:

- *Undeniable:* the actions of an accountable actor are provable and non-repudiable.
- *Certifiable:* a client, peer or external auditor can verify that an accountable service is behaving correctly and prove any deviations from this correct behavior to a third party.
- *Tamper-evident:* any attempt to corrupt a service state should carry a high probability of detection.

Yumerefendi et al. argue that *"system builders should view accountability as a first-class design goal of services and federated distributed systems..."*. In examining accountability as a general design goal, they conclude that *"a key limitation of accountable design is that it is not fully general and that it must be 'designed in' to application structure and protocols"*. The purpose of this framework is to address the limits of approaches like Byzantine Fault Tolerance BFT *(see appendix 2)* which may be vulnerable to collusion of faulty nodes and to provide accountability for a general class of services that access and updates internal state in response to clients or peers requests.

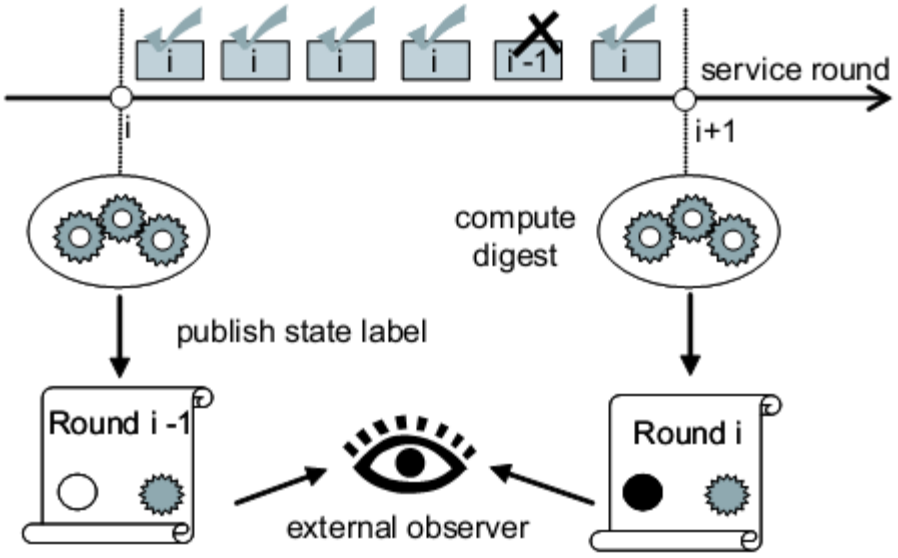

Figure 4.3: *Round Processing.*

The service processes sequences of requests in rounds. All accepted requests bear a timestamp matching the round. At the end of the round the service computes a state digest and publishes it externally. The process is shown below in Figure 4.3.

They make the following assumptions:

- The state consists of an indexed set of named, typed, data objects.
- Execution occurs in a sequence of numbered rounds.
- In each round the service updates one internal state variable and retrieval operations execute from the values at the start of the current round.
- At the end of each round the service publishes a signed, time stamped non-repudiable digest of its internal state and provides this to an external observer.

When a client requires an execution from a service, it digitally signs and time-stamped the request. After verifying that the request is valid the service executes the request and on completion returns a result with cryptographic evidence certifying its correctness relative to the published round requests. It should be noted that this approach requires access to the internal state of the services in question and that round digests are visible to all participants.

According to Yumerefendi et al. this approach is tamper-evident since the digests are signed and time-stamped.

4.4.2 PeerReview

PeerReview [103] is a system proposed by Haeberlen et al. in order to provide a practical accountability in distributed systems. PeerReview is a more generic system, designed for deployment in the Internet which can be applied to different types of distributed systems such as peer-to-peer email system and an overlay multicast system. Haeberlen et al. recognize that accountability by itself is not sufficient and must be combined with others techniques:

- Deterring faults: accountability can reduce the incidence of certain faults.
- Detection in fault tolerant systems: accountability can complement the Byzantine fault tolerant *(see appendix 2)* techniques.
- Detection in best-effort systems: accountability enables recovery, by identifying additional faults.
- Assigning blame: accountability provides non-repudiable evidence of all actions in the system which helps assigning responsibility when something goes wrong.

PeerReview is a message based system which has some limitations that can be divided in inherited Internet's limitations and design implementation of any detection system. The Internet's limitations are:

- Internet issues such as packet loss, network or processing delay can make a correct node suspected by PeerReview until acknowledging the message.
- Faulty node are detected after is causally affected by the fault, because PeerReview can detect only faults that manifests themselves through these messages.

The limitation of design implementation:

- Can only detect faults that directly or indirectly affect a message.

- Verify whether a node correctly reports its external inputs.
- As an asynchronous system, it can be difficult to distinguish omission faults from messages.

PeerReview is modeled as follows:

Each node *i* is modeled as a state machine S_i, a detector model D_i which implements the PeerReview and an application A_i. The detector module sends messages to its local application indicating the state of the inspected nodes. The states are *exposed* if *i* has obtained proof of *j*'s misbehavior, *suspected* if *j* does not send a message that it is supposed to send and trusted if the *j* respond correctly.

Figure 4.3 gives an idea on how the system is modeled and the information flow between application state machine and detector module on node *i*. Haeberlen et al. starts from the assumption that there is trusted entity that can reliably coordinate the communications between all nodes which is something lack in the cloud computing especially in scenario where cloud customer is based on different providers in order to meet cloud computing goals such as reducing costs and high performance computing. For example customer can use Microsoft Azure for the application logic and Amazon S3 for storage which may raise again the need for federated cloud computing. The need for federation in cloud computing environment was discussed in section 3.3.

