Cloud Broker Based Trust Assessment of Cloud Service Providers

Pramod S. Pawar

Department of Computer Science, School of Mathematics, Computer Science and Engineering,

City University London

A thesis submitted for the degree of

Doctor of Philosophy (Computer Science)

June 2015

Supervisors:	Prof. Muttukrishnan Rajarajan	(City University London, UK)
	Prof. Andrea Zisman	(The Open University, UK)
	Prof. Theo Dimitrakos	(British Telecommunications, University of Kent, UK)

Contents

List of Figures

List of Tables

Acknowledgments

First and foremost, I express my deepest sense of gratitude to my supervisors, Prof. Muttukrishnan Rajarajan (City University London, UK), Prof. Andrea Zisman (The Open University, UK), and Prof. Theo Dimitrakos (British Telecommunications, University of Kent, UK), who supported me throughout my PhD with their guidance, motivation, directions, encouragement and excellent advices, to keep working towards my goal. Their extraordinary experiences, truly scientific intuition, constant oasis of ideas have inspired me and enriched my growth as student and as a researcher.

I am thankful to all my colleagues at City University London and at BT Research lab, Ipswich, for the research discussions that have contributed substantially to this work. I am thankful to Dr. Srijith K. Nair for his guidance during early stages of my research and thankful to Dr. Yogachandran Rahulamathavan for proof-reading my thesis.

I gratefully acknowledge the OPTIMIS consortium for their co-operation during the OPTIMIS project which was beneficial to a large extent for the Phd work.

I am thankful to the examiners Dr. Karim Djemame and Prof. George Spanoudakis for their insightful and valuable comments which have added much to the clarity and enhanced the scope of the manuscript.

My sincere thanks to Dr. Subramanian Neelakantan (CDAC) and Prof. Srinath Srinivasa (IIIT, Bangalore), who have been my inspiration for pursuing the PhD.

Finally I take this opportunity to express my profound gratitude to my beloved parents, parents-in-law, my wife Deepa, my daughter Shruti, my son Swayam, my brothers and sisters and all my relatives and friends, for their blessings, love, affection and moral support. Without them all success of my endeavours would have been difficult to envisage.

Declaration

I, the undersigned, declare that this work has not previously been submitted as an exercise for a degree at this or any other University, and that, unless otherwise stated, it is entirely my own work

Pramod S. Pawar

15 June 2015

Abstract

Cloud computing is emerging as the future Internet technology due to its advantages such as sharing of IT resources, unlimited scalability and flexibility and high level of automation. Along the lines of rapid growth, the cloud computing technology also brings in concerns of security, trust and privacy of the applications and data that is hosted in the cloud environment. With large number of cloud service providers available, determining the providers that can be trusted for efficient operation of the service deployed in the provider's environment is a key requirement for service consumers.

In this thesis, we provide an approach to assess the trustworthiness of the cloud service providers. We propose a trust model that considers real-time cloud transactions to model the trustworthiness of the cloud service providers. The trust model uses the unique uncertainty model used in the representation of opinion. The *Trustworthiness* of a cloud service provider is modelled using *opinion* obtained from three different computations, namely (i) *compliance of SLA (Service Level Agreement) parameters* (ii) *service provider satisfaction ratings* and (iii) *service provider behaviour.* In addition to this the trust model is extended to encompass the essential Cloud characteristics, credibility for weighing the feedbacks and filtering mechanisms to filter the dubious feedback providers. The credibility function and the early filtering mechanisms in the extended trust model are shown to assist in the reduction of impact of malicious feedback providers.

The performance of the trust model in the cloud environment is studied using different modes of the cloud broker. The novel architecture designed for cloud broker enables trust evaluation of the cloud service providers using different modes of cloud broker such as cloud service recommendation, cloud service intermediation, cloud service aggregation and cloud service arbitration.

The evaluation of the trust model for cloud environment is performed as a case study using a cloud computing application, a cloud broker based deployment architecture and also part of the trust model is evaluated using Amazon data set. The evaluation of trust model incorporated with uncertainty function and using the Amazon marketplace data set reveals low prediction errors in comparison to some of the well-known trust models. The simulation of the trust framework shows the robustness of the trust model against malicious feedback providers due to the incorporation of credibility function and the early stage filtering.

The proposed trust model is validated in the EU funded project OPTIMIS and some of the open future research challenges are presented.

Dedicated to my parents, my wife Deepa, my daughter Shruti and my son Swayam…

Chapter 1 Introduction

Cloud computing has been recognised as an important new paradigm to support small and medium size businesses and general IT applications. Cloud service providers offer variety of services that includes softwares, platforms as well as infrastrcucture services. The advantages of cloud computing are multifold including better use and sharing of IT resources, unlimited scalability and flexibility, high level of automation, reduction of computer and software costs, and access to several services. This attract organizations to adopt cloud services to incorporate cheaper and more agile IT resources into their systems. The popularity of cloud computing technology introduces several cloud service providers in the market to offer cloud services. However, despite the advantages and rapid growth of Cloud computing, it brings several security, privacy and trust issues that need to be addressed(Pearson, 2013).

1.1 Motivation and Research Challenges

Cloud services offered by cloud service providers require consumers data and application to cross the organization boundries of the consumer. The movement of data and application worries the consumers for the confidentiality and privacy of the data. The consumer is also concerned about the security of the application running in the cloud providers environment which can be targeted by network attacks.

The consumer expects high availability, reliability, security and elasticity for its services running in the cloud environment. The current cloud market offers the consumer with huge number of cloud service provider for its service, but the consumer may not have prior experiences with any of the the cloud service providers or the large number of cloud service providers complicate the decision process to select the service provider that is suitable for its service. The consumers expect high level of trust in the cloud service providers to deploy its service in the cloud environment, however it is difficult to recognize the trustworthiness due to the dynamic behaviour exhibited by the cloud service providers.

1.1.1 Trust in Cloud Computing

The concept of trust is fundamentally applicable in diverse fields like psychology, economics, sociology and political science and also extensively used in computer science (Mcknight and Chervany, 1996). Cloud computing being a new paradigm, the exploration of trust concepts within cloud computing, for its various service delivery mechanism and deployment models have just begun. Trust in the context of security of applications, data protection, resource requirement, legal constraints and many such topics in cloud computing environment are yet un-explored areas and need intense study for wider adoption of the clouds.

Several challenges such as specification of SLAs (Service Level Agreements), cloud standards, security measures, selection of service providers and computation of trust still persists, depicting that the cloud environments are still not sufficiently trustworthy from customer's perspective (Habib et al., 2010). Trust is an important concept for cloud computing given the need for consumers in the cloud to select cost effective, trustworthy, and less risky services (Ferrer et al., 2012). The issue of trust is also important for service providers to decide on the infrastructure provider that can comply with their needs, and to verify if the infrastructure providers maintain their agreements during service deployment. Other challenges that distress the consumers or cloud users are the lack of flexible application to infrastructure mapping and the requirement of non-trivial networking among all resource providers (Zhao et al., 2012).

In this research, we propose a trust model and trust framework tailored for the cloud computing environment. The challenge in defining a trust model for cloud is mainly due to the cloud computing environment, that largely differs from the other areas, such as: electronic market environment, peer-to-peer network, mutli-agent systems, grids, Service Oriented Architectures (SOA), where most of the current trust models are available (Halberstadt and Mui, 2001; He et al., 2009; Maximilien and Singh, 2004; Olmedilla et al., 2005; Resnick and Zeckhauser, 2002; Sabater and Sierra, 2002; Zhou and Hwang, 2007).

Below are some of the specifics of cloud computing environment which pose challenges while designing the trust framework for cloud:

- Trustworthiness of entities in electronic market is predominantly based on the transaction history. However, in the cloud environment, a transaction can last for over a long period of time, hence the trustworthiness of entities have to be based on the historical information of the transactions as well as based on the performance of the in-progress transaction.
- Cloud consists of several deployment architectures and service delivery models; hence a single trust model may not satisfy all deployment architectures and service models.
- The essential cloud characteristics i.e. *on-demand self-service*, *broad network access*, *resource pooling*, *rapid elasticity* and *measured service* demands the trust model to be sensitive to these characteristics

Existing trust models either do not consider or partially consider the cloud specific behaviours within the trust framework.

To deal with the challenge of identifying dependable cloud service providers for the service, cloud market places are gaining popularity that assists sellers and buyers, however, the concern about the trustworthiness of the providers still remains unanswered (Zhao et al., 2012). Habib *et al.* also highlighted several challenges and proposed recommendations which indicates the need for a mediation layer to evaluate the cloud service providers and that the third parties like cloud broker can play an important role to assist the consumer in selecting an appropriate provider as well as assist in the deployment of the service (Habib *et al*., 2010).

1.2 Research Hypothesis and Objectives

The primary objectives of this research are:

- To devise a trust model suitable for the cloud environment that allows trust evaluation of the cloud service provider for a particular service and enables the selection of trustworthy provider for the service
- To devise a robust trust model that is resistant to malicious feedbacks
- To design and develop an independent mediation layer in the form of cloud broker in cloud computing environment for evaluating the trust model
- To incorporate trust model to provide optimized cloud services along with other factors such as risk, eco-efficiency and cost, within the EU funded OPTIMIS (Optimized Infrastructure Services) project.

The research carried out in the area of Cloud for defining the trust framework and the trust model presented in this thesis and the evaluation results obtained will determine if the findings lead to an acceptance of the following hypothesis:

H1: Trust models can be built for evaluating trustworthiness of Cloud entities to create a trusted environment within Cloud. This hypothesis can be split into two sub-hypothesis:

H1a: The method to analyse trust in the cloud environment may involve consideration of different aspects of the cloud service providers

H1b: Interactions between interconnected cloud entities and the information of interactions may be valuable sources of information that can be considered for the trust models

The research objectives to accomplish the primary objectives are as follows:

- Literature review

  To provide literature review in the area of trust systems, reputation and recommendations systems, to analyse existing trust models and techniques, to identify their purpose, strengths and weaknesses. To provide comprehensive study towards advances in the framework for trust management in the cloud computing.

- Trust Models for Cloud

  To design and implement trust models that considers different aspects of Cloud providers. To design trust model considering the behaviours exhibited by the Cloud computing entities and the parameters relevant to essential Cloud characteristics.

- Cloud broker mediation layer to evaluate trust model

  To design and implement an independent mediation layer of cloud broker with different modes of operation that will deal with the surrounding challenges such as establishing SLAs, providing flexible layer to map application to infrastructure mapping and providing security support. Exploit the different modes of cloud broker to achieve the required trust evaluation of cloud service providers

- Evaluation

  To perform comprehensive evaluation, of the proposed trust models for cloud. Assess if the evaluations of the proposed trust models support the hypothesis or not.

1.3 Contribution of this Thesis

This section presents the work done to achieve the objectives and summarizes the achievements.

The contributions of this thesis are:

- We propose a trust model for cloud that considers in progress transaction information in terms of SLA (Service Level Agreement) violations to model the trustworthiness of the cloud providers. This trust model is supported with the proposed uncertainty model that is used in the representation of the opinion. Evaluation of this opinion model representation provides significant enhancements over existing trust models.

The proposed trust model calculates trust values based on three different parameters, namely (i) *compliance of SLA parameters* (ii) *service and infrastructure providers satisfaction ratings* and (iii) *service and infrastructure provider behaviour*. This trust model is supported with the opinion model that considers *belief*, *disbelief*, and *uncertainty* where the uncertainty is considered based on the amount of evidence and on the dominance that exist between the positive and negative evidences. To combine beliefs, the trust model uses the Subjective logic framework which is a belief calculus specifically developed for modelling trust relationships.

- We extend the trust model by incorporating the essential cloud characteristics and credibility features into the trust model

  The extended trust model considers the essential cloud characteristics such as *resource pooling*, *rapid elasticity*, as the dimensions of the trust model. The trust model considers several features relevant to the dimensions such as *availability*, *time*, to build context. The trust model is supported with an opinion model that considers *uncertainty* for building context specific trust and *credibility* to reduce the impact of malicous feedback providers. Early filtering of malicous feedback mechanism compliments the *credibility* by further reducing the influence of malicous node. The evaluation of the proposed trust model exhibits the robustness against malicious feedback providers.

- We propose a mediation layer of cloud broker with different modes of operation that enables variety of trust assessments within cloud computing environment. The cloud broker architecture can be used as a) *cloud service recommendation* b) *cloud service intermediation* c) *cloud service aggregation* or d) *coud service arbitrage*. The cloud broker used as *cloud service recommendation* enables trust assessment of the individual cloud service providers. The cloud broker used as *cloud service intermediation* enables assessment of security reputation of the cloud service providers while the cloud broker used as *cloud service aggregation/arbitrage* enables trust assessment for the group of providers.

- We propose the detailed architecture for assessing the security reputation of a cloud service provider using the cloud broker. To select a cloud service provider that meets the expectations and needs of one's security requirements is not easy. As a solution, we use the proposed broker architecture model that enables us to build a security reputation framework for cloud service providers, capturing comprehensive evidence of security information to build its trust and security reputation.

  *C*loud broker not only support trust computation but also provides tools that aid the consumer during deployment and perform monitoring during operational stages of the service. Our broker architecture with intermediary mode is flexible enough to support a wide range of value-added services which are usually expected by the consumers. We present additional Cloud based services such as VPN (Virtual Private Network), Intrusion Prevention System (IPS) and Secure Storage service as value-added security services which are provided by using an intermediary cloud broker. Due to the capability of security value added services, *cloud service intermediation* supports capability to provide security reputation of the cloud service providers.

1.4 Organisation of this Thesis

The overall thesis is organized as follows:

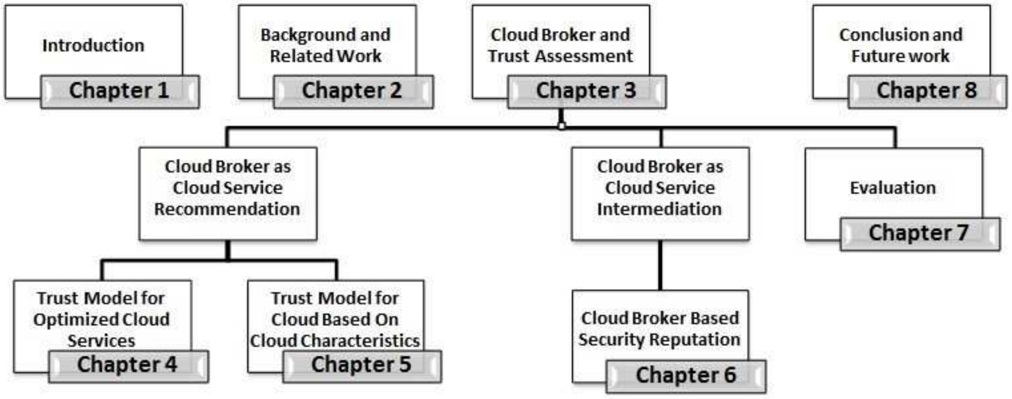

Introduction
Chapter 1
Background and Related Work
Chapter 2
Cloud Broker and Trust Assessment
Chapter 3
Conclusion and Future work
Chapter 8
Cloud Broker as Cloud Service Recommendation
Cloud Broker as Cloud Service Intermediation
Evaluation
Chapter 7
Trust Model for Optimized Cloud Services
Chapter 4
Trust Model for Cloud Based On Cloud Characteristics
Chapter 5
Cloud Broker Based Security Reputation
Chapter 6

Figure 1.1: Structure of the thesis

- **Chapter 1: Introduction**

This chapter provides the motivation for this research and the research challenges. This chapter also defines the research objectives, contributions of this thesis and the organisation of this thesis.

- **Chapter 2: Background and Related Work**

This chapter provides a detailed report on the literature review study done to progress with this research work and also provide the necessary background information of various topics required for this research.

- **Chapter 3: Cloud Broker and Trust Assessment**

This chapter proposes the various trust evaluations of the cloud service providers with the use of different modes of the Cloud Broker (CBR).

- **Chapter 4: Trust Model for Optimized Cloud Services**

This chapter provides a detailed report on the trust model proposed that computes trust based on three different parameters, namely (i) *compliance of SLA parameters*

(ii) *service and infrastructure providers satisfaction ratings* and (iii) *service and infrastructure provider behaviour*.

- **Chapter 5: Trust Model for Cloud Based On Cloud Characteristics**

This chapter provides a detailed summary on the trust model extended with essential cloud characteristics such as 'on-demand' self-service, resource pooling, rapid elasticity and measured service. These cloud characteristics are considered as the dimensions of the trust model.

- **Chapter 6: Cloud Broker based Security Reputation**

This chapter provides a detailed report on the proposed cloud broker architecture that is used for trust evaluation of the cloud service providers. In this chapter, we also propose a security reputation framework for cloud service providers using the cloud broker architecture.

- **Chapter 7: Evaluation**

This chapter provides the evaluation of the trust models defined in Chapter 4, Chapter 5 and Chapter 6. It describes in detail, set of the experiments performed, the purpose of the experiments, the results obtained and inferences made from the evaluation results.

- **Chapter 8: Conclusion and Future Work**

This chapter concludes this thesis by summarising the presented work and outlining issues that remain open for future research.

Chapter 2 Background and Related Work

2.1 Introduction to Trust

The concept of trust is fundamentally applicable in diverse fields (Mcknight and Chervany, 1996) like psychology, economics, sociology and political science and also extensively used in computer science. In the last decade, the use of trust in the field of computer science is observed in diverse areas such as e-commerce, peer-to-peer communications systems, multi-agent systems, Service Oriented Computing (SOC), security and access control in computer networks, reliability in distributed networks, game theory and agent systems, and policies for making decision under uncertainty (Blaze et al., 1999; Jia et al., 2012; Kokash et al., 2007; Mui et al., 2001; Pujol et al., 2002; Rasmusson and Jansson, 1996; Resnick and Zeckhauser, 2002; Spanoudakis and LoPresti, 2009; Zhou and Hwang, 2007).

The rapid growth of cloud computing technology also exhibit concerns of trust. Several solutions are being proposed to ensure the security issues within the cloud environment but these are still at their infancy. Likewise service models being most important in the cloud environment and associated trust issues related to the service provider, service consumer and overall trust within the cloud environment have also received high attention in recent years.

To position the work in this research, some of the background work and the related works mainly in the areas of trust and cloud computing are reviewed.

2.1.1 Trust Definitions

Trust plays an important role in our day to day life and is a key to interpersonal relationships in various settings. Even though trust is extensively studied in various disciplines, such as economics, philosophy and computer science, its definition still remains as a debatable topic and the diverse definitions of trust continue to be used. The most commonly adopted definition by many of the researchers is the definition provided by sociologist Diego Gambetta (Gambetta, 2000) :

> *"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 ever to be able to monitor it) and in a context in which it affects his own action."*

The definition refers to the following important elements of trust: a) trust involves two parties: an assessor and one who is assessed; b) trust is subjective; c) trust involves context.

Trust is a relationship and building of trust requires two parties: *trustor* and a *trustee*. The *trustor* (or assessor) is a relying party that evaluates the *trustworthiness* of the trustee. The *trustee* is a party under evaluation of its *trustworthiness*.