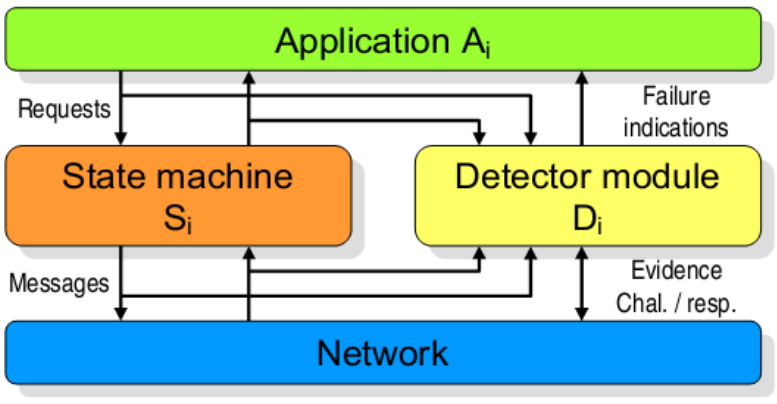

Figure 4.4: *PeerReview System Model*

PeerReview was refined to make it practical:

- Each node only keeps a full copy of its own log and retrieves other logs when necessary. For instance PeerReview during evaluation has shown that log per node per month may range from 10GB to 100 GB depending on the application. This storage cost may be increased due to the complexity and the size of cloud computing which may not be sustainable for cloud service provider or customer provider due to the additional costs.
- Tamper-evident logs and commitment protocol ensures security and consistency for node's log. By using logs and authenticators combined with the hash security mechanism, PeerReview can keep a secure record of all inputs and outputs of each node. The commitment protocol ensures that a node cannot add an entry to its log for a message it has not received. But, in cloud computing this may introduce some latency given that every record of the log must be hashed and concatenated to a previous record hash and transmitted to others nodes when requested. For example for exchanging messages each node must have public/private keypair bound to a unique node identifier, in the evaluation PeerReview with RSA 1024 *(see appendix 2)* introduce 1.5ms delay per RPC[17] and may cause CPU overhead due to the cryptographic operations for each message.
- Each node is associated with a small set of other nodes who acts as its witnesses. Periodically each witness's node compares the node's log entries with his authenticators. An authenticator is a signed statement by the node in question of its log entry. Through consistency protocol PeerReview ensures that each node maintains a single and linear log.
- Through the audit protocol PeerReview periodically replay all inputs of the node in question using a reference implementation of the node software to check logs for faulty behavior. The audit protocol in cloud computing environment add transfer

[17] RPC = Remote Procedure Call

and replay overhead. For example PeerReview drops to approximately one-third the throughput.

- PeerReview uses a challenge/response protocol to ensure that if a node I fails to respond to a challenge or does not acknowledge a message, it is eventually suspected by at least one correct witness. In addition PeerReview uses the evidence transfer protocol to make sure that all correct nodes eventually collect the same evidence against faulty nodes.

Even though PeerReview provides accountability for general distributed systems it cannot be applied for the cloud environment for these reasons:

- PeerReview must be closely integrated with the application, which requires source code modifications and a detailed understanding of the application logic.
- In a service model like IaaS, it would be impractical to apply PeerReview to an entire VM image with dozens of application and without access to the source code of each.
- As stated before PeerReview add unsustainable overheads such as time latency, CPU Overhead, and a large amount of storage space to store logs.

4.4.3 Accountable Virtual Machines

The accountable virtual machines (AVMs) was proposed by Aditya et al. [104] in order to provide a generic approach for accountability where different hosts and organizations do not necessarily trust each other or where software is hosted on third-party operated platforms. The goal is to provide detection when there is a faulty node and to obtain evidence that would convince a third party when a dispute is verified. AVMs offer many of the capabilities discussed in the previous solution such as the tamper-evident log, the commitment protocol, the evidence protocol and the audit protocol. What's different from the previous solution (PeerReview) is that AVMs can provide this capability for any black-box binary image that can run inside a virtual machine. However, AVM solely does not fulfill all the requirements to deal with an arbitrary binary executable such as recording the relevant nondeterministic

events (for example hardware interrupts). To overcome this limit Aditya et al. have constructed an Accountable Virtual Machine Monitor (AVMM) which implements the AVMs. This approach can convince a third party without having to trust either the provider or the customer.

AVMs can be used in a competitive system such as auction, peer-to-peer or cloud computing where failures can be caused both by bugs in the customer's software and by faults or misconfiguration of the provider's platform. Both customer and provider can use the AVM log to produce evidence and to prove innocence. AVMs as the previous solution affect various metrics such as latency, traffic or CPU utilization. They recognize some limitations:

- AVMs cannot detect bugs or vulnerabilities in the software. Because, the detection depends upon the behavior in the reference implementation of the machine M_R and the software. For example if a bug is exercised during an execution the audit will succeed.
- Any behavior that can be achieved by providing appropriate inputs to M_R is considered correct such recording local network inputs in the log.

For instance the HP researchers in [105] argue that *"The scenario of this non-cloud based game was not a practical business scenario for accountability, and did not address the needs of logging virtual-to-physical mapping."*

Aditya et al. recognize that even if AVM are a perfect match for IaaS, it still faces additional challenges:

- Auditors cannot easily replay the entire execution for lack of resources ;
- Accountable services must be able to interact with non-accountable clients ;
- Signing every single packet it may not be practical.

In addition if AVM is adopted in the cloud it will not resolve the privacy issue because AVM records enough information to replay the execution of the software it is running which may expose some sensitive information to the auditor.

4.5 Cloud computing Accountability

Because cloud computing is a large-scale distributed system, recently accountability was proposed [8] [64] [105] as an alternative approach to overcome the limits of the preventive controls and enhance the trust level. To my knowledge, no practical solution at the moment of writing this thesis was approved to be widely adopted in the cloud computing environment in order to enforce accountability in the cloud computing environment.

HP researchers in [105]argue that *"The complexity resulting from the sheer amount of virtualization and data distribution carried out in current clouds has also revealed an urgent need for research in cloud accountability, as has the shift in focus of customer concerns from server health and utilization to the integrity and safety of end-users' data"*.

They provide a framework called Trust Cloud which addresses accountability in cloud computing via technical and policy-based approaches. According to HP researchers, accountability is the third component of trust in cloud computing after security and privacy. They also highlight the importance of auditability which is considered as the retrospective of accountability.

4.5.1 TrustCloud Framework

The TrustCloud framework is a conceptual model that potentially can be used to give cloud users a single point of view for accountability of the CSP. They examined accountability in the cloud from all aspects, starting from analyzing the complexities introduced in cloud computing to the definition of Cloud Accountability Lifecycle.

The main complexities introduced:

- Challenges introduced by Virtualization such as tracking of virtual-to-physical or physical-to-virtual mapping and the problem of using multiple operating system environments.

- The importance of logging from the file-centric perspective *"By the file-centric perspective, we mean that we need to trace data and files from the time they are created to the time they are destroyed"*.
- Managing the exponential size in log size while taking into account the private and sensitive information of cloud customers.
- The need for a real-time accountability that capture key suspected events instantaneously.

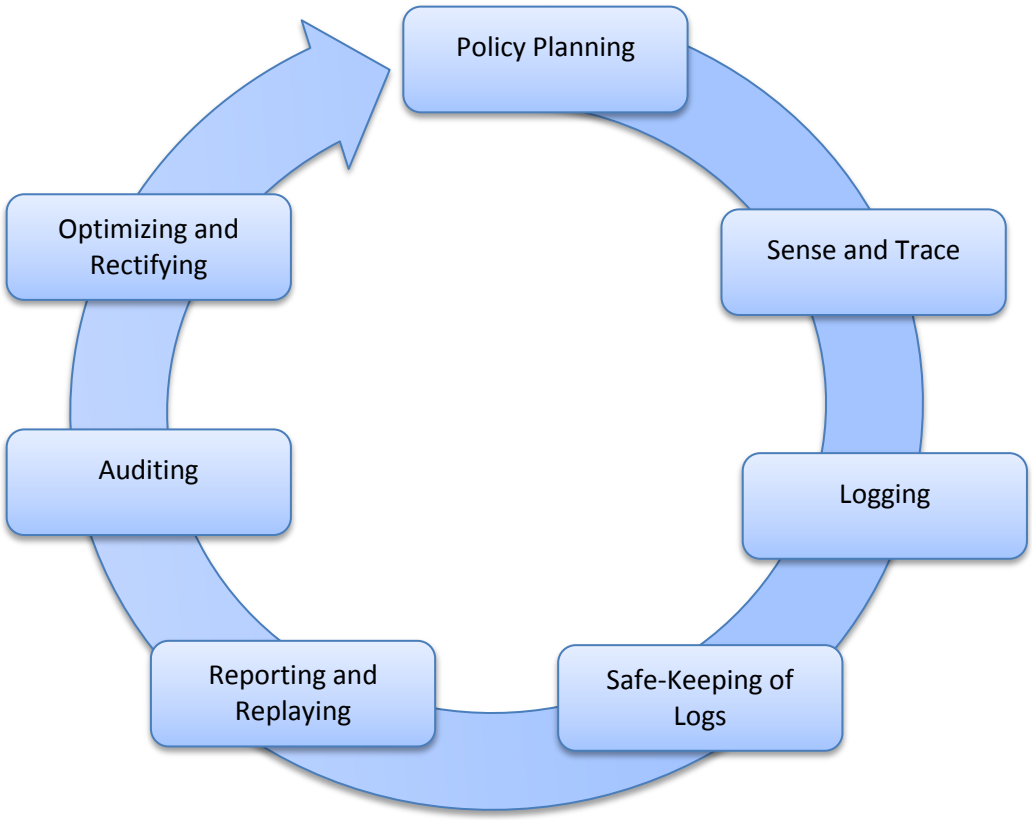

Figure 4.5: *The Cloud Accountability Life Cycle*

The lifecycle consist of the 7 phases of cloud accountability that include (see Figure 4.5):

1. Policy planning: CSPs have to decide what information and which events to log.
2. Sense and trace: trigger logging and performing a total track whenever an expected phenomenon occurs.
3. Logging: performing a file-centric logging.

4. Safe-keeping of logs: applying the necessary security mechanisms to ensure that the logs are tamper-free.
5. Reporting and replaying: generating summaries from logs file-centric and suspected irregularities must be signaled to the customer.
6. Auditing: Logs and reports are checked and potential irregularities highlighted
7. Optimizing and rectifying. Any vulnerabilities or security issue must be fixed and governance of the cloud processes are improved.

Then they examine the five layers of cloud accountability and recommend the technical and policy-based approaches for each layer that will help to achieve a trusted cloud. Figure 4.6 shows the abstraction layers for the type of logs needed for an accountable cloud.

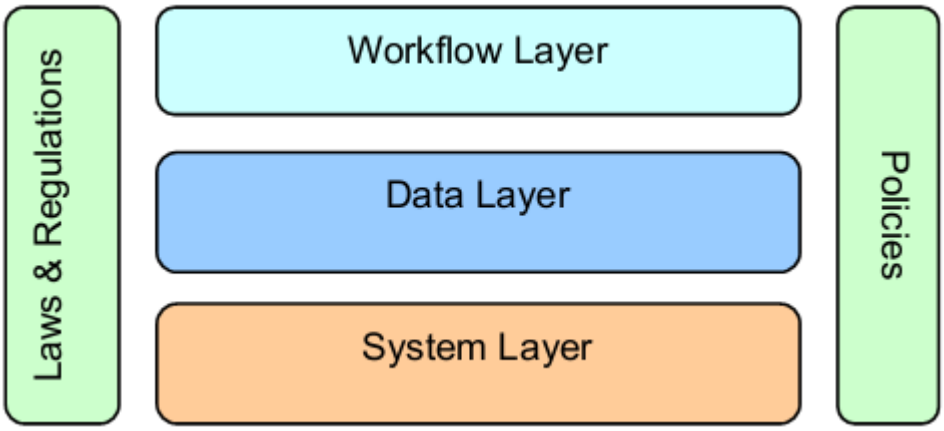

Figure 4.6*: Abstraction Layers of accountability in Cloud computing*

These five layers include:

1. *System Layer:* the lowest layer of the framework and performs file-centric logging for operating systems, file systems and cloud's internal network.
2. *Data Layer:* responsible for tracking the provenance of data and maintain a consistency logs which improve mechanisms such as rollback, recovery, replay, backup and restoring of data.