The *trustor* evaluates the *trustworthiness* of a *trustee* for a specific context. The trust relationship between trustor and trustee is always specific to some *context*. Two parties can have multiple trust relationships for a variety of contexts.

Many other general definitions from existing research provide reference points for understanding of trust. Mui *et al.* defines trust referring to historical evidence which is given as follows (Mui and Mohtashemi, 2002):

> *"[Trust is] a subjective expectation an agent has about another's future behaviour based on the history of their encounters."*

The definition by Grandison and Sloman (Grandison and Sloman, 2000) refers to belief and competence of an entity in addition to context:

> *"[Trust is] the firm belief in the competence of an entity to act dependably, securely, and reliably within a specified context."*

The definition by Olmedilla *et al*. (Olmedilla *et al*., 2005) refers to action and not competence like the previous definition:

> *"Trust of a party A to a party B for a service X is the measurable belief of A in that B behaves dependably for a specified period within a specified context (in relation to service X)."*

Jøsang *et al*. (Jøsang et al., 2005) differentiates between reliability trust and decision trust and provides definitions for the different forms of trust. Reliability trust is defined as:

> *"Trust is the subjective probability by which an individual, A, expects that another individual, B, performs a given action on which its welfare depends."*

The decision trust is defined as:

> *"Trust is the extent to which a given party is willing to depend on something or somebody in a given situation with a feeling of relative security, even though negative consequences are possible."*

The definition of reliability trust is close to the definition of trust defined by Gambetta. The definition of decision trust extends the previous definitions by introducing aspects of a broad notion of trust that includes dependence, reliability, utility and risk attitude.

2.1.2 Trust and Security

The earliest work describing the difference between trust and security approaches is by Rasmussen & Jansson (Rasmusson and Jansson, 1996) who used the term hard security for traditional security mechanisms like authentication and access control, and soft security for what they called social control mechanisms in general, of which trust and reputation systems are examples. Cryptographic techniques, firewalls, authentication and other security mechanisms are termed as hard security mechanisms that either allow complete access or no access. These mechanisms protect systems and data from malicious entities and can be considered highly reliable and thus more trustworthy. Authentication provides so-called *identity trust* and CAs (Certifying Authorities) and other authentication service providers support verifying and managing identities. However, user may also be interested in knowing the reliability of the authenticated party or the quality of service they provide. The latter type of trust will be called as *provision trust* and only soft security mechanisms like trust and reputation systems are useful in deriving provision trust.

2.1.3 Trust and Reputation Systems

Trust and Reputation often goes hand in hand, however there is a certain difference. To provide more clarity on this, the section provides a definition of reputation related to trust. Jøsang *et al* (Jøsang *et al.*, 2005) defines reputation as :

> *"Reputation is what is generally said or believed about a person's*
> *or thing's character or standing."*

Trust is more of a personal and subjective phenomenon that is based on the various factors or evidence. Reputation can be considered as a collective measure of trustworthiness based on the referrals. The main difference between trust and reputation system is that trust systems produce score that reflects relying party's subjective view of an entity's trustworthiness while the reputation systems produce an entity's reputation as seen by the whole community. Trust may be used to determine reputation of an entity and the other way round, reputation may also be used to determine the trustworthiness of an entity.

2.1.4 Properties of Trust

Trust in general have the following properties (Abdul-Rahman and Hailes, 2000; Grandison and Sloman, 2000): Trust is *subjective*. Every *trustor* have its own perspective towards a *trustee* which may result in different trust value for a trustee by each of the trustor i.e. the trust of an entity A in an entity B does not need to be the same as the trust of entity C in an entity B. The degrees of belief associated with trust value ranges from complete distrust to complete trust. Trust is *asymmetric* i.e. trust of an entity A in B does not mean trust of an entity B in A. Trust is *context-dependent* and *situation-dependent*. Entity A may trust entity B as provider of banking service but may not trust as a provider of computer hardware service. In terms of recommender, entity A may trust entity B as a good recommender for films but may not trust as a recommender for medicines. Trust is *dynamic* and *non-monotonic* i.e. experience can increase or decrease trust. By the property of trust transitivity, if Alice trusts Bob, and Bob trust Claire than Alice will also trust Claire. Trust is *not transitive*, however some trust scenarios such as, trust delegation, do exhibit transitivity. Also the concept of recommendations is important to establish trust in entities about which any or only little direct experience is available.

2.1.5 Trust Categorization

McKnight *et al.* (Mcknight and Chervany, 1996) categorizes trust in three major categories: *impersonal/Structural* trust, *Dispositional* trust and *personal/interpersonal* trust. *Impersonal/structural* trust is based not on person or person attributes but rises from social or institutional structures, for example: (i) trust as a function of the assurance provided by such social structures as banking regulations; (ii) trust that the judicial system will uphold contract law. *Dispositional* trust means that the trust is based on the personality attributes of the trusting party i.e. trustor has a general tendency to trust others across situations or general faith in human nature. Personal/Interpersonal trust means that two or more people trust each other in a specific situation.

McKnight also introduced a concept called *Trusting Beliefs* which means to expect the person to be *benevolent* (willing to serve another's interest), *honest* (proving the willingness by making fulfilling agreement to do so), *competent* (capability to service another's interest) and *predictable* (one's willingness and ability to serve another's interest does not vary over time). If it is possible to find a person with these qualities, interaction with this person would be expected to have a positive outcome.

2.2 Trust Management

Grandison *et al.* identifies a number of trust classes: access to trustor's resource, provision of trust by the trustor, certification of trustees, delegation and infrastructure trust (Grandison and Sloman, 2000). Trust also has been segregated in two other dimensions: trust in an entity to perform action, and trust in an entity to recommend other entities to perform action. Distinctions also have been made between trust resulting from direct observation and assessment of the trustee and trust that is derived from the trust conveyed by the recommenders. However, the most trust research classifies trust management into two main areas: p*olicy based* trust management and r*eputation based* trust management (Artz and Gil, 2007; Bonatti et al., 2005, 2004). The other trust management approaches are also depicted in Figure 2.1.

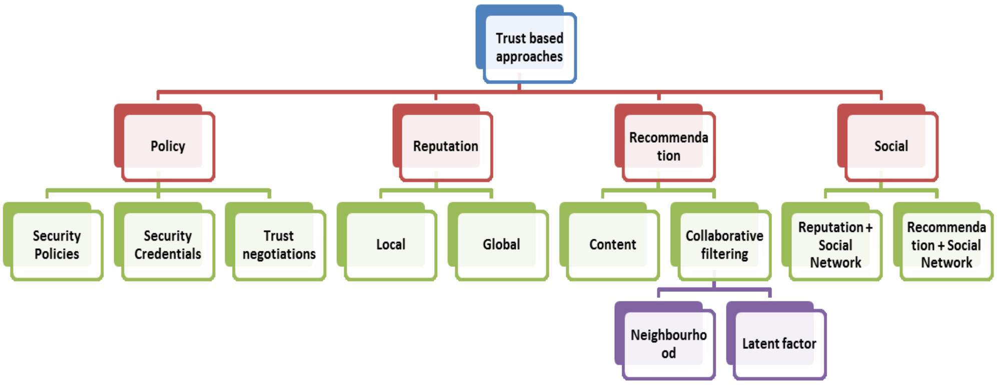


Figure 2.1: Classification of Trust approaches

2.2.1 Policy Based Trust Management

Policy based trust involves policies that describe the conditions necessary to obtain trust and can also prescribe actions and outcomes if certain conditions are met(Maximilien and Singh, 2004). It involves managing and exchanging credentials and enforcing access policies. It is assumed that trust is established by gaining sufficient amount of credentials pertaining to a specific party. The credentials are usually certificates, which have been signed by trusted third party. The credentials may state information about the identity of the owner or information about the rights of the owner. Blazé *et al*. defined trust management in the context of policies which is given as follows (Blaze et al., 1999, 1999):

> *"a unified approach to specifying and interpreting security policies, credentials, and relationships which allow direct authorization of security-critical actions"*

The definition describes what can be considered to be traditional approach to trust management, i.e. trust is only treated implicitly and in a rather static manner.

While unknown parties interacting with each other, certain level of trust must be established. Trust negotiation (Bonatti et al., 2004) is an approach that enables trust establishment by gradually disclosing credentials and requests for credentials.

In environments where interacting parties do not know each other, a certain level of trust is established through exchange of information between the interacting parties and the trust establishment process is bi-directional. Trust negotiation is an approach to automated trust establishment which regulates the exchange of sensitive credentials using access control policies. It is an iterative process where trust is established gradually by disclosing credentials and requests for credentials(Bonatti et al., 2004; Winsborough and Li, 2002). Trust negotiation is triggered when one party requests access of resource owned by another party. The goal of a trust negotiation is to find a sequence of credentials (C1, C2,…,Ck, R), where R is the resource to which access was originally requested such that when credential Ci is disclosed, its policy has been satisfied by credentials disclosed earlier in the sequence or to determine that no such credential disclosure sequence exist. As trust negotiation can expose sensitive attributes, it is essential to protect sensitive attributes in the process(Lei and Li, 2014). One of the drawbacks of policy-based trust management is that it usually relies on a trusted third party that issues the certificate, stating that an entity is considered to be trustworthy. Furthermore, certificate based approaches that rely on public key infrastructure require further means for the distribution, verification and revocation of key.

2.2.2 Reputation Based Trust Management

Reputation based trust is established using the past interactions of an entity to assess the future behaviour. Reputation based trust for an entity is computed by considering the historical information and the referral trust in absence of (or in addition to) direct interactions. In reputation systems, parties can rate each other and a reputation score can be derived from aggregated ratings about a given party. This assists other parties in deciding whether to transact with a given party in the future.

Resnick *et al.* (Resnick et al., 2000) discusses the importance of reputation system in the Internet services despite of several theoretical and practical difficulties. In the Internet scale where large number of producers or consumers may not know each other, reputation systems help people to decide whom to trust and encourage trustworthy behaviour. A popular site such as eBay (Resnick and Zeckhauser, 2002), hosting online market, attributes its high rate of successful transaction to reputation systems, where buyer and seller rate each other (1, 0 or -1). Resnick *et al.* (Resnick *et al.,* 2000) emphasizes the importance of reputation systems not only to the auction sites but to various other sites such as Bizrate, expert sites. Bizrate.com rates registered retailers through the feedback from consumers after each purchase, expert sites (www.expertcentral.com) where experts provide answers to questions posted by other users, product review sites (www.epinions.com) offer rating services for product reviewers, (iExchange.com) tallies and displays reputation for stock market analyst.

A Reputation system helps examine how trust builds naturally in long term relationships using the past transactional data. Ratings are not the only way to convey reputation, while agreeing for being rated (such as registered retailer as bizrate.com) is an indication of high quality services offered by service provider.

Reputation systems require three properties: a) Long lived entities, b) Capture and distribution of feedback about current interactions, c) Use of feedback to guide trust decisions.

Though Internet can vastly accelerate and structure the process of capturing and distributing information, significant challenges remain in the operating phase of reputation systems: eliciting, distributing and aggregating feedback. Eliciting feedback incurs problems such as, (a) people may not bother to provide feedback, (b) common practice of not providing negative feedback unless until really bad performance, (c) difficulty of ensuring honest reports and (d) providing negative or positive feedback intentionally. Distributing feedback incurs problems such as use of pseudonym which can be frequently changed and lack of portability from system to system. Aggregating feedback refers to influence on decision making, about whom to trust, based on the information gathered and displayed (e.g. numerical ratings fail provide information whether the feedback came from low value or high value transaction).

Despite of the theoretical and practical difficulties, reputation systems play an importance and significant role in building trust.

Reputation systems can be categorised into *local* reputation or *global* reputation based on the information used to build the reputation. In a global reputation system (Zhou and Hwang, 2007), the trustee is evaluated by different trustors considering the trustee's transaction information in the whole system. In such systems trustors always obtain the same reputation value for a trustee. In local reputation systems (Jia *et al*., 2012), the trustee is evaluated by trustors considering the trustee's transaction information with subset of the entities in the whole system.

Advantage of the reputation based approach is that it poses very little requirements to the environment it is applied to. The evidence which is required to evaluate the trustworthiness of an entity is created by participants of the system and distributed within the system. This approach does not require additional infrastructure or trusted third parties.

The limitation of this approach is that it does not directly state whether an entity is trustworthy or not, rather it provides a degree of trust associated with an entity. As trust establishment may include recommendation, the system needs to cope up with misleading recommendations. This approach requires bootstrapping of trust model which may need some external information about trustworthiness of entities.

2.3 Recommender Systems

Current competitive markets provide consumers with enormous choices. To help consumers in their decision, organisations that host marketplaces, use recommender systems that provide recommendations based on the analysed patterns of consumer interest in products. Recommender systems are broadly classified into two categories (Koren et al., 2009): a) *content filtering systems* and b) *collaborative filtering systems*. The *content filtering* approach creates profiles for user or product that characterizes its nature and allows programs to associate users with matching products. The collaborative filtering approach analyses relationship between users and interdependencies among products to identify new user-items associations. Despite of being domain independent, *collaborative filtering* approaches seem to provide more accurate results compared to *content filtering*. Two primary areas of *collaborative filtering* approaches are *neighbourhood* methods and *latent factor* models. *Neighbourhood* methods are centred on computing the relationships between items or, alternatively, between users. *Latent factor* models are an alternative approach that tries to explain the ratings by characterizing both items and users on, say, 20 to 100 factors inferred from the ratings patterns. *Latent factor* models (Hofmann, 1999; Ma et al., 2009; Salakhutdinov and Mnih, 2008) based on matrix factorization methods are being the most successful realizations which characterizes both items and users by vectors of factors inferred from item rating patterns.

2.4 Social Network Trust

Social networks can be derived in many ways such as: user connected though transaction in online auctions, users who post within same thread on a news group, or even member of groups listed in an HTML document can be turned into a social network. With network topologies that can be automatically extracted, social network provide large source of data for the more mathematical and structural types of analysis.

In recent years, social networks have proliferated (FaceBook, del.icio.us, Y! Answers, Flickr, MySpace, LinkeIn, Twiter, CourseRank). Social networks have become as a common medium for disseminating and connecting like-minded people. In these networks, users can contribute and share different types of resources, ranging from personal information and photos to opinions and ratings.

Some social networks such as LinkedIn have trust implied in the network connection while other networks such as Orkut has notion of trust where users assign trust ratings to their friends. Network topologies extracted and network data collected can be composed to produce information about the trust between two individuals without a direct connection and can also be used to recommend to one user on how much to trust other user. The use of social networks to assist with reputation systems has been advocated in (Pujol et al., 2002; Sabater and Sierra, 2002). The work in (Pujol et al., 2002) deduces the reputation of members in a community based on the social network topology (i.e. a member's position in the network specified by the number of relations that the member has with other members). The approach in (Sabater and Sierra, 2002) extends REGRET (Reputation model for gregarious societies) system (Sabater and Sierra, 2001a, 2001b) to consider social relations for reputation of agents.

The work in (De Meo *et al.*, 2009) uses the trust and reputation to promote interactions among users of different social networks by suggesting the most reliable users with whom to interact. In this work, trust is modelled as a multi-dimensional concept and considers it to be context specific.

2.5 Recommender Systems and Social Network Trust

It has been a common practice for collaborative filtering techniques to depend only on the user-item rating matrix for recommendations. However, with the recent growth of social networks and the intelligence that can be extracted from this network, inspires recommender system to utilize the social trust relations among users for recommendations. In recommender systems, Massa and Avesani (Massa and Avesani, 2007, 2004), replaced the similarity finding process with the use of trust metric to propagate trust over the trust network and estimate trust weight. A very popular factor analysis method based on probabilistic graphical model proposed by Ma *et al.* (Ma et al., 2008) fuses user-item matrix with users social trust networks by sharing common latent low-dimensional user feature matrix. In (Ma et al., 2009) this work also proposes the probabilistic factor analysis framework with the aim of modelling recommender system accurately.

2.6 Trust Models

Trust management approaches differ on how trust is represented and how trust is computed. This creates a separation between two aspects of trust models, namely representational aspects and computational aspects, leading to *representational* and *computational models* respectively.

A representational model defines how trust is represented and established and a computational model defines how different sources of trust related evidence are aggregated.

2.6.1 Aspects of the Computational Model

The computation model defines how different sources of trust i.e. direct evidence and recommendations are integrated. It is important to consider here if the trust value is supposed to be subjective trust value or whether it is a global trust. If the trust value depends on the entity which evaluates the trust in another entity, then it is called as subjective trust value. If the trust values computed is independent from the entity that evaluates the trust then it is considered as global trust value. In computation of subjective trust values, the recommendations gathered from subset of all entities are applied to the subjective measure to define the impact of the collected recommendations. Also trust model that provide means for the computation of subjective trust values differ in the mechanisms to filter and weight the recommendations before calculating the trust value. In computation of global trust value, recommendations from all the entities and trust links between them are taken into account and the trust value is independent from the entity that evaluates the trust.

2.6.2 Aspects of the Representational Model

Representation model of trust defines how trust is represented and established. Differences in the representation of trust can be found with respect to the *domain* of trust value. A binary domain allows only two states: “trusted” or “untrusted”. This can be observed in certification based trust or policy based trust approaches, where access is either granted or denied based on the credentials available.

Binary models are insufficient as trust relationship may bear a *trust level (Bonatti et al., 2005)*, that characterizes the degree of trust in the trustee. The trust levels can be represented either as *discrete* or *continuous* numbers.

Representation of trust can also differ on the dimensions, i.e. the number of parameters. One dimensional representation allows only trustworthiness of an entity to be expressed as a single parameter. Multi-dimensional representations (Jøsang, 2001) include parameters such as uncertainty, reliability and confidence associated with the trust value to be expressed as supporting parameters in addition to the trustworthiness of an entity.

Another important aspect is the interpretation of the meaning of the trust value. In (Jøsang et al., 2007), semantics of trust measure is described in terms of *specific-generality* dimension and a *subjectivity-objectivity* dimension. A specific measure relates to a specific trust aspect such as ability to deliver on time where as general measure is supposed to represent an average of all aspects. A subjective measure means rating based on subjectifve judgement whereas objective measures means rating which have been determined objectively by assessing the trusted party against formal criteria.

Semantics provided by Jøsang et al. to the calculated trust values are as follows: *ranking*, *rating*, *probability*, *belief* and *fuzzy logic*(Jøsang *et al*., 2007). The trust values computed in *ranking* based approaches provide no meaning to the trust values except that it specifies that higher ranked entities have higher trustworthiness than the lower ranked entities. The trust values that are directly linked to the trust semantics are referred to as *ratings*. For example, ratings in the interval [1-2] can be treated as "very untrusted", ratings in the interval [9-10] "very trusted" and the intermediate ratings can be considered to have semantics something between "very untrusted" to "very trust". If trust is modelled as *probability*, the trust value expresses the probability that the entity will behave as expected. Trust values expressed as *belief* allows to express uncertainty associated with the trust in an entity. Trust models based on the *fuzzy logic* introduce their own semantics to the calculated trust values based on membership functions.

2.7 Reasoning with uncertain information

Trust based systems as well as most task requiring intelligent behaviour have some degree of uncertainty associated with them. This section describes the study of some of the approaches to handle reasoning with uncertain information (Amour, 2014).