3. *Workflow Layer:* audit and control the high level fraudulent risks such as procurement approval routes, decision making flows and role management in software services run within the cloud.
4. *Policies, Laws & Regulations: "Policies and laws require information to be logged on what data items are processed, accessed, stored or transmitted. They may also require information on why, when, where, how and by whom this processing takes place."*

4.5.2 Accountability service for the cloud

Chen et al. [64] propose that accountability can be offered as a service. According to Chen et al. accountability can be incorporated into activity based process by the actor (conductor) of the process to log activities in a separate domain from the domain of its own. Figure 4.7 shows an example of such incorporation.

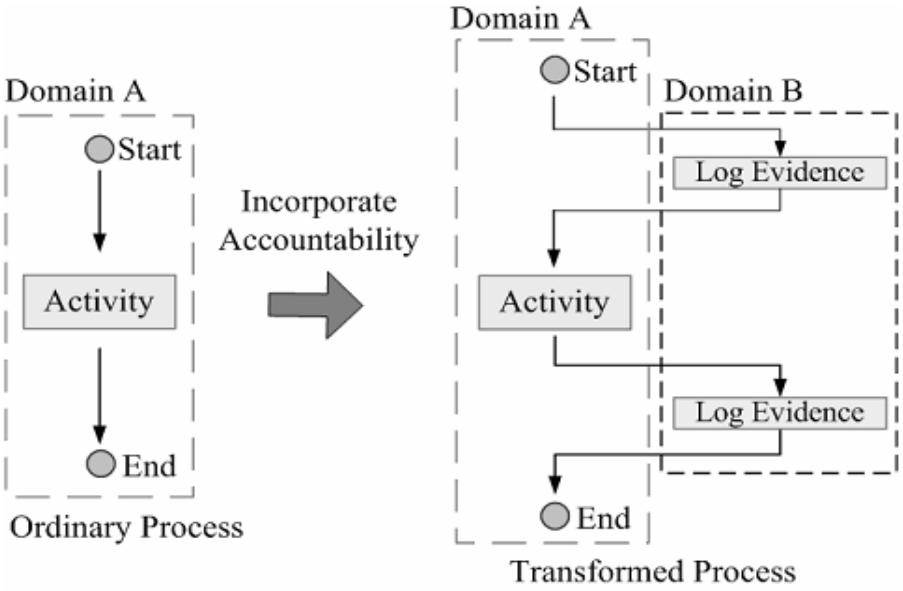

Figure 4.7: *Example of incorporating accountability into process*

They start from the fact that cloud computing shares many properties of SOA and the adoption of the two (cloud & SOA) can offer a serviceably cloud computing environment Through the use of the Public Key Infrastructure technique they can ensure the evidence preserved logs are tamper-evident in addition to its origin. Chen et al. argue that *"SLA*

definitions need to contain the admitted obligations from service provider, the methods to evaluate the compliance to those obligations, and the compensation rules for violations".

Chen et al. recognize in a cloud environment where services are dynamically allocated and released at will implementing the solution using the normal approach *(such the one implemented in PeerReview) where* every node must witness for the correctness of another node's log can be difficult task and they propose to use a central approach. The solution requires two domains which are the *Accountability Service Domain (ASD)* and the *Business Service Domain (BSD)*. Figure 4.8 shows the system model of the *Trustworthy Service Oriented System (TSOA)* where each entity of the BSD must communicate with the ASD in order to logs events.

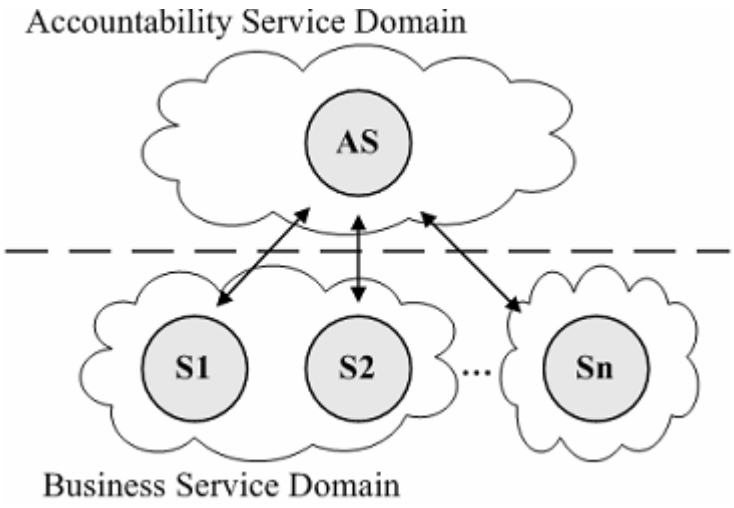

Figure 4.8: *Overall system design*

The accountable include three core functionalities:

- *Logging:* through the modeling of business activities into several basic activity types using BPEL (Business Process Execution Language) *(see appendix 2),* accountability can be incorporated by inserting logging operations into the original process sequence before or/and after activities. BPEL defines the correct interactions between services. After the BPEL transformation, the logger needs to register with AS to start the logging. Certain documents should be submitted in order to give the AS sufficient knowledge about the logger such as transformed BPELs, Associated

WSDLs *(see appendix 2)* and SLAs in order to guarantee certain QoS and meet the requirements of the SLA.

- *System Monitoring and Auditing:* the monitoring and auditing logic is generated from the service's Business Logic (BPEL) and SLA registered. By continuously monitoring and analyzing the records, the functionality determines if the operations performed or proposed by services meet the SLA requirements.
- *Fault Resolution:* when an exception is linked to the guilty service, actions such as sending warning or dismissing the violating service node are taken by AS.