Uncertainty refers to situations where the information available to the decision maker is too imprecise to be summarized by a probability measure. Uncertainty arises in partially observable systems and/or stochastic environments, as well as due to ignorance. For example, in knowledge based system uncertainty may be caused by problems with data: (a) Data may be missing or unavailable (b) Data may be present but unreliable (c) Representation of the data may be imprecise or inconsistent.

Some of the most common ways of handling uncertainty include:

- Bayesian Probability
- Dempster-Shafer Theory
- Subjective Logic

2.7.1 Bayesian Probability

The Bayesian interpretation of probability can be seen as an extension of propositional logic. It allows for reasoning with proposition whose truth values are uncertain. Unlike a frequentist view of probability, in which probability of a proposition represent frequency of the event occurring, in Bayesian probability the probability of a proposition represents a state of belief and can be interpreted as a degree of belief (Amour, 2014; Nau, 2001).

The Bayesian methods are characterized by the following concepts:

- Use of random variables to model all sources of uncertainty in statistical models
- Determine prior probability distribution taking into account the available (prior) information
- When more data becomes available, calculate the posterior distribution using the Bayes formula; subsequently, the posterior distribution becomes the next prior
- The frequentist probability of hypothesis is either one or zero. In Bayesian statistics, a probability can be assigned to a hypothesis that can differ from 0 or 1 if the truth value is uncertain

Bayes' Formula:

$$P(A|B) = \frac{P(B|A) * P(A)}{P(B)} \qquad (2.1)$$

P(A) represents the prior probability of the proposition A being true and *P(A|B)* is the conditional probability of A being true given B is true. Therefore as new evidence becomes available, the probability distributions describing the propositions are updated and these updated probabilities are then used as priors for further calculations with new evidence

Bayesian probability theory indicates its suitability in uncertainty management due to its sound theoretical foundation in probability theory and being the most mature of all the uncertainty reasoning methods. However Bayesian methods are not without limitations, it requires significant amount of probability data to construct a knowledge base and also requires prior and conditional probabilities.

While Bayesian Probability appears to be fairly simple method of extending propositional logic to handle uncertainty, one issue arises is when one wants to carryout abductive inference. The base rate fallacy occurs when one assumes that *P(A|B) = P(B|A)*, and therefore when one wants to reason backwards from some observable evidence to the likely hypothesis, the conditional probabilities must first be inverted (Jøsang and Sambo, 2014; Koehler, 1996). Subjective Logic, as will be shown, supports both deductive and abductive reasoning as operators, and thus no confusion can occur as long as the correct operator is chosen (Jøsang, 2008).

2.7.2 Dempster-Shafer Theory

The Dempster-Shafer theory is based on two ideas: the idea of obtaining degrees of belief for one question from subjective probabilities for a related question, and Dempster's rule of combining such degrees of belief when they are based on independent terms of evidence (Shafer, 1992). It is an extension of Bayesian Probability in which probabilities are assigned not to individual random variables, but to sets of them. The belief of an individual random variable is bounded above and below by two values: plausibility of the random variable and the belief of it (Kay, 2007).

The frame of discernment (or Power set) of X is the set of all possible subsets of X e.g. If $X = \{ x_1, x_2, x_3\}$, then the frame of discernment is : ($\emptyset, x_1, x_2, x_3, \{ x_1, x_2\}, \{ x_1, x_3\}, \{ x_2, x_3\}, \{ x_1, x_2, x_3\}$). Ø is the empty set and has a probability of zero, since one of the outcomes has to be true. Each of the other elements in the power set has probability between 0 and 1.

Given a frame of discernment, a set containing all mutually exclusive atomic events that are of interest to our reasoning systems, one constructs a basic belief assignment or BBA, which assigns a measure of belief between zero and one to subsets of the frame. BBAs are additive: if X is a frame of discernment and m is BBA over X, then $\sum_{x \subset X} \mathrm{m(x)} = 1$. Furthermore no mass is assigned to the empty set m(Ø). Given a BBA m over a frame X, one can compute the belief and plausibility of a subset A of X. The value of belief and plausibility bound the probability of A from below and above. That is $bel(A) \leq P(A) \leq pl(A)$. The real novelty of Dempster-Shafer Theory is the Dempster's rule of Combination, which states how the two BBA's generated by two observations can be combined together.

Let m_1 and m_2 be the two BBAs over a frame of discernment *X*. We combine together the two BBAs by computing what is referred to as joint mass denoted my m_{12} that takes into account the conflicting belief between m_1 and m_2. Though Dempster's rule of combination has straight forward calculation, it has been criticized mainly because highly conflicting belief mass distributions produce counter-intuitive results (Jøsang, 1997; Zadeh, 1986). Josang and Pope claimed that Dempster's rule actually represents a method of preference combination while serving as an approximation for other forms of belief combination such as cumulative or average fusion of two beliefs (Jøsang et al., 2009). Despite these criticisms, Dempster-Shafer theory has seen much success when applied to problems such as sensor fusion and neural network classification (Denoeux, 2000; Wu et al., 2002).

Subjective logic that we introduce next, contains several operators for combining beliefs (Jøsang, 2001; Jøsang et al., 2006). Together these operators serve as better tool for combining evidence from different sources in different scenarios.

2.7.3 Subjective Logic

Subjective logic, introduced by Jøsang (Jøsang, 2001), uses elements from Dempster-Shafer theory and models opinions of agents or observers based on the beliefs, disbeliefs and uncertainty. Subjective logic is an extension to probabilistic logic that fixes some of the issues with Dempster-Shafer Theory. Though it is more recent and not as well studied as Bayesian Probability theory or Dempster-Shafer Theory and is under constant refinement, subjective logic has been shown to be effective across a range of areas that require uncertain reasoning, such as trust network analysis , modelling trust on mobile ad-hoc network, and arguing with evidence (Amour, 2014; Jøsang and Bhuiyan, 2008; Li et al., 2004; Oren et al., 2007).

The primary building block of subjective logic expression are objects called subjective opinions (Jøsang, 2001). Opinions in subjective logic can be mapped to and from probability density functions from probability theory (Jøsang, 2001). Binomial opinions correspond to beta probability density functions (PDFs). This is particularly useful for evidence based reasoning because the beta PDF acts as a conjugate prior to the binomial distribution. When the posterior distribution p(θ/y) is in the same family as the prior probability distribution p(θ), the prior and posterior are then called conjugate distributions. This means that through the mapping, subjective opinion can be used anywhere one could use Bayesian Inference, where the Bayesian update mechanism updates the opinions to take into account new evidence. Bayesian statistical inferences can be made based on the evidence available and thus provides a theoretically sound basis for computing reputation scores (Amour, 2014; Jøsang, 2001; Jøsang and Ismail, 2002).

The work on subjective logic has been effectively used for reasoning based on the evidence. The opinion of a proposition x, represented as *w(x)* is defined in terms of belief $b(x)$, disbelief $d(x)$ and uncertainty $u(x)$ where(Jøsang, 2001):

$$b(x) + d(x) + u(x) = 1 \qquad (2.2)$$

The probability expectation of an opinion *w(x)* is given as:

$$E(w(x)) = b + au \qquad (2.3)$$

Where *E(w(x))* is in the range [0, 1] and *a(x)* is relative atomicity in the range of [0, 1].

The subjective logic operates on opinion where the standard logical operators such as *AND* and *OR*, are applied to the opinions and are upgraded to incorporate uncertainty. In the case of absolute belief (bx=1) or disbelief (dx=1), these binomial operators behave the same as traditional logic(Jøsang and McAnally, 2005). Using this opinion model as a base, several mechanisms have been devised to model the belief, disbelief and uncertainty from the evidences available. Jøsang models the belief, disbelief and uncertainty as

$$b = \frac{\mathrm{r}}{r + s + 2} \qquad (2.4)$$

$$d = \frac{\mathrm{s}}{r + s + 2} \qquad (2.5)$$

$$u = \frac{2}{r + s + 2} \qquad (2.6)$$

Where *u* is uncertainty which is not equal to 0, *r* = number of positive evidence and *s* = number of negative evidence.

The subjective logic also define non-standard operators such as discounting and consensus operating on opinions. The discounting operator is useful to take second hand evidence for example if an entity *A* can form an opinion about a proposition *x* by discounting *B*'s opinion about *x* with *A*'s opinion about *B*. The consensus operator enables to combine the opinions of entity *A* and entity *B* representing and imaginary entity [*A,B*]s opinion about proposition *x*. Subjective logic operators for belief constraining, can be used when multiple agents need to reach a consensus opinion. This operator is in fact equivalent in meaning to Dempster's rule of combination (Jøsang, 1997).

Subjective logic also includes operators for performing uncertain reasoning (Jøsang, 2001; Jøsang et al., 2006). It includes deduction and abduction operators for subjective opinions, thereby allowing Subjective logic to be used for intelligence analysis, Bayesian network analysis, and other actions that require reasoning when uncertainty is present (Jøsang, 2008).

2.7.4 Motivation for using Subjective Logic

Through the study of three widely used techniques for reasoning with uncertain information, it can be observed that the subjective logic provides various advantages over the other two. Subjective logic is based on concepts of Dempster-Shafer theory and uses some elements (such as frame of discernment) of DS theory. Though the two theories (Dempster-shafer and Subjective logic) have been used for information fusion and handling ignorance, they are distinct. However both the theories have a foundations of Bayesian probability theory and have commonalities as well.

In Subjective logic, beliefs are expressed on frames of discernment (set of possible states). Subjective logic framework can be considered as an alternative to combine information and handle ignorance and uncertainty. It has been proven that Subjective logic is compatible with traditional mathematical framework and is also suitable for handling ignorance and uncertainty (Jøsang, 1997). Subjective Logic framework consists of a belief model called opinion, and set of operations for combining opinions. Subjective logics is an extension of standard logic that contains operators for belief theory such as consensus and recommendations. The Dempster's rule for combination is equivalent to the consensus operator of Subjective logic and both yield similar results in some of the specific scenarios (Jøsang, 1997). Moreover subjective logic contains several logical operators beyond the consensus that assists in reasoning such as deductive as well as abductive that makes it highly suitable for reasoning under uncertainty.

Considering these advantages of subjective logic over other reasoning mechanisms for the purpose of assessing trust and belief in qualities of service offered by service providers and consumers in the presence of uncertainty, this thesis adopted the subjective logic framework and the trust model defined in this thesis is based on the Subjective logic.

2.8 Trust Models based on Belief

Trust and belief have similar meanings and are often used interchangeably. The terms trust and belief are commonly used in relationships or to define relationship between two entities. Philosophically, trust means to place complete confidence in another entity and is considered as a long lasting concept for an entity. If either party breaks the trust, it takes a long time building it back. Belief usually reflects individual facts and is considered as more temporary concept that requires an entity to place faith in another entity for a select time frame.

Research on trust aspects has shown various associations of belief with trust. Researchers have used measures of belief in the 'just world' to associate with inter personal trust and observation shows that individuals with strong belief in a 'just world' show more trust in their future and in others behaviour towards them (Dalbert, 2009). Castelfranchi et al. presents specific beliefs as ingredients of trust and describes the use of trust in social theory (Castelfranchi and Falcone, 1998).

Belief theory represents an extension of classical probability by allowing explicit expression of ignorance. Belief theory has its origin in a model for upper and lower probabilities proposed by Dempster in 1960. Shafer later proposed a model for expressing beliefs. The main advantage of using beliefs is that ignorance i.e. lack of information, can be explicitly expressed. Belief results from uncertainty, and the uncertainty sometimes result from a random process or sometimes only from the lack of information that induces belief.

Trust modelled as probability often use approaches such as Bayesian or maximum likelihood to derive probability from the collected evidence. However Bayesian approaches have been widely criticised for requiring assignment of subjective probability to every event. Dempster-Shafer theory is the most widely known belief based approach used to model Trust which uses belief functions and plausibility functions to attach numerical lower and upper bounds on the likelihood of events. This approach allows assignment of an interval rather than a point value probability, to an event as a representation of the uncertainty of the event.

As handling uncertainty is very crucial for trust and work of Dempster-Shafer (DS) to handle uncertainty motivates the use of belief based approaches in reputation modelling. Yu et al. (Yu and Singh, 2001) proposes reputation model based on the Dempster-Shafer (Kay, 2007) theory of evidence. In (Yu and Singh, 2001) this work the agents are rated based on their own observation and ratings provided by other agents. Trust is modelled as belief of one agent about another and reputation as a cumulative belief from group of agents. This work describes the construction of Trust network by information exchange to gather evidence and mechanisms of combining the evidences using the DS combining rule(Kay, 2007). However, Dempster's rule of combination is a method for fusing belief constraints and only represents an approximate fusion operator in other situations such as cumulative fusion of beliefs providing incorrect results in such situations (Jøsang et al., 2009).

Handling uncertainty is very crucial for trust and belief based approaches are highly suitable for handling uncertainty in trust and reputation modelling. Jøsang's work on subjective logic and opinion modelling is based on the belief theory and takes into account the uncertainty(Jøsang, 2001). An opinion is represented using belief, disbelief and uncertainty. Since the opinions can be mapped to beta PDFs, Bayesian statistical inferences can be made based on the evidence available and thus provides a theoretically sound basis for computing reputation scores (Jøsang, 2001; Jøsang and Ismail, 2002). The cloud being a highly distributed environment, the belief networks become natural choice for representing the probabilistic relationships between the cloud elements. This makes the opinion model that accommodates belief, suitable for the cloud environment. Jøsang's work on subjective logic also provide several logical operators that assists in combining opinions created from the evidence gathered. It has better resistance to the attacks in comparison to many of the Trust and Reputation models(Kerr and Cohen, 2009).

The opinion model proposed in this thesis also considers *belief*, *disbelief*, and *uncertainty* values and is based on an extension of the Josang's opinion model (Jøsang, 2001), in which we consider uncertainty when calculating *belief* and *disbelief* values. In (Jøsang, 2001), uncertainty is considered based on the amount of evidence, in which uncertainty increases if the amount of evidence decreases. In our model uncertainty is considered based on the amount of evidence and on the dominance that exist between the positive and negative evidences.

2.9 Attacks on Reputation Systems

Trust and reputation systems are assumed to predict future quality and the success of a reputation system is measured by how accurately the calculated reputations predict the quality of future interactions. In a distributed environment any party can attempt to exploit the system to its own benefit which creates difficulties in achieving accuracy within reputation systems.

A reputation system that depends on the feedback from other entities in the system is prone to several attacks. Reputation systems attacks can either have narrow focus that only affect the reputation of few selected targets or it can have broader influence affecting large percentages of identities within the system.

Dellarocas (Dellarocas, 2000) identifies two classes of attacks on reputation systems i.e. unfairly low ratings (negative discrimination) and unfairly high ratings (positive discrimination) and proposes a set of mechanisms to eliminate or significantly reduce the effect of these attacks. Kerr et al. also describes different types of attacks on reputation systems and compares the performance of the trust models against the reputation attacks (Kerr and Cohen, 2009).

Hoffman et al. presents a classification of various attacks against reputation systems based on the goals of the reputation systems targeted by attacks (Hoffman et al., 2007). This classification includes: a) Self-Promoting: Attackers manipulate their own reputation by falsely increasing it b) Self-Serving or Whitewashing: Attackers escape the consequence of abusing the system by using some system vulnerability to repair their reputation. c) Slandering: Attackers manipulate the reputation of other nodes by reporting false data to lower their reputation d) Orchestrated: Attackers orchestrate their efforts and employ several of the above strategies. Jøsang presents various strategies for attacking trust and reputations systems that include: Playbooks, Unfair Ratings, Review spam, Discrimination, Collusion, Proliferation, Reputation Lag, Re-entry, Value Imbalance and Sybil Attack (Jøsang, 2012).

The decentralized nature and lack of controlling authority exposes broad range of security attacks. Noor et al. describe several attacks on trust management system and proposes a credibility model that not only identifies fake feedbacks but also preserves privacy of cloud consumers (Noor et al., 2013a). Koutrolli et al. focus on credibility of reputation system and their resistance to comprehensive adversary models (Koutrouli and Tsalgatidou, 2012). Koutrolli et al. classifies type of attacks against reputation system: a) unfair recommendations: entities can spread unfair ratings for other entities or can do it with cooperation with each other to maximize the effect of the attack b) Inconsistent behaviour: Peers can strategically have an inconsistent behaviour that can lead to an incorrect estimation of their reputation allowing them to misbehave and still keep a high reputation c) Identity management related attacks: A malicious entity with multiple identities can have dishonest behaviour and then escape its low reputation by entering these system with new identity.

Several techniques (Dellarocas, 2000; Whitby et al., 2004; Yang et al., 2009) to immunize the effect of unfair ratings or resist the attacks on reputation based system exist in literature.

Hoffman et al. also presents some of the strategies adopted to mitigate the reputation attacks which include (Hoffman et al., 2007) : a) Preventing generation of false rumours either by fabrications or modifications b) Preventing spreading of false rumours either by relying on pre-trusted identities or by employing statistical methods to identify misbehaviours.

The trust and reputation model in this thesis is designed to be robust against reputation attacks and evaluated specifically against unfair recommendations.

The unfair recommendations can either be sent by individuals or by strategically acting collusions of peers. Unfair recommendations from individuals may send unfair negative or positive recommendations (bad mouthing or unfair praises), random opinions or inaccurate recommendations. Unfair recommendations from a group of malicious peers may subvert the system and these attacks can include collusive badmouthing, collusive reducing recommendation, collusive deceit wherein all entities of a group behave badly but provide positive recommendations for each other.

2.10 Cloud Computing

The NIST definition of cloud computing (Mell and Grance, 2011) comprises of five essential cloud characteristics, three service models and four deployment models. The essential cloud characteristics define the characteristics that the cloud computing environment needs to exhibit. The service models define the way services are offered in the cloud computing environment which mainly comprises of IaaS (Infrastructure as service), PaaS (Platform as a service) and SaaS (Software as a service). The deployment models mainly comprises of Private cloud, Community cloud, Public cloud, Hybrid cloud.

2.10.1 Cloud Characteristics

The five essential cloud characteristics are: *on-demand self-service*, *broad network access*, *resource pooling*, *rapid elasticity* and *measured service*.

The *on-demand self-service* characteristic enables the consumer to unilaterally provision computing resources without requiring human interaction.

The *broad network access* characterizes the provider's capabilities to provision over the network and provides access through standard mechanisms that promote use by heterogeneous thin or thick client platforms.

The *rapid elasticity* characteristic of the cloud provider enables the consumer to scale resources rapidly up and down with demand. To the consumer, the capabilities available for provisioning often appear to be unlimited, but in practice, the consumer needs to agree with the provider for the extent of elasticity expected which directly impacts the cost of the service. The elasticity characteristics agreed with the consumers enables the provider to plan its resources. The key to this characteristic is the service level agreement (SLA) that the consumer has with the service provider. The agreement between the service consumer and the service provider enables the consumer to establish the availability and time expectation from each of the service provider.

The *resource pooling* characteristics of the cloud environment enables cloud service providers to use multi-tenant model, dynamically assigning physical and virtual resources with location independence. Multi-tenant model refers to a single instance of software running on a server and serving multiple tenants, while a tenant is a group of users sharing the same view on the software they use. These characteristics of the cloud environment concern the consumer regarding the safety of its service and data residing on the cloud provider's physical infrastructure. The agreement between the consumer and the provider allows the consumer to put constraints and expectation level from the cloud providers such as affinity and location.

The *measured service* characteristics of cloud enables it to control and optimize resources by metering capability at certain level of abstraction such as storage, bandwidth and processing. The controlling of the resources can be as per the agreement between the consumer and the provider. The resource usage can be monitored, controlled and reported providing transparency for the provider and the consumer.

2.10.2 Cloud Deployment Models

Deployment in cloud computing can take place in a number of ways as follows (Mahmood, 2011):

Private cloud: The cloud infrastructure is provisioned for exclusive use of by a single organization comprising multiple consumers.

Community cloud: The cloud infrastructure is provisioned for exclusive use of by a specific community of consumers from organizations that have shared concerns.

Public cloud: The cloud infrastructure is provisioned for open use by general public.

Hybrid cloud: The cloud infrastructure is composition of two or more distinct cloud infrastructures (private, community or public) that remain unique entities, but are bound together by standardized or proprietary technologies that enables data and application portability.

Hybrid cloud deployment models emerged are as follows:

Bursting: Cloud bursting (Nair *et al*., 2010) enables an organization to scale out their infrastructures by renting resources from third party provider if and when needed. The renting of the external resources furthers exponentially the elasticity of the company's IT infrastructure and lets them confront dynamically the fluctuations on demand.

Brokerage: Cloud service broker (Nair et al., 2010) creates a governed cloud management platform to simplify the delivery of complex cloud services to cloud service customers. It enables customers to realize the full potential that cloud provider has to offer and enforce the correct IT policies and effectively handle service level agreements between cloud provider and cloud service consumer.

2.11 Cloud Brokerage

Broker in general as an intermediary is very common and its use is also seen in various areas of computer science. Significant research exists in the area of brokers used in grid environment, and the grid service and resource brokers are amongst the most common ones (Venugopal et al., 2006)(Dumitrescu et al., 2005)(Gourlay et al., 2008). The works in (Gourlay et al., 2008) proposes broker architecture in grids with the focus on evaluating the reliability of the risk information from the resource providers.

Within the context of cloud, usage of brokers have been observed in (Nair et al., 2010)(P. S. Pawar et al., 2012)(Li et al., 2012). The cloud broker architecture in(P. S. Pawar et al., 2012) enables to gather security events to model security reputation of the cloud infrastructure as service providers. In (Nair et al., 2010), the proposed use of cloud broker is as 1) *cloud service intermediation:* intermediation for multiple services to add value-additions like identity management or access control, 2) *cloud service aggregation:* bringing together two or more fixed cloud based services, and 3) *cloud service arbitrage:* similar to cloud service aggregation, but providing a more dynamic aggregation to support flexibility. The work in (Nair et al., 2010) proposes the use of cloud broker at abstract level without any concrete architecture to realize the proposed functionalities. The work in this paper extends the work described in (Nair et al., 2010) wherein a broker architecture design and implementation is provided to realise the cloud service intermediation, aggregation, and arbitrage, together with the provision of value added services.

Recently in the cloud computing domain, cloud broker usage has increased specifically in the context of multi-cloud and inter-cloud envoironments.The topic of cloud brokerage with respect to inter-cloud operations has recently acquired great interest, but only a few efforts exist in this area. However, there are some standardisation efforts ("IEEE SA - CPWG/2301 WG - Cloud Profiles WG (CPWG) Working Group," n.d.)("IEEE SA - ICWG/2302 WG - Intercloud WG (ICWG)

Working Group," n.d.) underway in this field, but these have not been completed and made publicly available yet. Furthermore, it will be a big challenge for these standards to be accepted and implemented by the major cloud providers, which will be a necessary condition for their wider adoption.

2.12 Service Level Agreements

Service Level Agreement is an official agreement between service provider and service consumer which guarantee definite level of performance based on various quality aspects. As the growth of cloud usage increases which is driven by the growing requirements of cloud computing resources, there is critical need to provide Quality of Service performance guarantees for each of the service offered by the cloud providers. Performance and availability problems often impact the business and customers. Hence in the contract, the service provider agrees to guarantee a certain level of QoS and in return each business agrees to pay the service provider for satisfying these QoS guarantees in servicing its customers. These contracts are based on Service Level Agreement (SLA) between each business and service provider that defines the QoS guarantees for service, the cost model under which these guarantees will be satisfied and the anticipated level of requests from customers of the business (Liu et al., 2001).

A typical SLA Management framework supports SLA life-cycle that usually includes the following stages: (a) SLA establishment which involves SLA negotiation. (b) Service Provisioning that includes resources provisioning and service activation (c) Assurance which is in charge to monitor, validate and report the SLA, detect SLA violations and handle them (d) Assessment or testing involves, checking the satisfaction of its requirements for a customer, whereas, for an operator it is checking the overall service quality and key problems (Marilly et al., 2002).

Traditionally SLA establishment is performed through coordinated negotiation which can be viewed as the interaction among parties in the context of deriving mutual commitments. The negotiation begins with initial proposal that includes each party's goals and objectives and continues with the negotiation process which ends when both parties agree to a specified document (Demirkan et al., 2005). SLA negotiation can be particularly complex depending on the requirement and affordance of the two parties and may require to be carried out at runtime to minimize runtime interruption of the service based client (Di Modica et al., 2009). However runtime negotiations discussed in Di Modica et al are reactive, supporting corrective actions only after SLA violations and thus, they cannot ensure uninterrupted runtime when service fails (Di Modica et al., 2009). To minimize the runtime interruptions of the service based clients, the discovery of back up replacement service for the service based client should be performed proactively before any of these services become unavailable or fails to perform according to its established SLA. Proactive SLA negotiation is performed immediately after the execution of service discovery queries to ensure adequate SLAs are provisionally agreed for given period of time with providers of the discovered service if possible (Mahbub and Spanoudakis, 2011).

SLA Monitoring is another essential component of the SLA management framework. Monitoring plays an important role in determining whether SLA has been violated and from legal point of view, monitoring appears as a pre-requisite for contract enforcement. Moreover, Comuzzi et al. links the SLA negotiation and monitoring of a service based system and show that during negotiation, service providers require historical data from monitoring to evaluate SLA offers made by service consumers and argue that before an SLA is established, the capability to monitor terms at runtime must be confirmed (Comuzzi et al., 2009). Foster et al. shows how complex service agreement terms can be decomposed into manageable monitoring configurations while including mechanism to support preferred monitoring component selection requirement (Foster and Spanoudakis, 2011).

SLA violation of guarantee terms lead to the application of potential penalties that compels service provider to recognise the need to identify and detect possibilities of SLA violations. Monitoring is a useful technique to observe the behaviour with the aim of detecting SLA violations, however, this is a reactive approach which detects the problem after it has occurred. In critical scenarios it may not be recommended to wait until the problem occurs but provider must detect beforehand situations that may derive to SLA violations. Hence monitoring may not be enough and must be complemented with a proactive approach in order to detect SLA violations before they produce undesirable consequences. Palacios et al. (Palacios et al., 2010) presents an proactive method to generate test specifications from the information contained in the SLA that enables to uncover problems in provider and client software and detect potential violations of the SLA. Palacios et al (Palacios et al., 2015) presents a logic to evaluate the elements of SLAs which includes analysing information gathered from the monitors and checking the guarantee terms and finally making decisions about the fulfilment of such terms. In this context Palacious et al. proposes evaluation and testing of logical composition of guarantee terms in a service level agreement and defines four-valued logic that allows evaluating both individual guarantee terms and compositor elements(Palacios et al., 2015).

In the cloud computing environments, virtualization technologies enable users to specify required software such as operating system, software libraries, applications and computing resources such as CPU, memory, disk which are then packaged all together into virtual machines. QoS requirement can be formalized in Service Level

Agreement that serves as the foundation for the expected level of service between the cloud consumer and the service or cloud provider.

For a cloud infrastructure provider, over provisioning of resources to maximize Service Level Agreement results in poor resource management. Contrary to this, under provisioning of resources will increase SLA violations. Cloud IPs are required to meet QoS requirements of cloud services that require a different quantity of VM resources at run-time. Inappropriate resource allocation may result in resource waste and service quality degradation. Cloud infrastructure providers optimize on resource allocation such that there are minimum SLA violations, while maintaining high system utilization by avoiding over provisioning the VM resources to the services(Hasan and Huh, 2013).

Quality of Service(QoS) usually refers to the parameters that determine the performance of the service provided. The parameter of QoS also known as Key Performance Indicators(KPIs) are used to evaluate cloud computing services and it is very important to identify the correct metric to measure the QoS. Several studies have been submitted to define measurements for the KPIs in cloud environment. Several studies have produced measurements of SLA metrics with respect to QoS (Al-Shammari and Al-Yasiri, n.d.; Bardhan and Milojicic, 2012; Bruneo, 2014; Garg et al., 2013; Saravanan and Kantham, 2013; Shao and Wang, 2011; Xianrong Zheng et al., 2014).

Parameters	**Sample Values**
CPU cores	1, 2, 4, 8 etc.
Memory size	2GB, 4GB, 8GB etc.
Response time	10ms , 20ms etc.
Availability	100%, 99% , 95% etc.

Table 2.1 : Sample SLA parameters for a cloud infrastructure provider

Sample SLA parameter list for Infrastructure Provider is given in the Table 2.1:

Bardhan et al. presents QoS measurements in cloud environment that enables the cloud providers to not only prevent SLA violations but also optimize resource allocation by provisioning resources only when it is needed

OPTIMIS also implements the Service Level Agreement (SLA) framework that evaluates the state of the parameters specified in the service manifest(Ferrer et al., 2012; Rasheed et al., 2012). OPTIMIS project uses existing WSAG4J framework, implementing WS-Agreement and WS-Agreement negotiation and defines new term languages in the OGF (Open Grid Forum) for Trust, Risk, Eco-efficiency, Cost, Data Security, Data Protection and Security. Since the service manifest is part of the SLA in the OPTIMIs project, it creates contractual relationship between the consumer and cloud service provider. This allows the provider to plan its resource utilization and the commitments made to the consumer(Rasheed et al., 2012).

2.13 Trust in Cloud Computing

In cloud computing environment customers lack control of cloud resources and are not in good position to utilize technical mechanisms in order to protect their data against unauthorised access or secondary usage or other forms of misuses. Instead they have to rely on contracts or other trust mechanisms for appropriate usage in combination with mechanism that provide compensation in the event of breach such as insurance or penalties for breach of SLA (Pearson, 2013).

Despite of accelerated growth of cloud computing in the industry, trust management is still considered as one of the key challenges in the adoption of cloud computing. Trust has been extensively studied in environments such as: electronic market environment, peer-to-peer network, mutli-agent systems, grids, Service Oriented Architectures (SOA), where most of the current trust models are available (Halberstadt and Mui, 2001; He et al., 2009; Manuel et al., 2009; Maximilien and

Singh, 2004; Olmedilla et al., 2005; Resnick and Zeckhauser, 2002; Sabater and Sierra, 2002; Zhou and Hwang, 2007). However the trust models that are being proposed for these environments does not fully fit in cloud environment due to the essential cloud characteristics, the various deployment models and the parameters that have been taken into consideration in the trust models.

An effective trust management system helps cloud service providers and consumers to reap the benefits brought about by cloud computing technologies. However, several issues related to general trust assessment, distributed feedbacks, privacy of participants and lack of feedback integration which are still needed to be addressed for effective use in general and in cloud computing environment(Noor et al., 2013b). Specifically from cloud perspective, lack of consensus on what trust management approaches should be used, no suitable metrics of cloud are some of the issues to be addressed for effective use of trust management approaches in cloud environments. No suitable metrics exists for accountability and to date has only be considered at high level (Ko et al., 2011; Yao et al., 2010). There is no current consensus on the type of evidence required to verify the effectiveness of trust mechanisms. Although Cloud Trust Protocol (CTP) defines some categories, it has not covered others such as legal liabilities of parties involved (Pearson, 2013).

In literature, the most common classification for trust management techniques are given as: Policy, Recommendation and Reputation and these techniques are either applied to the service requester perspective (i.e. cloud service consumers perspective) or to the provider perspective (i.e. cloud service provider perspective) (Huang and Nicol, 2013; Noor et al., 2013b).

Policy as trust management techniques used in cloud environment uses set of policies each of which assumes several roles that control authorization levels and specifies a minimum trust threshold in order to authorize access. The trust thresholds are based on trust results or the credentials. Trust-result-based threshold approach such as monitoring and auditing, verifies Service-Level Agreement (SLA) violations in cloud services(i.e. if SLA is satisfied, then cloud service is considered as trustworthy and vice-versa) (Alhamad et al., 2010)(Noor et al., 2013b).

Recommendation as trust management technique is also used in the cloud environment. Trust is derived from recommendations using several operations including consensus (i.e. where trust feedback is aggregated from different cloud service consumers) and discounting (i.e. where trust feedback is weighted based on trustworthiness of cloud service consumers) (Habib et al., 2011; Jøsang, 2001; Jøsang et al., 2006).

Similarly there are efforts that use reputation as trust management techniques in cloud computing environments. Habib et al describes research trend on aggregating the reputation of particular cloud service based on feedback using QoS and other attributes such as geographical location(Habib et al., 2011). Noor et al. propose reputation based trust management framework that distinguishes the credible feedback from the misleading ones (Noor et al., 2013a).

Trust based on reputation systems for cloud environment has been discussed in (Alhamad et al., 2010; Ferrer et al., 2012; Hwang et al., 2009). Ferrer *et al*. considers trust as one of the core components used by SP (Service Provider), along with risk, eco-efficiency and cost for evaluating the IP (Infrastructure Provider) for their service (Ferrer et al., 2012). Hwang et al. (Hwang et al., 2009; Hwang and Li, 2010) identifies several vulnerabilities in the existing cloud service providers such as Google, IBM, and Amazon and proposes architecture to reinforce the security and privacy in the cloud applications. It suggests a hierarchy of peer-to-peer reputation system to protect cloud resources. To address the confidentiality and integrity of the client data in cloud, Santos et al. proposed a trust cloud computing platform(TCCP) (Santos et al., 2009) whereas to protect the information on the cloud environment, Kruatheim et al. proposes a trusted virtual environment module (TVEM) (Krautheim et al., 2010). Alhamad *et al*. (Alhamad et al., 2010) proposes a trust model for cloud computing based on the usage of SLA information whereas Brandic et al proposes an architecture and language support for user driven compliance management in clouds that assists in enactment and enforcement of compliance level agreements (Brandic et al., 2010). Noor et al. proposes "Trust as a Service" (TaaS) with an emphasis on credibility of trust feedbacks (Noor and Sheng, 2011).

2.14 Analysis of Trust based approaches

The policies allow to express, when, for what and even how to determine trust in an entity (Artz and Gil, 2007). The application of a policy is based on a set of information about entity regards to trust. Most common form of policy based trust is established using credentials which are usually certificates and rely on the trusted third party. Trust established using trust negotiation, tends to reveal credentials that may incur loss of privacy or control of information. Security policies consider how to represent trust. Policy specification for negotiating interactions is essential for building trust as the rules of negotiation determine how and if trust is achieved.

The other trust based approaches can be generalized into the following big classes: a) *Direct experiences*: experiences of the consumer with the service provider b) *Indirect experiences*: feedbacks/opinions from other consumers about the service provider c) Hybrid: combination of direct and indirect experiences.

Direct experiences are the best source of information that enables to establish the trust for any entity in the trust management system. However, in large distributed and unmanaged environments, very few or no direct experiences with many of the entities, may limit the trust evaluation that is only based on direct experiences.

In the absence of direct experiences, trust based on reputation, recommendation and social network, which is based on indirect experiences are used by the trustor to evaluate the trustworthiness of the trustee.

When direct experience is rare, the trust of an entity can be based on the opinion of the community. The reputation of the entity is based on the ratings collected from the members of the community. Reputation systems are prone to variety of attacks (Kerr and Cohen, 2009) and amongst the most common attacks are the ones performed by colluding malicious entities.

When direct experience is rare, the recommendations can provide trust evaluation of an entity. Since recommendations can influence the decision making process, selection of recommenders and weighting of the influence of the recommendation have to be done carefully. The critical issue is that recommenders may intentionally or accidentally provide misleading recommendations.

There are two approaches that deal with recommendations. The first is *Endogenou*s filtering or *exogenous* filtering (or discounting). In endogenous handling of recommendations, one can reduce the impact of misleading recommendations by considering the provided recommendations independent from recommenders. The second is misleading recommendations. These can be identified by statistical properties of the provided recommendations. Exogenous approaches consider information such as trustworthiness of the recommender. Additionally, social trust component (Sabater and Sierra, 2002) can also be considered.

Many approaches do not weight recommendation or weight recommendations based on entities behaviour as interaction partner. However the approaches proposed in (Jia et al., 2012) weighting recommendations according to the accuracy of the recommenders past recommendations, which seems a better choice but with the overhead of storing recommendations per recommender and per interaction partner. Recommenders rank is also being considered to improve the model's resistance to attacks.

The trust evaluation of an entity in the social network is formulated using the direct or indirect interaction occurring in the social network that can be in the form of information exchange or opinions or ratings. The global evaluation may not be necessarily computed by the member of the social network but might be the result of the centralized data/graph mining technique applied on the network. The trust computed for entity based on the interaction in the social network is value information about the entity under evaluation and can be independently used when direct evidence is not available.

In large distributed environment, trust computed only based on indirect experiences are prone to a variety of attacks not only in the recommendation systems but also in the reputation based systems as well as trust provided through social networks.

Integration of direct experiences with either, reputation, recommendation or social network based trust approaches can provide strong basis for trust evaluation of entity. Based on the confidence associated with the direct experience the trustor can weigh the trust based on reputation, recommendation and social network, to evaluate the trust of an entity. Also due to the massive growth of social networks, combined approaches based on indirect experiences such as reputation + social network or recommendation + social network are becoming popular and assists trust computation of entity with higher confidence.

2.15 Research Methodology

This section briefly describes the research methodology used in this thesis which includes research design approach, the primary study and the evaluation.

The primary objective of this research is to devise a trust model that is suitable for the cloud environment that allows selection of trustworthy cloud provider and which is robust against malicious feedbacks. The main hypothesis of this research was that trust models can be built for evaluating trustworthiness of Cloud entities to create a trusted environment within Cloud. To meet the objectives of the research we study various methods to analyse: trust in the cloud environment that may be involved considering different aspects of the cloud providers; the interactions between interconnected cloud entities; the information of interaction that may be valuable sources of information to be considered for the trust models.