This solution is not limits free:

- The service implements a centric approach which may be exposed to well-known issues in the sense that when the entity that represents the ASD crashes there is no way to provide evidence of any violation.
- Some types of violation or errors can be uncovered only after long time.
- Averagely a 30% increase in the overall process latency after BPEL transformation.
- They use a SOA approach while SOC covers many aspects of cloud environment *(see section 2.3.4)*
- Must be closely integrated with the application, which requires source code modifications and a detailed understanding of the application logic

Chapter 5

Conclusion

Cloud computing provides a powerful and flexible paradigm towards an economies of scale. However, this new paradigm is still in its infancy, thereby many challenges need to be addressed.

This thesis has provided an examination of accountability in cloud computing starting from studying related concepts to the examination of previous proposed works.

The first part of this thesis has presented SLA which is legal mean that must guarantee a certain level of QoS and address even partially the problem of mistrust. However, the two types of SLA's, negotiable and non-negotiable *(see section 3.2)* may include some risks. For example there is a risk that the terms of service will not meet the needs of the organization if the SLA was signed without obtaining adequate legal and technical advice. It is essential that SLAs are initially well-defined before any information is provided or shared, because it is very hard to negotiate them later.

A further work on federated cloud computing can guarantee a certain level of service as can help the SLA management and transform the business requirements to specific internal provisioning manifestations.

The second parts of this thesis examine accountability as an urgently needed mechanism that can help by leveraging the level of trust between the customer and the provider.

The outsourced nature of the cloud means the loss of immediate physical control and the delegation of some tasks to the CSP without any possibilities to control if SLA was honored. The downside of cloud computing, relative to SLAs, is the difficultly in determining root cause for service interruptions due to the complex nature of the environment. The most important barriers that are holding back the adoption of the cloud are security and privacy concerns. While many privacy and security issues can be addressed through preventive

controls they are not enough to overcome the problem of trust in the cloud environment. Even if concepts and previous work on accountability in distributed systems are efficient and practical they cannot be implemented in cloud environment.

Research on accountability for cloud computing environment is still in its infancy. There is some initial proposed work to make cloud accountable, but further progress is needed regarding the efficiency of accountability mechanisms and on managing the tension between accountability and privacy. For example reducing the cost of maintaining tamper-evident logs and reducing the cost of automatic fault detection.

Until now neither self-regulation nor legislation is likely to offer a complete response to the privacy and security issues raised by cloud computing. The conflicts between accountability and privacy must be managed carefully. In general, the system should expose information only to legitimate auditors, and only the information strictly necessary for the auditor to verify the properties of interest. For example techniques based on state-machine replay may expose much more information about the execution than necessary. Accountability if implemented efficiently can be a powerful tool for motivating better practices and consequently more reliable and trustworthy cloud computing systems. In order to implement accountability we must keep in mind that there's no "one size fit all" and each cloud's service model must have his accountability's solution.

Future Work

There are numerous possibilities to continue the work of this thesis. It is required a work on how accountability can be integrated in the SLA. Further research is needed to investigate how can be possible to implement accountability in highly scalable environment whilst maintaining high performance and flexibility of the cloud. In addition, it is required to investigate on how the output of accountability data can be used in the development of reputation and trustworthiness of cloud services.

Bibliography

[1] Zhang *et al.*, "Cloud computing: state-of-the-art and research challenges," *Journal of Internet Services and Applications.* vol. 1, no. 1, pp. 7-18, 2010.

[2] Mell *et al.*, "The NIST Definition of Cloud Computing," 2011. *(Draft)–Recommendations of the National Institute of Standards and Technology*. Special publication 800-145 (draft), Gaithersburg (MD) (2011)

[3] Vaquero *et al.*, "A break in the clouds: towards a cloud definition," *ACM SIGCOMM Computer Communication Review,* vol. 39, no. number 1, pp. 50-55, january 2009.

[4] Lodi *et al.*, "SLA-Driven Clustering of QoS-Aware Application Servers," *in IEEE Transactions on Software Engineering*, vol. 33, no. 3, pp. 186 - 197, 2007.

[5] "IDC Cloud Research". Available: http://www.idc.com/prodserv/idc_cloud.jsp.

[6] "Definition cloud computing". Available: http://searchcloudcomputing.techtarget.com/definition/cloud-computing.

[7] "Amazon cloud outage derails Reddit, Quora,". Available: http://news.cnet.com/8301-30685_3-20056029-264.html.

[8] A. Haeberlen, "A case for the accountable cloud*," ACM SIGOPS Operating Systems Review*, vol. 44, no. 2, 2010, pp. 52-57.

[9] "Wikipedia, Utility Computing". Available: http://en.wikipedia.org/wiki/Utility_computing.

[10] L. Kleinrock., "UCLA to be first station in nationwide computer network.," UCLA, Office of Public Information, 1969,pp. 5.

[11] "Google App Engine,". Available: http://code.google.com/appengine.

[12] "Gartner Highlights Five Attributes of Cloud Computing,". Available: http://www.gartner.com/it/page.jsp?id=1035013.

[13] "vertical scalability,". Available: http://searchcio.techtarget.com/definition/vertical-scalability.

[14] "horizontal scalability,". Available: http://searchcio.techtarget.com/definition/horizontal-scalability.

[15] Jefery *et al.*, "The future of cloud computing," *Expert Group Report*, 2010.

[16] Harms *et al.*, "The economics of the cloud,"Microsoft whitepaper, *Microsoft Corporation,* November 2010.

[17] "Google App Engine,". Available: http://code.google.com/intl/it-IT/appengine/.

[18] "Windows Azure,". Available: www.microsoft.com/azure.

[19] "Force.com,". Available: http://www.salesforce.com/platform.

[20] "Salesforce CRM,". Available: http://www.salesforce.com/platform.

[21] "SAP Business ByDesign,". Available: www.sap.com/sme/solutions/ businessmanagement/businessbydesign/index.epx.