A literature survey is conducted to collect research publications and other documents, for understanding the defined problem related to trust in general and more specifically in cloud environment. Literature survey for trust in cloud environment leads to some of the crucial requirements that are essential to evaluate the cloud provider's trustworthiness which are given as: a) An independent mediation layer capable of performing a variety of trust assessment, is needed to evaluate the service providers b) An evaluation framework that is trusted enough such that malicious providers cannot manipulate the evaluation process c) Cloud service providers should be evaluated based on fine-grained QoS parameters together with consumer feedbacks, recommendation and further specific parameters related to the cloud computing environments (Habib et al., 2010). The thesis proposes trust assessment of the cloud service providers with the use of the Cloud Broker (CBR) architecture that assists in this evaluation. The trust model cohesively works along with the cloud broker in different settings to evaluate the trustworthiness of the cloud service providers. Several trust models are available which are used in environments such as electronic market, grids etc. However the analysis show that trust models based belief are highly suitable due to their capability to model uncertainty. This encourages building a trust model for cloud environment based on the belief functions and incorporates credibility methods and filtering mechanism for robustness of the trust model. This trust model is supported with the enhanced opinion model that considers belief, disbelief, and uncertainty. The trust framework uses subjective logic operators to combine evidences from different sources. The sources of information in cloud environment, for the trust model is studied and parameters such as SLA information, service provider feedback are considered crucial for obtaining satisfactory results in evaluating the trustworthiness of the cloud infrastructure provider.

The opinion defined in the proposed trust model is evaluated with real data from Amazon market place as well as the entire trust model is evaluated building a simulated environment. The subset of data used from the Amazon market place is good representative of a real environment for testing trust models. The evaluation of the opinion model and comparison with existing models verifies that the proposed opinion model has enhanced accuracy over the other models. The evaluation of the trust model in the simulated environment shows that the credibility and filtering mechanisms are very effective to resist malicious feedbacks to provide proper trust results within the cloud environment. The trust model proposed for the cloud environment and its evaluation directs in accepting the initial hypothesis that trust models can be built for evaluating trustworthiness of Cloud entities to create a trusted environment within Cloud.

2.16 Summary

This chapter describes in detail the study of literature available in trust, reputation and recommendation systems that has been performed to gain understanding of the subject area. The report also describes topics in cloud computing environment that are required as background knowledge for the understanding of this research. This chapter also brief discussed on the trust concepts used in the area of cloud computing.

As described in this chapter, representational model defines how trust is represented and there exists different representations of the trust that exists in the literature of which multi-dimensional representations are very common. For the trust model defined in this thesis, we also adopt the multi-dimensional belief based representation wherein it is also important to capture uncertainty appropriately to model the more accurate belief. The opinion model in the proposed trust model is represented in terms of belief, disbelief and uncertainty, uniquely capturing the uncertainty to enhance the accuracy of the trust model.

In this thesis, to be resistant to the misleading recommendations, we consider both the *endogenou*s filtering and *exogenous filtering* to be included in the trust framework and the inclusion of social trust is planned as future work. The endogenous handling of recommendations is done by using the outlier detection mechanism in (Arning et al., 1996; Zhang and Feng, 2009) to detect unfair ratings and filter these ratings to reduce the impact on reputation due to unfair ratings. The exogenous handling of recommendation is performed by including the credibility model that computes credibility associated with each recommender.

Most existing trust models consider the transaction life to be small and assess the trust or reputation of an entity purely based on the historical transactions. However, in the cloud environment the current transactions being active for longer durations, inspires us to incorporate performance of these transactions to evaluate the trustworthiness of the entity. SLA (Service level agreements) provides crucial information about the active transactions in terms of violations with the current agreements that is used as information to evaluate the trustworthiness. The trust model proposed in this thesis includes SLA compliance information to model trust and also complements the trust model with SP (Service Provider) ratings and SP (Service Provider) behaviour to assist modelling.

The literature review on the trust models for cloud indicates that there are very few trust models that exists tailored for cloud environment and none of these models capture the wider scope of the cloud environment. In this thesis, we propose a trust model that comprehensively captures the essential cloud characteristics to evaluate the trustworthiness of cloud entities.

As described in this chapter, there are different basic deployment models that exist and advanced deployment models such as cloud bursting, cloud brokerage and cloud federation that are being proposed recently. In this thesis, we also propose a cloud brokering architecture that is used as the use case scenario for the evaluation of the proposed trust model. Also this cloud broker architecture is used to support for evaluating security reputation of the cloud providers.

Chapter 3 Cloud Broker and Trust Assessment

Despite the advantages and rapid growth of Cloud computing, the cloud environments are still not sufficiently trustworthy from customer's perspective. The emerging cloud market, introduces multitude of cloud service providers that complicates the decision of consumers to select providers that are trustworthy for its service. Several challenges such as specification of service level agreements, standards, security measures, selection of service providers and computation of trust still persists that concerns the customer. To deal with these challenges and provide a trustworthy environment, a mediation layer may be essential. In this chapter we propose a cloud broker as a mediation layer, to deal with complex decision of selecting trustworthy cloud service provider that fulfils the service requirements, create agreements and also provision security. The cloud broker operates in different modes and this enables a variety of trust assessments. Cloud broker used as cloud service recommendation allows computation of trust based on resource requirements while cloud broker used as cloud service intermediation supplements with the trust based on value added service such as security service. Cloud broker used as cloud service aggregation/arbitrage allows computation of trust for a multi-cloud deployment of a service.

As briefed in Chapter 1, in this thesis we propose uncertainty based trust model supported with credibility model for evaluating cloud service providers. More details about the trust model are available in Chapter 4 & Chapter 5.

In this chapter we introduce the different modes of Cloud Broker that enables a variety of trust evaluations. While this chapter briefly describes the different modes of cloud broker, a more detailed architecture of the cloud broker is available in Chapter 6.

3.1 Introduction

Organizations are beginning to realize the economic advantages of cloud computing and are increasingly turning to cloud services. Despite this cloud-friendly shift in thinking, most organizations still continue with their concerns about trust and security of cloud infrastructures. Several challenges such as specification of SLAs, standards, security measures, selection of service providers and computation of trust still persists, depicting that the cloud environments are still not sufficiently trustworthy from customer's perspective(Habib et al., 2010). To deal with the challenge of identifying dependable cloud service providers for the service, cloud market places are gaining popularity. The marketplaces enable the cloud providers to publish their services and the end users to select the services. The market place either belongs to a single provider such as Amazon ("Amazon Web Services," n.d.) or an open market place exists that allow resources published by multiple cloud service providers (Zhao et al., 2012). Marketplaces are supported with the application management capabilities such as performance monitoring and billing. Marketplaces allow the consumers to select the resources as per their requirements and accordingly select the providers that best match their requirements. CloudBay (Zhao et al., 2012) assist sellers and buyers by offering a comprehensive solution for resource advertising and connect of transaction management and application with infrastructure mapping. The complex requirements and these multiple choices available to the consumer make it difficult to decide on a provider to host their service. In addition their concern on the trustworthiness of the providers remains unanswered.

Service Level Agreement (SLA) is a crucial parameter to assess trustworthiness of a cloud provider, however lack of standards in the SLA formats and content across the cloud providers, creates constraints in the process of selecting trustworthy cloud provider. Moreover, the cloud characteristics (Mell and Grance, 2011) such as elasticity and the complex deployment models like multi-cloud and federated clouds create major challenges in the assessment of trustworthy cloud providers. A unanimous trust assessment across all deployment architectures may not be suitable and creates compelling requirements for having separate trust assessments suitable for deployment architecture.

The assessment of the cloud computing environment leads to some of the crucial requirements that are essential to evaluate the cloud provider's trustworthiness. These are: (a) An independent mediation layer capable of performing a variety of trust assessment, is needed to evaluate the service providers, (b) An evaluation framework that is trusted enough such that malicious providers cannot manipulate the evaluation process, and (c) Cloud service providers should be evaluated based on fine-grained QoS parameters together with consumer feedbacks, recommendation and further specific parameters related to the cloud computing environments (Habib et al., 2010).

Due to the complexity of service requirements and difficulty of trustworthiness evaluation of the cloud providers, third parties like cloud brokers can play important role to assist the consumer in selecting an appropriate provider and also assist in deployment of the service. Sundareswaran et al. (Sundareswaran et al., 2012) propose a cloud broker based architecture that enables selecting and ranking the cloud service providers, however the architecture supports encoding techniques that captures similarity among the providers and does not provide support for negotiating SLA terms.

The main focus of this chapter is to propose the trust assessment of the cloud service providers with the use of the Cloud Broker (CBR) architecture that assists in this evaluation. In this chapter we propose the use of trust model that cohesively works along with the cloud broker in different settings to evaluate the trustworthiness of the cloud service providers. The use of proposed cloud broker architecture aids in obtaining solutions towards some of the research challenges described above. The proposed trust model is described in Chapter 4 & Chapter 5 which considers SLA parameters and the cloud characteristic parameters for evaluating the trustworthiness of the providers and is robust against malicious group of entities performing reputation based attacks.

Chapter 6 proposes a Cloud Broker architecture that can operate in different modes. In comparison with (Zhao et al., 2012), the proposed cloud broker architecture in, Chapter 6 supports mapping of application-to-infrastructure mapping and automatic networking configurations across multiple providers. Additionally, the cloud broker also provides support for matching of consumer requirements, establishing agreements and providing value added services such as security to the consumers. In addition, the cloud broker also performs trust evaluation of the cloud service providers. The mediation layer of cloud broker allows trustworthy selection of cloud providers and the service management functionality including security that eases the burden of the consumer and creates sufficient trust in the cloud environment.

The main actors of the system used in this research are shown in Figure 3.1 and described below (Hogan et al., 2011):

- Infrastructure Provider (IP): Infrastructure Providers are organisations making cloud infrastructures available to the Service Providers (SP), Cloud Brokers(CBR) and Users. IP provisions and manages the physical resources such as compute, storage, networking and the hosting environment and cloud infrastructure for IaaS consumers.
- Service Provider (SP): Entities that use the cloud infrastructures provided by IPs and making service available to the Users. SP installs, manages, maintains and supports the software applications on a cloud infrastructure; provides development and administration tools to platform consumers; provides software and platform services for SaaS and PaaS consumers.
- Cloud Broker (CBR): Entities that manages the use, performance and delivery of cloud services and negotiates relationships between IPs and SPs. As cloud computing evolves, integration of cloud services can be too complex for SP and cloud users to manage. An SP or user may request cloud services from a cloud broker, instead of contacting the cloud provider directly.
- User (U): Entities that maintain business relationship with and uses services from IP and SP. The cloud user is the ultimate stakeholder that the cloud computing service is created to support.

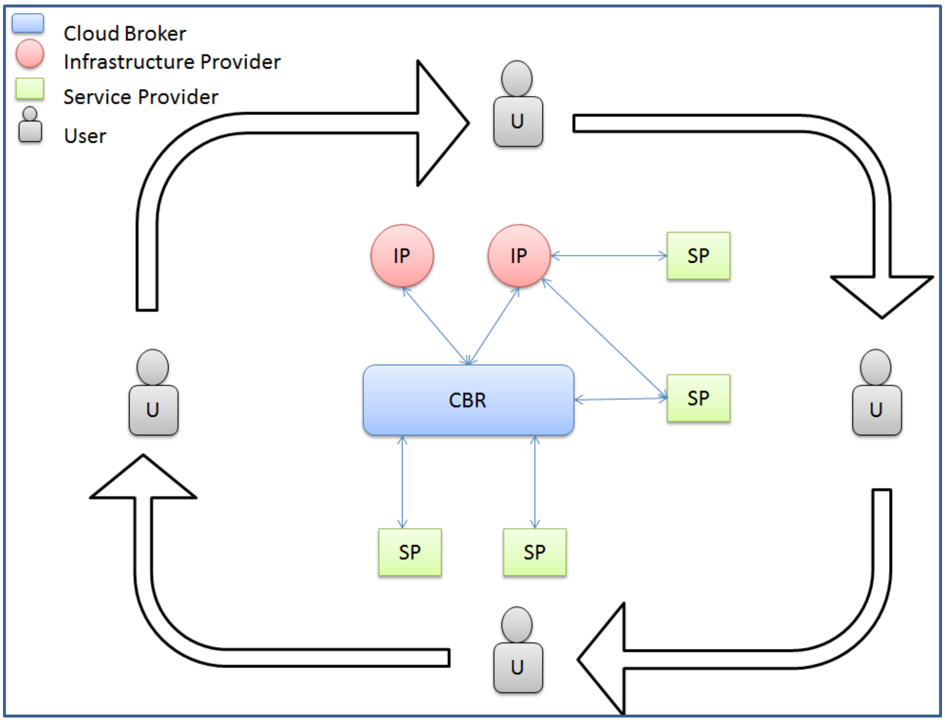


Figure 3.1: Main actors used in this research

The remaining Chapter is structured as follows. Section 3.2 briefly describes the different modes of operation for the cloud broker. Section 3.3 describes the type of trust evaluation in each of these cloud broker modes. Finally section 3.4 provides the concluding remarks

3.2 Cloud Broker Service

This section proposes the cloud broker to be used as 1) *cloud service recommendation* 2) *cloud service intermediation* 3) *cloud service aggregation* and 4) *cloud service arbitrage.*

3.2.1 Cloud Service Recommendation

CBR(Cloud Broker) used in *cloud service recommendation* mode enables the consumer/user to get recommendations from the CBR about the most suitable cloud infrastructure provider for hosting their service, based on the degree of Trust, Risk, Eco-efficiency and Cost (TREC) (Kiran et al., 2011; P.S. Pawar et al., 2012). However, this thesis considers evaluation only based on trust. The CBR as a *recommender* reduces the effort of the consumer to identify the suitable cloud service provider for its service, but the actual deployment of the service to the cloud infrastructure is performed by the consumer after obtaining the deployment solution from the CBR.

3.2.2 Cloud Service Intermediation

CBR used as *cloud service intermediation* provides management functionalities like *Value Added Services (VAS)* that are cloud provider specific and which may be essential for the consumer's service that is deployed in the cloud provider environment. Examples of VAS for security can include VPN, secure storage, and intrusion detection system. These services are provisioned as VAS in the OPTIMIS where the broker architecture proposed in this thesis is implemented, however, the services are beyond the scope of this thesis("Optimis - Optimized Infrastructure Services," n.d.). As an *intermediary*, the CBR also takes complete responsibility of the consumer's/user's services to identify the most suitable IP based on TREC, performs the deployment on the selected IP, and manages smooth functioning of the service in its operational stage.

3.2.3 Cloud Service Aggregation

The use of CBR as *cloud service aggregation* provides management functionalities for *multi-cloud* deployment and operation of a service by combining the multiple cloud infrastructure provider services. The CBR also provides VASs that are independent of cloud providers. The multi-cloud deployment capabilities of the cloud broker proposed in this thesis is implemented in OPITMIS and the thesis provides the performance evaluation of this architecture with a multi-cloud deployment scenario.

3.2.4 Cloud Service Arbitrage

CBR used as *cloud service arbitrage* can be considered as dynamic aggregation wherein the multi-cloud deployment of consumer service is dynamically decided based on the service requirements. In this mode of operation, the cloud broker system decompose the service requirements at component level and negotiates with multiple cloud providers for each of the service components to formulate an optimized deployment solution taking into account the basic service requirements as well as additional requirements such as trust, risk, eco-efficiency, cost, compliance and security.

3.3 Trust model

The trust assessment could be performed using the different modes of cloud broker. Figure 3.2 depicts the different modes and the corresponding trust assessments.

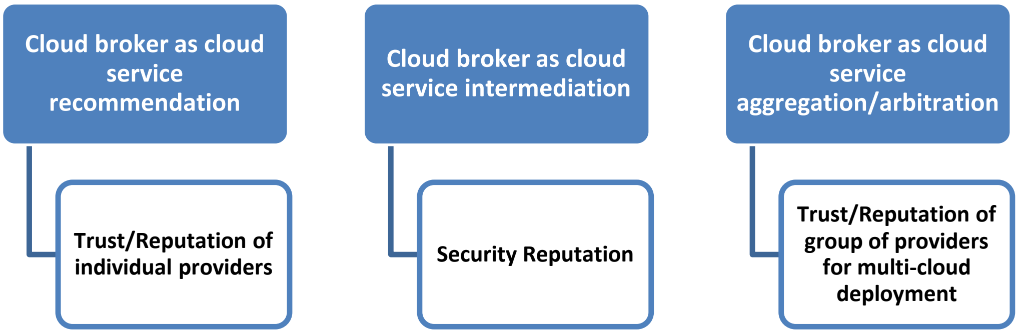

Cloud broker as cloud service recommendation
Trust/Reputation of individual providers
Cloud broker as cloud service intermediation
Security Reputation
Cloud broker as cloud service aggregation/arbitration
Trust/Reputation of group of providers for multi-cloud deployment

Figure 3.2: Trust evaluations in different modes of cloud broker

The cloud broker uses the trust assessment results for the deployment of a new service as well as during the service in operation and the cloud broker is responsible for data monitoring in service operation. For the broker to perform trust assessment of cloud infrastructure providers, the cloud broker expects to get feedback information from the SPs as well as data from IPs which is agreed in the SLAs.

3.3.1 Cloud Broker as Cloud Service Recommendation

In this mode of operation the cloud broker is approached by the consumers or Service Providers (SP) for providing the trustworthiness of the cloud Infrastructure Provider (IP). The cloud broker uses the trust model for cloud environment as proposed in Chapter 4 & Chapter 5.

The *Trustworthiness* of an IP is modelled using *opinion* obtained from three different computations, namely: (i) *compliance of SLA parameters (SLA monitoring),* (ii) *service provider satisfaction ratings (SP ratings)*, and (iii) *service provider behaviour (SP behaviour).*

> **SLA Monitoring:** The SLA monitoring determines the opinion about an IP from the SLAs that the IP have established with the SPs for their services. The SP for each of its service has a single SLA that includes several indicators (e.g. CPU, memory, disk space, number of virtual machines (VMs)). For each indicator of an SLA, there is an associated monitor that evaluates the compliance/non-compliance of the indicator.
>
> **SP Behaviour.** The SP behaviour is defined in terms of the number of times the SP has used the infrastructure of an IP against the SPs total usage. An SP using a single IP for the majority of the times indicates the SPs good

behaviour towards an IP. The SP may use the infrastructure of an IP for one or more indicators specified in the SLA.

SP ratings: The service provider satisfaction rating is calculated based on the rates of the services given by an SP using an IP. The SP provides separate ratings for each SLA indicators of the IP's services. The ratings are used to form an opinion about an IP.

The proposed opinion based trust model is supported with credibility model complimented with early filtering to reduce the impact of malicious feedback providers

The cloud broker uses this trust model to provide recommendations about the cloud service provider based on the consumer requirements.

3.3.2 Cloud Broker as Cloud Service Intermediation

The cloud broker in the intermediary mode of operation, have capabilities of provisioning security value added services. This enables the broker to have access to most of the security events that enables it to perform security reputation of the cloud service providers.

Chapter 6 proposes the use of cloud broker architecture that enables gathering security events required for trust evaluation of the cloud provider based on its security capabilities. The cloud broker uses the trust model described in Chapter 4 & Chapter 5 for security based trust evaluation of the cloud providers.

The reputation of a cloud service provider is calculated in terms of its *trustworthiness (T)* using opinion obtained from computations, namely i) *Incidence Monitoring (M)*: Security incidence events received from monitoring ii) *Service Provider Rating (SPR)*: Ratings provided by the sevice provider for satisfaction of the security features provided by cloud service providers. The *trustworthiness (T)* is given by combining opinions obtained from each of these computations and then calculating the expectation of the combined opinion.

a. Incidence Monitoring

The incidence monitoring records evidence about the incidences related to parameters such as authentication, authorization, inside attacks, multi-tenent attack, data leakage and malware propogation. These incidences can either be identified by the cloud infrastructure provider and sent to the broker or the broker after receiving the security events carries further analysis to identify the incidences from the data received

b. Service Provider Rating

For every usage of the services from the Cloud Infrastructure Provider (IP), the service provider rates the satisfaction of security features and capabilities provided by the IP corresponding to the requirements set forward initially by the SP. Service providers register with the cloud broker and provide ratings about the IP for each of the monitoring parameters and the opinion for IP is formulated based on the service provider rating.

c. Trust of Infrastructure Provider

The *trustworthiness (T)* of the cloud infrastructure provider (IP) is given by combining the opinions W_M and W_{EUR} given by incidence *monitoring* and the *Service Provider* respectively. Where W_M is the opinion formed based on monitoring the incidents of authentication, authorization, inside attacks, multi-tenant attacks, data leakage and malware propagation. W_{EUR} is the opinion formed based on the service provider ratings for the satisfaction of security features provided by the IP.

3.3.3 Cloud Broker as Cloud Service Aggregation/Arbitration

The cloud broker used as cloud service aggregation/arbitration is with capabilities of devising multi-cloud deployment solution based on the user requirements. This enables the cloud broker to also assess trust assessment that can be performed on the group of providers obtained as multi-cloud solutions.

3.4 Trust management prototypes comparison

The Table 3.1 shows the comparison of the proposed trust management framework with representative trust management research prototypes specifically built for cloud computing environment. The recent survey of trust management services in cloud environment presents various trust management prototypes and thirteen different criteria for comparing the prototypes(Noor et al., 2013b). The same criteria are used in this thesis to compare the proposed trust management framework with the trust management prototypes available in the area of cloud computing.

The thirteen criteria identified represent comprehensiveness of functionalities available in the research prototypes and helps to compare the trust management framework in this thesis with the available trust management prototypes. The Table 3.1 shows that the trust management framework proposed in this thesis is sufficiently comprehensive with its functionalities provided. However the novelty of this research also lies in various other aspects of the trust management framework and the trust model developed in this thesis.

The proposed trust model for the cloud environment considers in progress transaction information in terms of SLA (Service Level Agreement) violations to model the trustworthiness of the cloud providers. This trust model is supported with the proposed uncertainty model that is used in the representation of the opinion.

Evaluation of this opinion model representation provides significant enhancements over existing trust models contributing to high accuracy provided by the trust model.

The proposed trust model incorporates the essential cloud characteristic as dimensions to evaluate the cloud infrastructure provider and also incorporates credibility and filter for malicious feedbacks. The credibility model in this thesis is evaluated to demonstrate an effective mechanism to resist malicious feedback providers. The filtering mechanism proposed is also demonstrated to be very effective against the malicious feedbacks. This thesis evaluates the effect of combined credibility and filtering mechanism in the trust model and shows a significant improvement to resist malicious feedbacks and feedback providers thus contributing to the security dimension by presenting the degree of robustness against malicious behaviours and attacks.

Beyond the trust model, the thesis also defines a trust assessment framework and architecture in the form of mediation layer of cloud broker that assist trust assessment in the cloud environment. The thesis presents variety of trust assessments possible using the different modes of cloud broker such as: trust assessment of individual cloud IP, security reputation of cloud IP and trust assessment of group of cloud IPs. This thesis provides a detailed architecture of the mediation layer such as a cloud broker and presents a mechanism for assessing the security reputation of the cloud IP.

Prototypes	TFSL				TAL						TRDL			
	Credibility	Privacy	Personalization	Integration	Perspective	Technique	Adaptability	Security	Scalability	Applicability	Response Time	Redundancy	Accuracy	Security
(P. S. Pawar et al., 2012)	FC\EC	SR	P	SFC	SRP	PocT\RepT\RecT	P	AFL	D	IaaS	NAT	N	P	N
(Ko et al., 2011)	EC	SR	N	NFC	SRP	PocT	N	AFL\CL	C	IaaS	NAT	N	F	ACL\CL
(Habib et al., 2011)	EC	N	P	SFC	SRP	RecT\RepT\PrdT	P	AFL\CL	C	All	NAT	N	F	ACL\CL
(Noor and Sheng, 2011)	FC\EC	SR	P	NFC	SRP	RepT\PrdT	F	AFL\CL	D	All	NAT	TR	F	ACL\CL
(Krautheim et al., 2010)	EC	SR	N	SFC	SRP\SPP	RecT\RepT	N	CL	C	IaaS	NAT	N	P	ACL\CL
(Brandic et al., 2010)	EC	SR	P	NFC	SRP	PocT	P	CL	C	IaaS\PasS	NAT	N	P	ACL\CL
(Yao et al., 2010)	EC	N	N	NFC	SRP	PocT	P	CL	C	IaaS	SAT	N	P	ACL\CL
(Hwang et al., 2009)	EC	SR	N	NFC	SRP	PocT	N	AFL\CL	C	All	SAT	N	F	ACL\CL
(Santos et al., 2009)	EC	SR	N	NFC	SRP	PocT	N	CL	D	IaaS	NAT	TR	P	ACL\CL
(Manuel et al., 2009)	FC\EC	SR	N	SFC	SRP	PocT\RepT	N	AFL\CL	C	All	SAT	N	F	ACL\CL
(Alhamad et al., 2010)	EC	SR	P	SFC	SRP	PocT\RepT	N	N	D	IaaS	SAT	N	P	N

Trust Feedbacks Sharing Layer (TFS1)

Credibility	**Privacy**	**Personalization**	**Integration**
FC Feedback Credibility EC Entity's Credibility N None	SP Focus on Service Provider's Privacy SR Focus on Service Request Privacy N None	F Full P Partial N None	SFC Strong use of feedbacks combination NFC No strong use of feedbacks combination

Trust Assessment Layer (TAL)

Perspective	**Technique**	**Adaptability**	**Security**	**Scalability**	**Applicability**

<table>
<tr><td>SPP Service Provider Perspective
SRP Service requester Perspective</td><td>PocT Policy Technique
RecT Recommandation Technique
RepT Reputation Technique
PrdT Prediction Technique</td><td>F Full
P Partial
N None</td><td>AFL Support Assessment Function level
CL Support Communication level
N None</td><td>C Centralized
D Decentralized</td><td>IaaS Infrastructure as a Service
PaaS Platform as a Service
SaaS Software as a Service
All All three models</td></tr>
<tr><td colspan="6"></td></tr>
<tr><td colspan="6"><b>Trust Result Distribution Layer (TRDL)</b></td></tr>
<tr><td colspan="2"><b>Response Time</b></td><td colspan="2"><b>Redundancy</b></td><td><b>Accuracy</b></td><td><b>Security</b></td></tr>
<tr><td colspan="2">SAT Strong Emphasis of Assessment Time
NAT No Strong Emphasis of Assessment Time</td><td colspan="2">AR Support Assessment Redundancy
TR Support Trust Data Redundancy
N None</td><td>F Full
P Partial
N None</td><td>ACL Support Access Control level
CL Support Communication level
N None</td></tr>
</table>

Table 3.1 : Comparison of trust management research prototypes for cloud environment

3.5 Conclusion

This chapter proposes the use of cloud broker and its various modes to perform variety of trust evaluations of the cloud service providers. The trust model used by the cloud broker is proposed in Chapter 4 & Chapter 5. The cloud broker as cloud service recommender uses the trust model proposed in Chapter 4 & Chapter 5 for providing recommendations to the cloud service consumers. The detailed architecture of the cloud broker and its features is described in Chapter 6. The security value added service provisioned by cloud broker as cloud service intermediation uses the trust model proposed in Chapter 4 & Chapter 5 for evaluating security reputation of the cloud infrastructure provider. The cloud broker architecture proposed has the capability to be used as cloud service aggregation/arbitrage to provide multi-cloud solutions. This enables the cloud broker to use the trust model to provide trust for group of providers.

Chapter 4 Trust Model for Cloud Services

In this chapter, we propose a trust model which computes the trustworthiness of cloud infrastructure provider. This model is mainly based on the reputation-based trust that model's the trust of cloud service providers based on available evidence. Many existing reputation based systems either ignore or give less importance to uncertainty linked to the evidence. In this chapter, we develop an uncertainty model and define our approach to compute the opinion for cloud service providers. Using subjective logic operators along with the computed opinion values, we propose mechanisms to calculate the reputation of cloud service providers.

4.1 Introduction

Trust is an important concept for cloud computing given the need for consumers in the cloud to select cost effective, trustworthy, and less risky services (Alhamad *et al.*, 2010). Entities such as Service Providers(SP),which offer services to the end users, are consumers of the cloud infrastructers, provided by Infrastructure Providers (IP). The issue of trust is important for Service Providers (SP) to decide on the Infrastructure Provider (IP) that can comply with their needs, and to verify if the infrastructure providers maintain their agreements during service deployment.

This chapter describes a trust model to support service providers (SP) to verify trustworthiness of the infrastructure providers (IP) during deployment and operational phases of the services supplied by the service providers.

The aim of the Service Provider (SP) is to offer efficient services to its customers using resources of the Infrastructure Provider (IP). The IP aims to maximize its profit by efficiently utilizing its infrastructure resources subject to good service to the SP and meeting all its requirements. The trust framework is active during the service deployment and service operation phases. The trustworthiness of the IP and the SP are monitored during these two phases of the service life cycle.

This chapter proposes a trust model mainly to evaluate the trustworthiness of the IP performed by the SP. During the *service deployment phase,* the objective of the SP is to select the most suitable IP for hosting its service based on the degree of trust expected from an IP. During the *service operation phase*, the SP monitors the IP's trust level and takes corrective actions. An example of an action is to select an alternative IP when the trust level of the IP is unacceptable, based on a negotiated trust level.

The trust model described in this chapter calculates trust values based on three different parameters, namely (i) *compliance of SLA parameters* (e.g., when the IP fulfils the quality aspect specified in the SLA between an SP and the IP), (ii) *service and infrastructure providers satisfaction ratings* (e.g., when SP supplies a rating for the IP where the SP is being deployed), and (iii) *service and infrastructure provider behaviour* (e.g., if the SP continues to choose the same IP independent of the rating that it has supplied for the IP). In the model, the satisfaction values can be either explicitly provided in terms of ranking measurements, or inferred based on relationships between the service and infrastructure providers, and behavior of the providers in terms of constant use of services, service providers, and infrastructure providers.

For each of the different parameters above, trust values are calculated based on an opinion model that considers *belief*, *disbelief*, and *uncertainty* values(Jøsang, 2001). The work on opinion model and subjective logic by Jøsang uses the element from Dempster-Shafer theory and is compatible with binary logic and probability calculus (Jøsang, 2001). Jøsang (Jøsang, 2001) defines uncertainty in a heuristic manner considering the amount of evidence, such that the uncertainty increases if the amount of evidence decreases and vice-versa. We have developed an opinion model that considers certainty when computing *belief* and *disbelief* values and is based on the extension of Josang's opinion model (Jøsang, 2001). In our model certainty is considered based on the amount of evidence and on the dominance that exist between the positive and negative evidences. If the number of positive (belief) evidences is closer to the number of negative (disbelief) evidences, the certainty about the proposition decreases and the uncertainty (i.e. one minus certainty) increases. For example, as the negative evidence (number of times IP1 violates a quality property) approaches to the positive evidence (number of times IP1 does not violate the same property), the level of certainty (of IP1 for that property) decreases.

As in our model, Wang et al.(Wang and Singh, 2010) also consider uncertainty to compute belief and disbelief. Wang et al. define certainty as a *Probability Certainty Density Function (PCDF)* which is probability density function of the probability of positive experience. With no knowledge (or evidence), the uniform distribution has certainty of zero.As the knowledge increases, the probability mass shifts deviating from the uniform distribution and increasing the certainty towards one. However, our approach is based on modelling uncertainties expressed in the form of *confidence ellipses* which is based on well recognised technique used to determine zones of uncertainty in surveying, navigation, and position location systems (Hoover and Rockville, 1984).

The remainder of this chapter is orgonised as follows: Section 4.2 presents an example that will be used throughout the chapter to illustrate the work. Section 4.3 describes the trust model used by the framework. Finally, Section 4.4 provides concluding remarks and future work.

4.2 Cloud Computing Example Scenario

In order to illustrate the work described in this chapter, we present a cloud computing *education application* that is being deployed for Bristish Telecom (BT) customers such as Universities and other educational institutions. The education application allows Universities and educational institutions to have virtual laboratory environments for students, staff, and all other members of the institutions. The application is hosted in the BT cloud and members can access the application via the Internet using their local desktops, and servers.

The key features of the application includes: i) flexibility to work from anywhere and anytime allowing the users to access the desktop and corporate applications from any PC, MAC, thin client or smartphone; ii) reduction of desktop management cost enabling the IT department to add, update, and remove applications in an easy way; iii) provision of good data security, good access control, and scalable storage platforms; iv) provision of scalability and elasticity for computing resources; v) comprehensive monitoring and management to support use and capacity planning and space usage; and vi) backup and recovery functions. The application has several components, namley: web interface, active directory, Desktop Delivery Controller (DDC), Virtual Machines (VM), and storage. The web interface passes user credentials to DDC, which authenticates users against the active directory. The VM is a virtual desktop accessed by the end users after receiving the connection details.

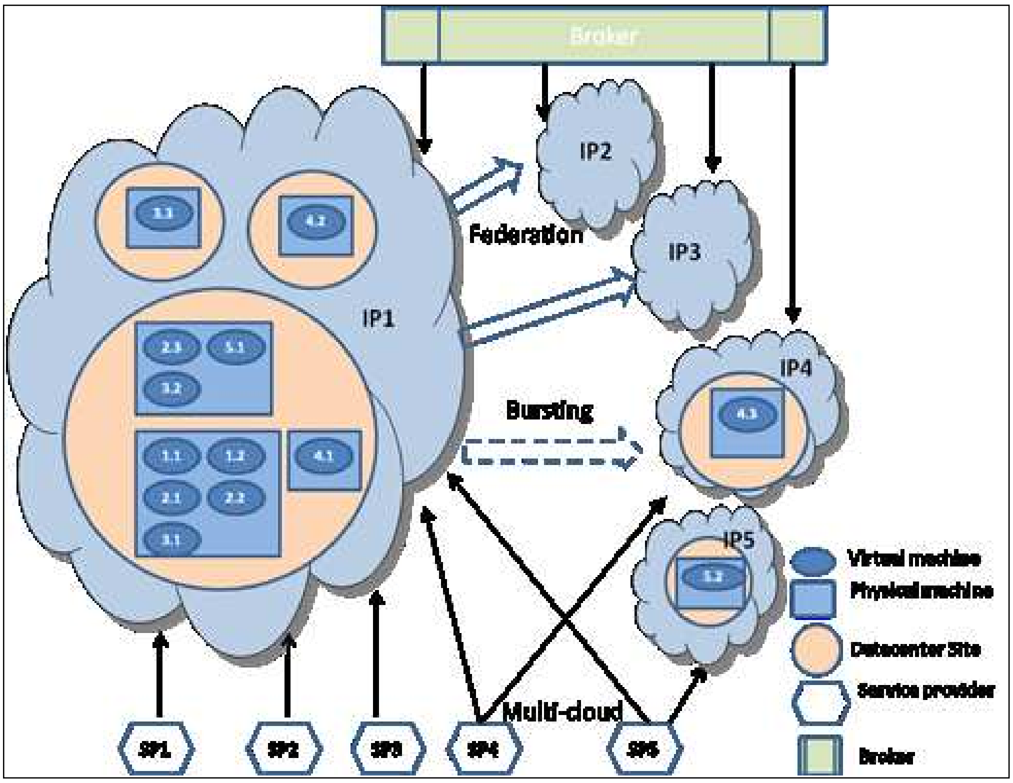


Figure 4.1: Cloud computing educational application example

For evaluating our proposed model an education application is considered with five SPs and five IPs. An SP hosts the application with its multiple components either at one IP or at multiple IPs. The SP may also use a broker for the IP services. This example scenario considers that all the SPs host education application in the Cloud environment. Figure 4.1 shows the education application deployed by various SPs. As shown in the figure, each IP has multiple datacenter sites which may be geographically distributed. Each of these datacenters can have a large number of physical hosts/machines available with capabilities to execute multiple virtual machines.

The three datacenters of IP1 are composed of three, one, and one physical hosts, respectively. The IP1's datacenter with three physical hosts deploy five, three and one virtual machines, respectively. Figure 4.1 shows that IP1 is in a federation with IP2 and IP3. In this case, IP1 is capable of leasing capacity from IP2 and IP3. Figure 4.1 also shows a situation of a bursting scenario, in which the organizations can scaleout their infrastructures and rent resources from third parties, as and when its is necessary. For example, as shown in Figure 4.1, infrastructure provider IP1 may burst to infrastructure provider IP4 to meet the SLA requirements of any SP. Figure 4.1 also shows the brokers that are associated with the IPs and are capable of renting infrastructure resources from all the IPs. Figure 4.1 indicates that the SPs have deployed the application in the cloud environment with different constraints (options), as described below.

Option 1: The application is deployed at a single IP, with a constraint of having all components of the application on the same host. SP1 in the figure have all its virtual machines (VM1.1, VM1.2, and VM1.3) running on a single physical host of IP1.

Option 2: The application is deployed in a single datacenter of an IP. SP1 and SP2 have all its virtual machines running on the same datacenter of IP1.

Option 3: The application is deployed in a single IP’s administration boundary (restrict usage of federation resources). SP1, SP2 and SP3 have all its virtual machines in the administration boundaries of IP1.

Option 4: The application is deployed in more than one IP. SP4 and SP5 deploy the application in IP1, IP4 and IP1, and IP5, respectively.

Several other deployment scenarios such as multi-cloud combined with federation, multi-cloud through broker and various combinations of cloud broker, multi-cloud, federation and bursting are possible, however for illustrative purpose, we will concentrate on the above situations. Although Figure 4.1 shows that SP1, SP2 and

SP3 have currently deployed applications on the infrastructures of only IP1, it is possible that they may have used other IPs (IP2, IP3, IP4 and IP5) in the past. Similarly, SP4 have currently deployed application components on IP1 and IP4, and SP5 have deployed application components deployed on IP1 and IP5, however they may have used other IPs in the past.

In this scenario, we assume that the institution that decides to use the education application has SLAs with the SP describing expected quality of the services. QoS requirements are formailzed in Service Level Agreements for the expected level of service between the SP and the IP. In the context of this research, meeting QoS requirements for a cloud service refers to meeting different quantity of VM resources at run-time. This research on trust and cloud computing considers general cloud computing utilities such as CPU and storage resources required for general data services. However, other QoS requirements such as bandwidth, delay can also be considered. Often QoS is associated with buisiness level objectives (BLOs) by the cloud infrastructure providers eg. Maximize profit without breaking more than certain fraction of SLAs. However such BLOs have high impact on the trust associated with the IP. The higher the SLA violation, the lower the trust for the IP which violates SLAs.