[22] Hosseini *et al.*, "Research Agenda in Cloud Technologies," *ArXiv preprint, submitted to 1st ACM Symposium on Cloud Computing* 2008.

[23] "Capex vs. Opex: Most People Miss the Point About Cloud Economics,". Available: http://www.cio.com/article/484429/Capex_vs._Opex_Most_People_Miss_the_Point_About_Cloud_Economics.

[24] "Oracle's Ellison nails cloud computing,". Available: http://news.cnet.com/8301-13953_3-10052188-80.html.

[25] "Gartner Says Contrasting Views on Cloud Computing Are Creating Confusion,". Available: https://www.gartner.com/it/page.jsp?id=766215.

[26] "What is virtualization ? Definition from whatis.com,". Available: http://searchservervirtualization.techtarget.com/definition/virtualization.

[27] Keller *et al*., "NoHype: Virtualized Cloud Infrastructure without the virtualization" *In ACM/IEEE International Symposium on Computer Architecture (ISCA),*ACM,2010.

[28] "Five myths of cloud computing," *Hewlett-Packard Development Company*, 2011.

[29] I. Foster, "What is the Grid? A Three Point Checklist," *Argonne National Laboratory & University of Chicago,* 2002.

[30] "Cloud computing versus grid computing,". Available: http://www.ibm.com/developerworks/web/library/wa-cloudgrid/.

[31] Foster *et al.*, "Cloud Computing and Grid Computing 360-Degree Compared," *Grid Computing Environments Workshop, GCE'08.*, Chicago, 2008.

[32] "Google Trends:Cloud Computing, Grid Computing,". Available: http://www.google.com/trends/?q=grid+computing,cloud+computing.

[33] "Cloud computing,". Available: http://en.wikipedia.org/wiki/Cloud_computing.

[34] Tsai *et al.*, "Introduction to Service-Oriented Computing," *In IFIP International Federation for Information Processing,* 2004, pp. 294–310.

[35] "A Framework for QoS-based Routing in the Internet,". Available: http://www.ietf.org/rfc/rfc2386.txt.

[36] V. B. Iversen, "Teletraffic Engeneering and Network Planning*," Technical University of Denmark*, 2010.

[37] W. C. Hardy, "QoS Measurement and Evaluation of Telecommunications Quality of Service", *JOHN WILEY & SONS, LTD,* 2001, pp. 5,6.

[38] Buco *et al,* "Utility computing SLA management based upon business objectives," *IBM SYSTEMS JOURNAL*, 2004 Vol. 43 no. 1,pp.159-178.

[39] "Service level agreement,". Available: http://en.wikipedia.org/wiki/Service-level_agreement.

[40] Goertzel *et al.*, "Cloud Computing Security Government Considerations for the Cloud Computing Environment," 2009.

[41] Ambros *et al.*, "Cloud Computing Security Risks SLA and Trust," 2010.

[42] Nathuji *et al*., "Q-Clouds: Managing Performance Interference Effects for QoS-Aware Clouds," 2010, *In EuroSys '10: Proceedings of the 5th European conference on Computer systems, New York, NY, USA*, pp.237-250.

[43] "Directive 95/46/EC of the European Parliament and of the Council of 24 October 1995 on the protection of individuals with regard to the processing of personal data and on the free movement of such data,". Available: http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=CELEX:31995L0046:en:HTML.

[44] "The Service Level Agreement,". Available: http://www.sla-zone.co.uk/.

[45] "Amazon EC2 Service Level Agreement,". Available: http://aws.amazon.com/ec2-sla/.

[46] Jansen *et al.*, "Guidelines on Security and Privacy in Public Cloud Computing," *NIST*, 2011.

[47] L. Schubert, " The Future of Cloud Computing," *Expert Group Report*, 2010.

[48] Buyya *et al.*, "CLoud Computing Principles and Paradigms", *Wiley*, 2010, pp. 425-429.

[49] E. Wustenhoff, "Service Level Agreement in the Data Center," *Sun BluePrints™,* 2002.

[50] Rochwerger *et al.*, "The RESERVOIR Model and Architecture for Open Federated Cloud Computing," 2008.

[51] Cloud Computing - A Primer. Available: http://www.cisco.com/web/about/ac123/ac147/archived_issues/ipj_12-4/124_cloud2.html.

[52] Goiri *et al.*, "Characterizing Cloud Federation for Enhancing Providers' Profit," *In 3rd International Conference on Cloud Computing, Miami*, Florida, USA, 2010.

[53] "RESERVOIR & SLA@SOI,". Available: http://62.149.240.97/index.php?page=reservoir-sla-soi.

[54] "Cloud Computing Platforms, Software and Applications" 2010.

[55] "Resources and services virtualisation without barriers (RESERVOIR),". Available: http://cordis.europa.eu/fetch?CALLER=FP7_PROJ_EN&ACTION=D&DOC=42&CAT=PROJ&QUERY=012047141165:8ec2:7ad7f9ef&RCN=85304.

[56] Theilmann *et al.*, "SLA@SOI Final Report," *European Community*, 2011.

[57] "SLAs Empowering a Dependable Service Economy," *European Commision*, 2008.

[58] "Open Cloud Computing Interface,". Available: http://en.wikipedia.org/wiki/Open_Cloud_Computing_Interface.

[59] "D.B6a Use Case Specification eGovernment – M17,". Available: http://sla-at-

soi.eu/results/deliverables/d-b6a-m12-use-case-specification-egovernment-m17/.

[60] Metsch *et al.*, "Using Cloud Standards for Interoperability of Cloud Frameworks," 2011.

[61] "Accountability". Available: http://en.wikipedia.org/accountability.

[62] J. G. Koppell, "Pathologies of Accountability: ICANN and the Challenge of Multiple Accountabilities Disorder," 2005.

[63] "Accountability". Available: http://www.merriam-webster.com/dictionary/accountability.