The SLAs specify several indicators with which the SP is required to comply, and any violations may lead to penalty payments, as well as negative impact in the customer's satisfaction. Examples of SLA indicators considered in this research are cpu, disk space, memory, and number of desktops. In order to meet the customer's requirements, the SP that uses the infrastructure services from the IPs also have SLAs with the IP. An SLA between an SP and an IP considers all the existing SLA's with the various customers and the possibility of growing the demand of the application.

An SLA between an SP and IP represents elasticity requirements to support the SP to demand more resources dynamically based on the requirements. For example, when the application receives request for a new desktop, it requests a virtual machine to be created in the infrastructure of the IP where the application is deployed. Similarly, the application can receive requests to increase memory, cpu, or disk space for the existing virtual desktops, which are forwarded to the IP to fulfil the requirements. If the IP, at any point of time fails to provide the requested resources, or is not able to maintain the resource requirements of existing virtual desktops, then this may lead to SLA violations for the corresponding indicators.

4.3 Trust Model

As described in Section 4.1, *Trustworthiness* of an IP is modelled using *opinion* obtained from three different computations, namely (i) *compliance of SLA parameters (SLA monitoring),* (ii) *service provider satisfaction ratings (SP ratings)*, and (iii) *service provider behaviour (SP behaviour).* The *opinion* is expressed in terms of *belief*, *disbelief*, *uncertainty* and *base rate* which is used in conjunction with the subjective logic (Jøsang, 2001).

The *opinion* of an entity (SP or IP) *A* for a proposition *x* is given as $W^A_x = (b^A_x, d^A_x, u^A_x, a^A_x)$, where b^A_x is the belief in the proposition, d^A_x is the disbelief in the proposition, u^A_x is the uncertainty of the proposition, a^A_x is base rate that provides the weight of uncertainty that contributes to the probability expectation. Without evidence, base rate alone determines the probability distribution and as more evidence becomes available, the influence of the base rate diminishes. Belief b^A_x, disbelief d^A_x, uncertainty u^A_x and base rate a^A_x can also be represented as b_x, d_x, u_x, a_x. All b_x, d_x, u_x, a_x Є [0.0, 1.0], and $b_x+d_x+ u_x=1$.

As *trustworthiness(T)* of an IP uses three different computations i.e. *(SLA monitoring)*, *SP ratings* and *SP behaviour*, it is essential to understand and use the correct methods to combine these different trust. Analogous to Dempster rule of combining beliefs, Josang presents subjective logic which is a belief calculus specifically developed for modelling trust relationships. Subjective logic is compatible with binary logic and probability calculus and defines rich set of operators for combining opinions (Jøsang, 2001). The trust defined in the thesis is dynamic and non-monotonic i.e. experiences can increase or decrease and the evaluation of trust model in Chapter 7 verifies this property.

The *trustworthiness (T)* of an IP is modelled as the expectation of the combined opinion of all the three computations. The opinions are combined using the *conjunction* operator (Λ), *consensus* operator ($\oplus$), and the *discounting* operator ($\otimes$) in the subjective logic (Jøsang, 2001). Let us consider $W_{SLA,}$ W_{SPR}, W_{SPB} are opinions obtained from the SLA monitoring (SLA), SP ratings (SPR), and SP behaviour (SPB) values, respectively. The *trustworthiness (T)* is given as follows:

$$T\text{=Expectation }(W_{(SPB \otimes SPR)\Lambda\, SLA}) \qquad (4.1)$$

$$W_{(SPB \otimes SPR)\Lambda SLA} = (W_{SPB} \otimes W_{SPR})\ \Lambda\ W_{SLA} \qquad (4.2)$$

The symbol Λ is the *conjunction operator* used to combine the opinions, and $\otimes$ is the *discounting operator* used as the recommendation operator. If $W_x = (b_x, d_x, u_x, a_x)$ and $W_y = (b_y, d_y, u_v, a_y)$ are the opinions of an entity for proposition x and proposition y, then the combined opinion is given as $W_{x\Lambda y} = (b_{x\Lambda y}, d_{x\Lambda y}, u_{x\Lambda y}, a_{x\Lambda y})$.

Consider two agents A and B, where $W^A_B = (b^A_B, d^A_B, u^A_B, a^A_B)$ is A's opinion about B's advice, and let x be the proposition where $W^B_x = (b^B_x, d^B_x, u^B_x, a^B_x)$ is B's opinion about x expressed as an advice to A. In this case, W^{AB}_x is called the discounting ($\otimes$) of opinion W^B_x by opinion W^A_B which is given as $W^{AB}_x = W^A_B \otimes W^B_x = (b^{AB}_x, d^{AB}_x, u^{AB}_x, a^{AB}_x)$ where $b^{AB}_x, d^{AB}_x, u^{AB}_x, a^{AB}_x$ is given as follows (Jøsang, 2001):

$$b^{AB}_x = b^A_B \, b^B_x \tag{4.3}$$

$$d^{AB}_x = b^A_B \, d^B_x \tag{4.4}$$

$$u^{AB}_x = d^A_B + u^A_B + b^A_B \, u^B_x \tag{4.5}$$

$$a^{AB}_x = a^B_x \tag{4.6}$$

Consider two agents A and B, where $W^A_x = (b^A_x, d^A_x, u^A_x, a^A_x)$ is A's opinion for a proposition x and $W^B_x = (b^B_x, d^B_x, u^B_x, a^B_x)$ is B's opinion about x. Let $W^{A,B}_x = (b^{AB}_x, d^{AB}_x, u^{AB}_x, a^{AB}_x)$ where $b^{AB}_x, d^{AB}_x, u^{AB}_x, a^{AB}_x$ is given as follows (Jøsang, 2001):

$$b^{AB}_x = (b^A_x \, u^B_x + b^B_x \, u^A_x) \,/\, k \tag{4.7}$$

$$d^{AB}_x = (d^A_x \, u^B_x + d^B_x \, u^A_x) \,/\, k \tag{4.8}$$

$$u^{AB}_x = (u^A_x \;\, u^B_x \,) \,/\, k \tag{4.9}$$

$$a^{AB}_x = (a^B_x \, u^A_x + a^A_x \, u^B_x + (a^A_x + a^B_x) \, u^A_x \, u^B_x) \,/\, (u^A_x + u^B_x - 2 u^A_x \, u^B_x) \tag{4.10}$$

Where $W^{A,B}_x = W^A_x \oplus W^B_x$ is called the consensus between W^B_x and W^B_x representing an imaginary agent [A,B]'s opinion about x as if it has represented both A and B. Where k= $u^A_x + u^B_x - u^A_x \, u^B_x$ such that k≠ 0 and $a^{AB}_x = (a^A_x + a^B_x)/2$ when $u^A_x, u^B_x = 1$.

4.3.1 Opinion Representation

For any proposition *x,* the opinion is given by $W_x = (b_x, d_x, u_x, a_x)$, with

$$b_x = c\, r / t \qquad (4.11)$$

$$d_x = c\, s\ / t \qquad (4.12)$$

$$u_x = t / (r\, s + f^2 + 1) \quad \text{for } t \geq 1 \quad \text{or}$$

$$u_x = 1 \quad \text{for } t < 1 \qquad (4.13)$$

$$c = 1 - u_x \qquad (4.14)$$

Where: *r* is the amount of positive evidence; *s* is the amount of negative evidence; *t* is the total evidence given as *t=r+s*; *f* is the distance of focus to the centre of an ellipse; and c, *c(t)* is certainty that is the function of total evidence *t* and can also be represented as a function of positive and negative evidence given as *c(r, s).* The opinion model uses certainty *c(t)* to model the *belief*, *disbelief* and *uncertainty*.

The proposed opinion model considers two aspects of uncertainty due to the evidence at hand, namely: (i) as the amount of evidence increases the uncertainty reduces; and (ii) in a given total evidence, as the positive or negative evidence dominates, the uncertainty decreases, and as the positive and negative evidence equals, the uncertainty increases. These two aspects of uncertainty exhibit behaviour similar to the properties of an ellipse, considering its size and shape, controlled by its axis and area.

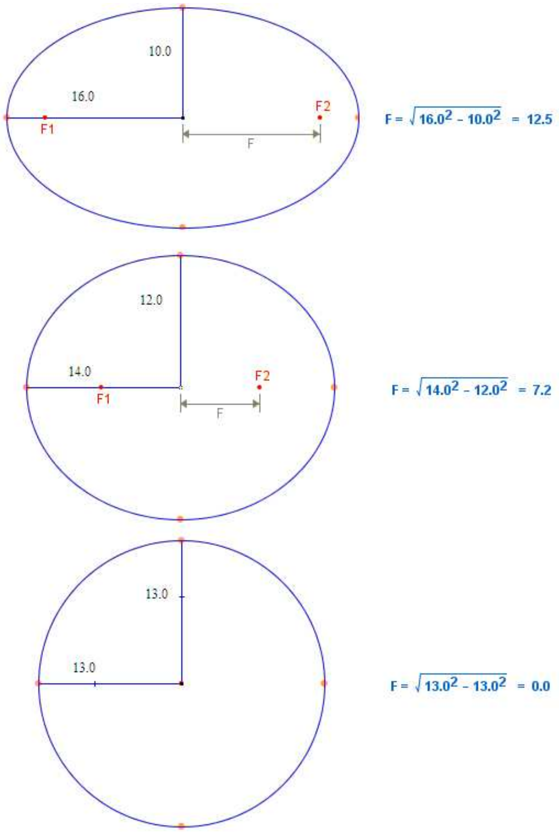


Figure 4.2 : **Ellipse shapes**

In our model, uncertainty is defined as a function of an ellipse area and shape as in Figure 4.2. More specifically, the uncertainty model is derived using the properties of an ellipse wherein the positive and negative evidence is mapped to the major and minor semi-axes of an ellipse, respectively. The first aspect of uncertainty (i.e., increases in evidence, decreases the uncertainty) is achieved by using the area of the ellipse given by the product of its two semi-axes. As the positive and negative evidence is being mapped to the major and minor semi-axes of ellipse, any increase in the major and minor semi-axes results in the increase of the area of ellipse and decrease of the uncertainty. The second aspect of uncertainty is due to dominance between positive and negative evidence, which is captured using the shape of an ellipse. The shape of an ellipse is a function of its two semi-axes. The positive and negative evidence being mapped to the semi-axes of an ellipse, as the major semi-axis continues to dominate, the distance of focus (either F1 or F2) with the centre is a positive value and as the two semi-axes equals, this distance approaches to zero, transforming to a circle.

The change in major and minor semi-axes affects the distance of focus with the centre which is given as $f = sqrt\ (a^2 - b^2)$ where '*a'* is the major semi-axis and '*b'* is minor semi-axis. If the total evidence is fixed to a constant, the variation of the positive and negative evidence affects the shape of the ellipse. If the positive and negative evidence equals, this makes $f = 0$, transforming the ellipse to a circle. This adds to a highest uncertainty in given total evidence. As the positive and negative evidence continues to dominate, this leads to a positive value for f and this value, is maximum, when either positive or negative evidence in the total evidence is zero. This adds to a lowest uncertainty in given total evidence. Both properties of uncertainty are captured in the uncertainty definition in equation 4.9:

The expectation of the opinion about a proposition x is given as:

$$E(x) = b_x + a_x u_x \qquad (4.15)$$

4.3.2 SLA Monitoring

The SLA monitoring determines the opinion about an IP from the SLAs that the IP have established with the SPs for their services. The SP for each of its service has a single SLA that includes several indicators (e.g., number of CPUs, memory, disk space, number of VMs). Continuous measurement of QoS at a minute granularity, for various services in cloud environments to meet SLAs is performed using monitors(Al-Shammari and Al-Yasiri, n.d.; Bardhan and Milojicic, 2012; Hasan and

Huh, 2013). For each indicator of an SLA, there is an associated monitor that evaluates the compliance/non-compliance of the indicator.

The SLA monitoring opinion about an IP is a two-step process. In the first step, a *consensus opinion* is created for an indicator type (e.g., number of CPUs) based on information from all the monitors verifying the compliance of the indicator. This opinion indicates the trust of an IP only based on the indicator used to create the *consensus opinion*. In the second step, a *conjunction opinion* is created about the IP for either a set of indicators or for all the indicators based on the requirement. The *conjunction opinion* indicates the trust of an IP for the set of indicators based on SLA monitoring.

Consider that there are *m* indicator types and *n* monitors associated with each indicator type. In this case, the opinion of the SLA monitoring is given as:

$$W_{SLA} = W_1^{(M1,1),\ldots,(M1,n)} \, \Lambda \, W_2^{(M2,1),\ldots,(M2,n)} \, \Lambda \, \ldots \Lambda \, W_m^{(Mm,1),\ldots,(Mm,n)} \tag{4.16}$$

where, $W_1^{(M1,1),\ (M1,2),(M1,3),\ldots,(M1,n)}$ is the consensus opinion for the indicator type ‘1’ given by monitors M1,1 to M1, *n* belonging to different SLAs. If $W^A_x = (b^A_x, d^A_x, u^A_x, a^A_x)$ *and* $W^B_x = (b^B_x, d^B_x, u^B_x, a^B_x)$ are the opinions given by agent *A* and agent *B,* respectively for the same proposition *x,* then the *consensus opinion* is given as follows (Jøsang, 2001): $W^{A,B}_x = W^A_x \oplus W^B_x = (b^{A,B}_x, d^{A,B}_x, u^{A,B}_x, a^{A,B}_x)$

Example: In order to comprehend SLA Monitoring, consider the education application described in Section 4.2. Consider a case wherein, at that end of academic year most university students need high computation resources such as large number of VMs, memory space, number of CPUs and disk space for doing individual projects. For each of the Universities the requested resource to the SP is within the agreed SLA. The SP demands resources from the IP. As in the example scenario, since IP1 have all five SPs hosting the education application, the demand to increase the resources occurs almost in the same time frame. Given the constraint that the IP1 cannot acquire resources from other IPs for these applications, there is a violation of the SLA after verifying that IP1 has no additional resource of its own to be provided.

In the scenario, IP1 has five SLAs, with each of the SPs (SP1 to SP5) for four different indicator types (number of CPUs, memory, disk, and VM). Assume SLA1 with SP1, SLA2 with SP2, and so on. Consider the existence of monitors associated with each indicator of the SLAs. Assume four monitors (M1, M2, M3 and M4) to be associated with the SLA1 for number of CPUs, memory, disk space, and VM, respectively. Similarly, monitors M5 to M8, M9 to M12, M13 to M16 and M17 to M20 are associated with SLA2, SLA3, SLA4 and SL5, for the various SLA indicators.

Each of the monitors associated with the indicators provides information about the compliance of the respective indicator for an IP. If we consider that monitors M1, M2, M3 and M4 indicated 150 compliances and 10 non-compliances (i.e., 150 positive and 10 negative evidences) for IP1. The opinions given by the monitors for SLA1 are calculated using the proposed opinion model as follows: positive evidence r=150 and negative evidence s=10 provide total evidence t=150+10=160; based on the evidence the focus f=sqrt(150^2 -10^2)=149.66; the uncertainty, belief and disbelief are finally computed as follows:

$$u_x=160 / (150 * 10 + 149.66 * 149.66 + 1) = 0.006694 \quad (4.17)$$

$$c_x = 1 – u_x = 1 -0.00669 = 0.9933 \quad (4.18)$$

$$b_x = 0.9933 * 150 / 160 = 0.93122 \quad (4.19)$$

$$d_x = 0.9933 * 10 / 160 = 0.062082 \quad (4.20)$$

$$W_{CPU}^{M1} = (b^{M1}{}_{CPU}, d^{M1}{}_{CPU}, u^{M1}{}_{CPU}) = (0.93122, 0.062082, 0.006694) \quad (4.21)$$

$$W_{mem}^{M2} = W_{disk}^{M3} = W_{vm}^{M4} = (0.93122, 0.062082, 0.006694) \quad (4.22)$$

If we consider that all the other monitors M5-M20 associated with SLA2, SLA3, SLA4 and SLA5 also have 150 compliance and 10 non-compliance indicators, the opinion provided by these monitors are the same as the above ones.

The opinion for IP1 with respect to number of CPUs is given as the *consensus opinion* of the five monitors M1, M5, M9, M13 and M17 as follows:

$$W_{CPU}^{M1,M5,M9,M13,M17} = (b^{M1,M5,M9,M13,M17}{}_{CPU}, d^{M1,M5,M9,M13,M17}{}_{CPU}, u^{M1,M5,M9,M13,M17}{}_{CPU})$$

$$= (0.936238, 0.062416, 0.001346) \quad (4.23)$$

Similarly, the opinion for IP1 based on memory, disk and virtual machine is:

$$W_{mem}^{M2,M6,M10,M14,M18} = W_{disk}^{M3,M7,M11,M15,M19}$$

$$= W_{VM}^{M4,M8,M12,M16,M20}$$

$$=(0.936238, 0.062416, 0.001346) \quad (4.24)$$

The overall opinion for IP1 based on all the indicators of the SLAs is given as the *conjunction opinion* of all *consensus opinions* for each of the indicator as follows:

$$W_{SLA} = W_{CPU}^{M1,M5,M9,M13,M17} \Lambda$$

$$W_{mem}^{M2,M6,M10,M14,M18} \Lambda$$

$$W_{disk}^{M3,M7,M11,M15,M19} \Lambda$$

$$W_{VM}^{M4,M8,M12,M16,M20}$$

$$= (0.768325, 0.227246, 0.004428) \quad (4.25)$$

4.3.3 SP Behaviour

The SP behaviour is defined in terms of the number of times the SP has used the IP against the SPs total usage. An SP using a single IP for the majority of the times indicates the SPs good behaviour or good opinion towards an IP. The SP may use the IP for one or more indicators specified in the SLA.