[64] Chen *et al.*, "Accountability Service for the cloud", *Proc. 6th IEEE World Congress on Services (SERVICES-1*) , 2010, pp. 91-98.

[65] Lkonomou *et al.*, "Privacy, Accountability and Trust -Challenges and Opportunities," *ENISA*, 2011.

[66] "what is fault tolerance,". Available: http://www.webopedia.com/TERM/F/fault_tolerance.html.

[67] "What is responsibility? definition and meaning". Available: http://www.businessdictionary.com/definition/responsibility.html.

[68] "Responsibility vs. Accountability,". Available: http://www.ozprinciple.org/blog/uncategorized/responsibility-vs-accountability.

[69] "AWS Customer Agreement,". Available: http://aws.amazon.com/agreement/.

[70] H. Nissenbaum, "Accountability in a Computerized Society", Science and Engineering Ethics,1994, pp. 25-42.

[71] "Cloud Computing Primer for IT Pros,". Available: http://blogs.technet.com/b/yungchou/archive/2010/11/15/cloud-computing-primer-for-it-pros.aspx.

[72] "Security Guidance for Critical Areas of Focus in Cloud Computing V 3.0," Cloud Security Alliance, 2011.

[73] "Top Threats to Cloud Computing V1.0," *CSA,* 2010.

[74] "Zeus bot found using Amazon's EC2 as C&C server,". Available:

http://www.theregister.co.uk/2009/12/09/amazon_ec2_bot_control_channel/.

[75] "Microsoft Cloud Data Breach Heralds Things to Come,". Available: http://www.pcworld.com/businesscenter/article/214775/microsoft_cloud_data_breach_heralds_things_to_come.html.

[76] Ristenpart *et al.*, "Hey, You, Get Off of My Cloud:Exploring Information Leakage in Third-Party Compute Clouds," *Copyright 2009 ACM*, 2009.

[77] "Security Standoff in the Cloud-CA technologies,". Available: http://www.ca.com/us/news/Press-Releases/na/2011/Security-Standoff-in-the-Cloud.aspx.

[78] Lombardi *el al.*, "Secure virtualization for cloud computing*" In Elsevier: Journal of Network and Computer Applications*, 2010.

[79] "Compliance And Cloud - Responsible Or Accountable,". Available: http://blogs.csoonline.com/1748/compliance_and_cloud_responsible_or_accountable.

[80] "What's 'Privacy'?". Available: http://www.rogerclarke.com/DV/Privacy.html.

[81] "EUR-Lex-31995L0046 en,". Available: http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=CELEX:31995L0046:en:HTML.

[82] Mather *et al.*, Cloud Security and Privacy, *O'Reilly,* 2009,pp. 145-171.

[83] "Cloud Computing and Privacy,". Available: http://www.oncloudcomputing.com/en/tag/privacy-breaches/.

[84] "Google SAML Single Sign on vulnerability,". Available: http://www.kb.cert.org/vuls/id/612636.

[85] "The Dropbox Blog Yesterday's Authentication Bug,". Available: http://blog.dropbox.com/?p=821.

[86] Ko *et al.*, "Towards Achieving Accountability, Auditability and Trust in Cloud Computing," *Cloud and Security Lab*, HP Labs, 2011.

[87] "what is trust? definition and meaning,". Available: http://www.investorwords.com/5084/trust.html.

[88] G. Diego, "Can We Trust Trust?," *University of Oxford*, 2000, p. 217.

[89] "Foundations of social theory," *Harvard University Press*, 1994.

[90] B. D. Adams, "Trust vs. Confidence," *DRDC*, *Toronto*, 2005.

[91] L. Mui, “Computational Models of Trust and Reputation: Agents, Evolutionary Games, and Social Networks”, 2002.

[92] M. Granovetter, "Economic Action and Social Structure: The Problem of Embeddedness," *The American Journal of Sociology,* vol. Vol. 91, no. No. 3, p. 481, 1985.

[93] "Computational trust,". Available: http://en.wikipedia.org/wiki/Computational_trust#cite_note-4.

[94] Chang *et al.*, Trust and Reputation for Service-Oriented Environments, *John Wiley & Sons, Ltd,* 2006,pp.27-49.

[95] "Personal data in the cloud : A global survey of consumer attitudes," *Fujitsu Research Institute*, 2010.

[96] Firdhous *el al.*, "Trust and Trust Management in Cloud Computing -A Survey," *Universiti Utara Malaysia*, 2011.

[97] "Trusted Computer System Evaluation Criteria," *DEPARTMENT OF DEFENCE*, 1983.

[98] Robinson *et al.*, "The Cloud: Understanding the Security, Privacy and Trust Challenges," 2010.

[99] Yumerefendi *el al.*, "Trust but Verify: Accountability for Network Services," *In Proceedings of the Eleventh ACM SIGOPS European Workshop,* 2004.

[100] Nelson *et al.*, "Trust and on Demand: Enabling Privacy, Security, Transparency, And Accountability in Distributed Systems" *IBM*, 2005.

[101] Yumerefendi *et al.* "The Role of Accountability in Dependable Distributed Systems," *In Proceedings of HotDep*, Jun 2005.

[102] Ruth *et al.*, "E-notebook Middleware for Accountability and Reputation Based Trust in Distributed Data Sharing Communities," *Purdue University*.

[103] Haeberlen *et al.*, "PeerReview: Practical Accountability for Distributed Systems," *SOSP’07*,October 14–17, 2007, Stevenson, Washington, USA.

[104] Haeberlen *et al.*, "Accountable Virtual Machines," 2010.

[105] Ko *et al.*, "TrustCloud: A Framework for Accountability and Trust in Cloud," *HP Laboratories*, 2011.

[106] "Service-oriented architecture (SOA) definition,". Available: http://www.service-architecture.com/web-services/articles/service-oriented_architecture_soa_definition.html.

[107] A. Rodriguez, "RESTful Web services: The basics,". Available: http://www.ibm.com/developerworks/webservices/library/ws-restful/.

[108] "server consolidation,". Available: http://searchdatacenter.techtarget.com/definition/server-consolidation.

[109] Wikipedia, "Security Assertion Markup Language,". Available: http://en.wikipedia.org/wiki/Security_Assertion_Markup_Language.

[110] "Byzantine fault tolerance,". Available: http://en.wikipedia.org/wiki/Byzantine_fault_tolerance.

[111] "Business Process Execution Language,". Available: http://en.wikipedia.org/wiki/Business_Process_Execution_Language.

[112] "MULTITENANCY,". Available: http://en.wikipedia.org/wiki/Multitenancy.

[113] "Web Services Description Language (WSDL),". Available: http://www.w3.org/TR/wsdl.

[114] "Best effort delivery,". Available: http://en.wikipedia.org/wiki/Best_effort_delivery.

[115] "Business service management,". Available: http://en.wikipedia.org/wiki/Business_service_management.

APPENDIX 1

"Cloud Computing offers the prospect of dramatically increasing your computing power, being able to balance workloads with demand and paying only for the services you use. Here is how a few of the leaders in the Cloud Computing marketplace define Cloud." **Spinning Clouds**

"A pay-per-use model for enabling convenient, on-demand network access to a shared pool of configurable and reliable computing resources (e.g., networks, servers, storage, applications, services) that can be rapidly provisioned and released with minimal consumer management effort or service provider interaction." **NIST**

"A style of computing in which scalable and elastic, IT-enabled capabilities are provided "as a service" across the Internet to multiple external customers." **Gartner**

"An emerging IT deployment, development and delivery model enabling real time delivery of products, services and solutions over the Internet." **IDC**

"Cloud Computing refers to both the applications delivered as services over the Internet and the hardware and systems software in the Data Centers that provide those services. The services themselves have long been referred to as Software as a Service (SaaS), so we use that term.The Data Center hardware and software is what we call a Cloud." **University of California – Berkeley**

"The key characteristics of the cloud are the ability to scale and provision computing power dynamically in a cost efficient way and the ability of the consumer (end user, organization or IT staff) to make the most of that power without having to manage the underlying complexity of the technology. The cloud architecture itself can be private (hosted within an organization's firewall) or public (hosted on the Internet)." **Cloud Manifesto**

APPENDIX 2

SOA:

"A service-oriented architecture is essentially a collection of services. These services communicate with each other. The communication can involve either simple data passing or it could involve two or more services coordinating some activity. Some means of connecting services to each other is needed" [106]

OCCI:

"The Open Cloud Computing Interface (OCCI) (TM) comprises a set of open community-lead specifications delivered through the Open Grid Forum[1][2][3], which define how infrastructure service providers can deliver their compute, data, and network resource offerings through a standardized interface. OCCI has a set of implementations that act as proofs of concept. It builds upon World Wide Web fundamentals by using the proven REST (Representational State Transfer) approach for interaction and delivers an extensible model for interacting with "as-a-Service" services" [58]

REST (Representational State Transfer):

"REST defines a set of architectural principles by which you can design Web services that focus on a system's resources, including how resource states are addressed and transferred over HTTP by a wide range of clients written in different languages. If measured by the number of Web services that use it, REST has emerged in the last few years alone as a predominant Web service design model. In fact, REST has had such a large impact on the Web that it has mostly displaced SOAP- and WSDL-based interface design because it's a considerably simpler style to use" [107]

Server consolidation

" is an approach to the efficient usage of computer server resources in order to reduce the total number of servers or server locations that an organization requires. The practice developed in response to the problem of server sprawl, a situation in which multiple, under-utilized servers take up more space and consume more resources than can be justified by their workload" [108]

SAML

"***Security Assertion Markup Language (SAML)*** *is an XML-based open standard for exchanging authentication and authorization data between security domains, that is, between an identity provider (a producer of assertions) and a service provider (a consumer of assertions). SAML is a product of the OASIS Security Services Technical Committee.*

The single most important problem that SAML is trying to solve is the Web Browser Single Sign-On (SSO) problem, a problem also addressed by the OpenID protocol. Single sign-on solutions are abundant at the intranet level (using cookies, for example) but extending these solutions beyond the intranet has been problematic and has led to the proliferation of non-interoperable proprietary technologies" [109]

E-business (electronic business)

"E-business (electronic business), derived from such terms as "e-mail" and "e-commerce," is the conduct of business on the Internet, not only buying and selling but also servicing customers and collaborating with business partners"

Byzantine fault tolerance:

"Byzantine fault tolerance is a sub-field of fault tolerance research inspired by the Byzantine Generals' Problem, which is a generalized version of the Two Generals' Problem. The object of Byzantine fault tolerance is to be able to defend against Byzantine failures, in which components of a system fail in arbitrary ways" [110]

BPEL

"Business Process Execution Language (BPEL), short for Web Services Business Process Execution Language (WS-BPEL) is an OASIS standard executable language for specifying actions within business processes with web services" [111]

Multitenancy

"Multi-tenancy refers to a principle in software architecture where a single instance of the software runs on a server, serving multiple client organizations (tenants)" [112]

WSDL

"WSDL is an XML format for describing network services as a set of endpoints operating on messages containing either document-oriented or procedure-oriented information. The operations and messages are described abstractly, and then bound to a concrete network protocol and message format to define an endpoint. Related concrete endpoints are combined into abstract endpoints (services). WSDL is extensible to allow description of endpoints and their messages regardless of what message formats or network protocols are used to communicate, however, the only bindings described in this document describe how to use WSDL in conjunction with SOAP 1.1, HTTP GET/POST, and MIME" [113]

Best effort delivery

"Best effort delivery describes a network service in which the network does not provide any guarantees that data is delivered or that a user is given a guaranteed quality of service level or a certain priority" [114]

Business Service Management

"Business service management (BSM) is an approach used to manage business-aligned IT services. A BSM philosophy promotes a customer-centric and business-focused approach to Service Management, aligning business objectives and priorities with IT or ICT from strategy through to operations and continual improvement" [115]