Consider that there are *m* indicator types that the IP has negotiated from all the ‘q` SPs in the past. Let there be a monitor associated with each indicator type. This result in *m* monitors associated with each of the SPs to monitor how many times the SP used this IP for a given indicator, against its total usage for that indicator. Suppose that SP1 used IP1 five times, IP2 three times, and IP3 four times for CPU usage. This indicates that for CPU total usage of 12 times, SP1 has used IP1 five times. This information is used to model the opinion of SP1’s behaviour towards IP1 for CPU usage. Assume monitor M1,1 associated with the indicator of type ‘1’ to monitor SP1’s behaviour towards IP1. In this case, the opinion is represented as $W_{SP1}^{M1,1}$. A single overall behaviour of an SP towards an IP is given as a consensus opinion of all its indicators. The behaviour of SP1 towards IP1 is given as:

$$(W_{SP1}^{M1,1} \oplus W_{SP1}^{M2,1} \oplus W_{SP1}^{M3,1} \oplus .... \oplus W_{SP1}^{Mm,1}) \qquad (4.26)$$

All ‘q’ behaviour of SP towards an IP is given as the conjunction opinion as:

$$W_{SPB} = (W_{SP1}^{M1,1} \oplus .... \oplus W_{SP1}^{Mm,1}) \wedge ... \wedge$$

$$(W_{SPq}^{M1,q} \oplus .... \oplus W_{SPq}^{Mm,q}) \qquad (4.27)$$

Example: In order to illustrate consider the education application described in Section 4.2 with monitors M1, M2, M3 and M4 verifying the compliance of the CPU, memory, disk and virtual machine usage, respectively, for SP1, and monitors M6-M8, M9-M12, M13-M16, and M17 - M20 for SP2, SP3, SP4 and SP5. Suppose that monitor M1 associated with SP1, records that SP1 has opted to use IP1 for 200 times against SP1’s 250 times total CPU usage. The opinion for the behaviour of SP1 towards IP1 for CPU usage is calculated as:

$$W_{SP1}{}^{M1}=(b^{M1}{}_{SP1},\ d^{M1}{}_{SP1},\ u^{M1}{}_{SP1})$$

$$= (0.79579,\ 0.198947,\ 0.005263) \qquad (4.28)$$

Similarly, assume that M2, M3 and M4 record the same usage as M1 for memory, disk space, and virtual machine, respectively. The opinions are calculated as:

$$W_{SP1}{}^{M2}= W_{SP1}{}^{M3}$$

$$= W_{SP1}{}^{M3}$$

$$= W_{SP1}{}^{M4}$$

$$= (0.79579,\ 0.198947,\ 0.005263) \qquad (4.29)$$

Consider that SP2 and SP3 have the same evidence as in the case of SP1, with the associated monitors for these SPs providing evidences as monitors M1, M2, M3 and M4. Consider SP4 with monitors M13-M16 and SP5 with monitors M17-M20 using other IPs different from IP1 for its resources consumption. Assume the monitors for SP4 and SP5 provide 100 positive evidences and 150 negative evidences for each of its indicators. This evidence is transformed to the opinions below:

$$W_{SP4}{}^{M13}=W_{SP5}{}^{M17}=W_{SP4}{}^{M14}=W_{SP5}{}^{M18}$$

$$=W_{SP4}{}^{M15}=W_{SP5}{}^{M19}=W_{SP4}{}^{M16}= W_{SP5}{}^{M20}$$

$$= (0.39636,\ 0.594546,\ 0.009091) \qquad (4.30)$$

The behaviour of SP1 towards IP1 (and of SP2 and SP3) are calculated as:

$$W_{SP1}{}^{M1...M4}=W_{SP1}{}^{M1} \oplus W_{SP1}{}^{M2} \oplus W_{SP1}{}^{M3} \oplus W_{SP1}{}^{M4}$$

$$= (0.798943,\ 0.199736,\ \text{and}\ 0.001321) \qquad (4.31)$$

The behaviour of SP4 and SP5 towards IP1 is given as:

$$W_{SP4}{}^{M13M14M15M16} = W_{SP5}{}^{M17M18M19M20}$$

$$= (0.399085, 0.598627, 0.002288) \qquad (4.32)$$

The total SPs behaviour towards an IP is given as the *conjunction* opinion of all SPs towards a single IP, given as:

$$W_{SPB} = W_{SP1}{}^{M1\ldots M4} \Lambda W_{SP2}{}^{M5\ldots M8} \Lambda W_{SP3}{}^{M9\ldots M12} \Lambda W_{SP4}{}^{M13\ldots M16} \Lambda W_{SP5}{}^{M17\ldots M20}$$

$$= (0.081223, 0.917435, 0.001342) \qquad (4.33)$$

4.3.4 SP Ratings

The SP satisfaction rating is calculated based on the rates of the services given by an SP using an IP. The SP provides separate ratings for each SLA indicators of the IP's services. The ratings are used to form an opinion about an IP. Similar to the other cases, the computation of SP ratings to provide an opinion about an IP is based on consensus and conjunction ratings. Consider *q* SPs available and each of these SPs providing its opinion for one or more of the *m* indicator types that the IP supports. The SP satisfaction rating is calculated as:

$$W_{SPR} = W_1{}^{SP1,SP2\ldots,SPq} \ \Lambda \ W_2{}^{SP1,SP2\ldots,SPq} \Lambda \ \ldots \ \Lambda W_m{}^{SP1,SP2\ldots,SPq} \qquad (4.34)$$

Where, $W_i^{SP1,SP2\ldots,SPq}$is the consensus opinion for indicator type 'i' from SP1 to SPq.

Example: Consider ratings provided by SP ranging in five intervals [excellent, Good, average, bad, worst]. As an example, suppose that SP1 has provided 100 excellent and 5 worst ratings for number of CPUs, memory, disk, and virtual machine indicators. These ratings are transformed into 100 positive and 5 negative evidences for each of these indicators, as per the mapping described above. Based on the evidence of ratings for IP1, the opinion that SP1 has about IP1 for its indicators is given as:

$$W_{CPU}{}^{SP1}= (b^{SP1}{}_{CPU}, d^{SP1}{}_{CPU}, u^{SP1}{}_{CPU})$$

$$= (0.94284, 0.047142, 0.010023) \quad (4.35)$$

$$W_{mem}{}^{SP1} = W_{disk}{}^{SP1}$$

$$= W_{vm}{}^{SP1}$$

$$= (0.94284, 0.047142, 0.010023) \quad (4.36)$$

Suppose that SP2, SP3, SP4 and SP5 have provided (200 excellent, 5 worst), (200 excellent, 10 worst), (200 excellent, 20 worst), (200 excellent, 30 worst) ratings, respectively for IP1 for each of the four different indicators. The ratings transformed to evidences provide the following opinions of SP2, SP3, SP4 and SP5 about IP1, calculated as:

$$W_{CPU}{}^{SP2} = W_{mem}{}^{SP2}$$

$$= W_{disk}{}^{SP2}$$

$$= W_{vm}{}^{SP2}$$

$$= (0.97073, 0.024268, 0.005003) \quad (4.37)$$

$$W_{CPU}{}^{SP3} = W_{mem}{}^{SP3}$$

$$= W_{disk}{}^{SP3}$$

$$= W_{vm}{}^{SP3}$$

$$= (0.94761, 0.04738, 0.005012) \quad (4.38)$$

$$W_{CPU}{}^{SP4} = W_{mem}{}^{SP4}$$

$$= W_{disk}{}^{SP4}$$

$$= W_{vm}{}^{SP4}$$

$$= (0.90450, 0.09045, 0.005046) \quad (4.39)$$

$$W_{CPU}{}^{SP5} = W_{mem}{}^{SP5}$$

$$= W_{disk}{}^{SP5}$$

$$= W_{vm}^{SP5}$$

$$= (0.86513,\ 0.12977,\ 0.0051) \tag{4.40}$$

The capability of IP1 for providing number of CPUs, memory, disk, and VM are given as the consensus of all SP's opinion by:

$$W_{CPU}^{SP1} \oplus W_{CPU}^{SP2} \oplus W_{CPU}^{SP3} \oplus W_{CPU}^{SP4} \oplus\ W_{CPU}^{SP5}$$

$$= (0.928743,\ 0.070133,\ 0.001124) \tag{4.41}$$

$$W_{mem}{}^{SP1 \ldots SP5} = W_{disk}{}^{SP1 \ldots SP5}$$

$$= W_{VM}{}^{SP1 \ldots SP5}$$

$$= (0.928743,\ 0.070133,\ 0.001124) \tag{4.42}$$

The overall opinion formed for IP1 based on the ratings from the SPs is given as:

$$W_{SPR} = W_{CPU} \Lambda W_{mem} \Lambda W_{disk} \Lambda W_{VM}$$

$$=(0.744015,\ 0.252376,\ 0.003609) \tag{4.43}$$

4.3.5 SP Ratings Discounted by SP Behaviour

The proposed trust model uses the behavior of the SP for discounting the opinion provided by the SP in SP ratings, for a particular indicator. More specifically, in the SP ratings, if SP1 is evaluating IP1 and is informed about the opinion of IP1 from SP2 regarding CPU indicator, this opinion of SP2 is discounted using SP2's behavior about CPU towards IP1.

In the case of SP behaviour, if monitor M1,2 is associated with indicator type '1' to monitor SP2's behaviour towards IP1, then this opinion is represented as $W_{SP2}^{M1,2}$. In the case of SP ratings, SP1 being informed about opinion from SP2 for IP1 based on indicator type '1' is represented as W_1^{SP2}. Based on the behaviour of SP2 towards IP1 for CPU indicator, SP2's opinion for CPU is discounted. In other words, the opinion W_1^{SP2} is discounted by $W_{SP2}^{M1,2}$ value and is given as $W^{(M1,2)SP2}_{1} = W^{M1,2}_{SP2} \otimes W_1^{SP2} = (b^{(M1,2)SP2}_{1}, d^{(M1,2)SP2}_{1}, u^{(M1,2)SP2}_{1}, a^{(M1,2)SP2}_{1})$

SP ratings after discounting opinions using the SP behaviour for each of the indicator, also follows the two-step process of *consensus* and *conjunction* to get the combined opinion of SP rating and SP behaviour which are given as follows:

$$
\begin{aligned}
W_{(SPR \otimes SPB)} &= W_{SPB} \otimes W_{SPR} \\
&= (W^{M1,1}_{SP1} \otimes W_1^{SP1}) \oplus (W^{M1,2}_{SP2} \otimes W_1^{SP2}) \oplus \ldots \oplus \\
&\quad (W^{M1,q}_{SPq} \otimes W_1^{SPq}) \Lambda \\
&\quad (W^{M2,1}_{SP1} \otimes W_2^{SP1}) \oplus (W^{M2,2}_{SP2} \otimes W_2^{SP2}) \oplus \ldots \oplus \\
&\quad (W^{M2,q}_{SPq} \otimes W_2^{SPq}) \Lambda \ldots \Lambda \\
&\quad (W^{Mm,1}_{SP1} \otimes W_m^{SP1}) \oplus (W^{Mm,2}_{SP2} \otimes W_m^{SP2}) \oplus \ldots \oplus \\
&\quad (W^{Mm,q}_{SPq} \otimes W_m^{SPq})
\end{aligned}
\tag{4.44}
$$

4.4 Conclusion

This chapter presents a new trust model to support service providers to verify trustworthiness of infrastructure providers in cloud computing environments. The model calculates trust values based on different parameters, namely (i) SLA monitoring compliance, (ii) service provider ratings, and (ii) service provider behaviour. The trust values are calculated based on an opinion model in terms of belief, disbelief, uncertainty and base rate.

The evaluation of the trust model proposed in this Chapter is discussed in Chapter 7.

Chapter 5 Trust Model for Cloud Based On Cloud Characteristics

Although, several trust models exist in different areas including for cloud, none of the trust models to-date is comprehensive enough to accommodate the characteristics of the cloud environment. This chapter extends the previously defined trust model, to include the essential cloud characteristics as the dimensions of the trust model together with several features relevant to the dimension to build the context. The previous trust model is supported with an opinion model that considers uncertainty for building context specific trust by providing opinion for each of the parameters and the extension supports credibility to reduce the impact of malicious feedback providers. The early filtering of malicious feedback mechanism compliments the credibility by further reducing the influence of malicious node. The proposed extension makes the trust model robust against malicious feedback providers.

5.1 Introduction

With huge number of cloud service providers available in the market, it is challenging for the consumers/SPs to decide which IP will be trustworthy for their services to be deployed in the cloud environment. Trust being a fundamental subject, several trust models exist to date in different areas. However, cloud being the recent advancement in computing, there are a very few trust models that characterize the cloud environment and also these trust models do not comprehensively incorporate cloud properties (Ferrer *et al*., 2012; Hwang *et al*., 2009; P.S. Pawar *et al*., 2012).

In this chapter we evaluate the trustworthiness of the IP using Cloud Broker (CBR) architecture. The CBR acts as an IP to the SP and it acts like an SP to the IP. The CBR acting as an intermediary receives service deployment request for which the CBR needs to select the most suitable IP for hosting the SP's service.

The trust model described in this chapter is comprehensively tailored specifically towards the cloud environment. The parameters of the trust model are derived from the essential cloud characteristics as defined by NIST(Mell and Grance, 2011). The trust model considers the essential cloud characteristics as the dimensions of the trust model and for each of these dimension certain features are identified that assists in modelling the trust value. The dimensions are: o*n-demand self-service*, *resource pooling*, *rapid elasticity* and *measured service*. The features of *On-demand self-service* are: *availability_d* and *timely_d.* The features of *Rapid elasticity* are: *availability_e* and *timely_e.* The features of *Resource pooling* includes: *affinity* and *legal*. The features *viewable*, *controllable* and *reportable* are for the *measured services*.

The trust model in this chapter defines trust in the form of *reliability* and *reputation* subject to the *credibility* of the feedback provider. A similar approach has been used in (Jia et al., 2012), but the fundamental advantage of the model proposed in this chapter is that it is sensitive to uncertainty of the information (i.e. feedback) provided by the feedback providers. This is very crucial for the computation of the reputation and later in Chapter 7 of this thesis report, an evaluation is presented to show the impact of uncertainty on the robustness of the trust model. The trust model in this chapter defines credibility, which reduces the influence of malicious nodes on the value of reputation score computed for an entity. A similar approach has been followed in (Jia et al., 2012), but the trust framework in this chapter incorporates an additional early filtering mechanism to filter malicious node which drastically reduces the influence of the malicious node. The mechanism of early filtering of malicious node complements the credibility approach of reducing the influence of malicious nodes. The work in this chapter evaluates the trust model based on filtering of malicous nodes by using an outlier detection technique that is proposed in (Arning et al., 1996; Zhang and Feng, 2009), showing the advantage of applying an early malicious node filtering technique

The rest of the chapter is structured as follows: Section 5.2 describes a Cloud Computing Example and a CBR scenario that is used across the chapter to illustrate the work. Section 5.3 describes the trust framework and the trust model in detail. Section 5.4 provides concluding remarks for the work in this chapter.

5.2 Cloud Computing Example

In order to illustrate and evaluate our work in this chapter, we present a CBR scenario that has been developed within the OPTIMIS project. As seen in Figure 5.1, for evaluating our proposed model we considered hundred SP's, hundred IP's, and a single CBR. The CBR acts as an intermediary that has capabilities of both the SP as well as an IP. The SP considers the broker as an IP for deploying its service, while the CBR acts as an SP to deploy the services in the infrastructure provided by the IPs.

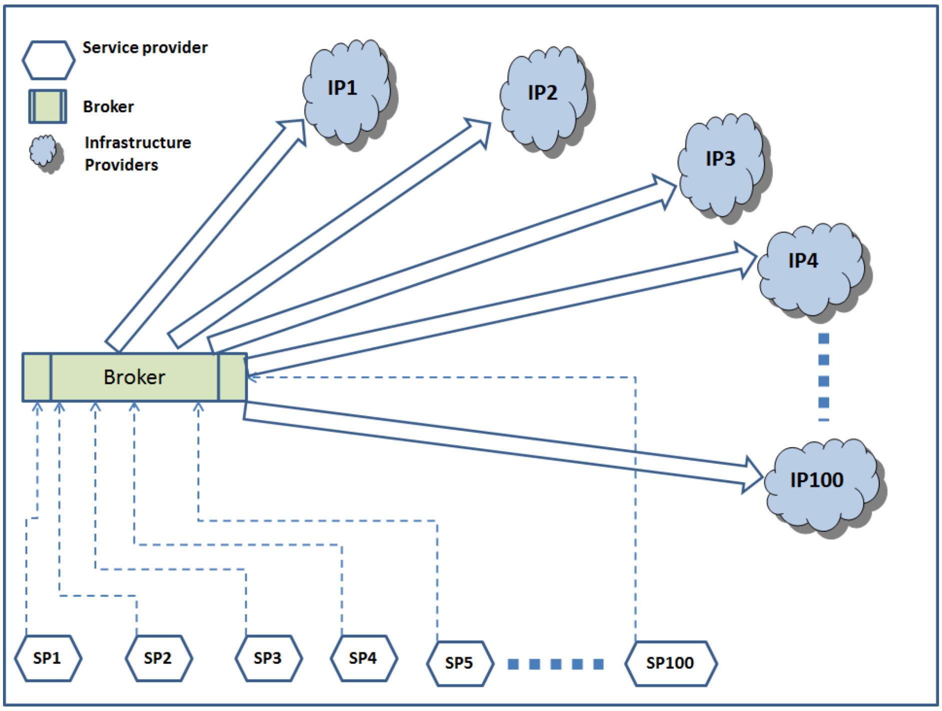


Figure 5.1 : Cloud Computing Environment

In the Scenario, we assume that the SPs have registered with the cloud broker for getting infrastructure services from the IPs. The SPs may also have independently taken infrastructure services from the IPs and may be continuing to do so.

5.2.1 Cloud Broker Scenario

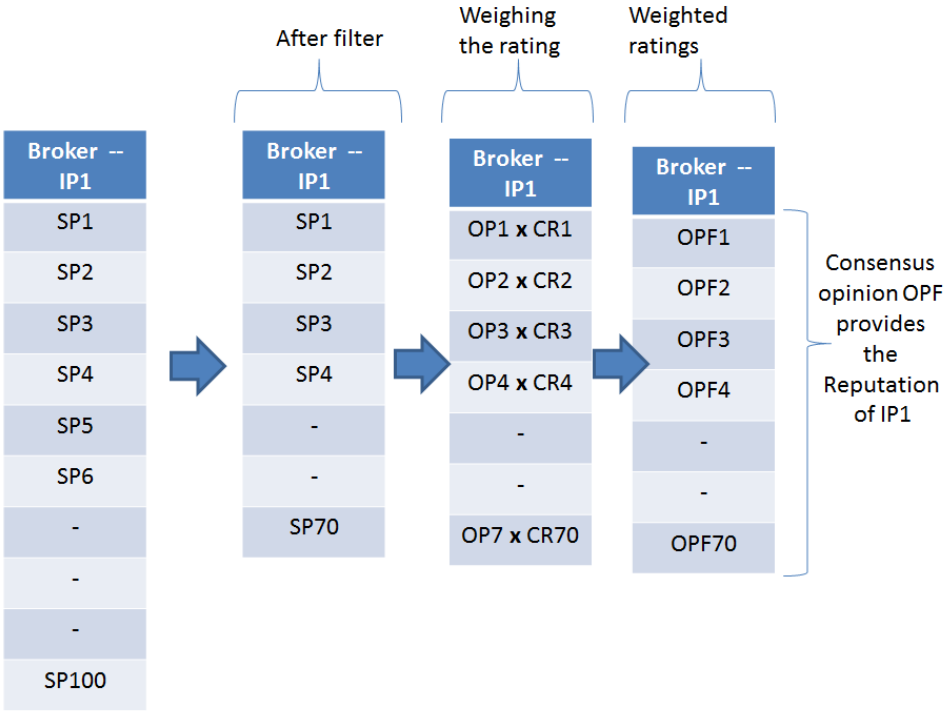


Figure 5.2 : Cloud Broker (CBR) example scenario

Figure 5.2 presents the CBR example scenario used for evaluating the Trust model. The example consists of the CBR evaluating the trust of an IP. The CBR receives feedback from SP1 to SP100 in the form of opinion, which passes through a filter, which in turn filters the nodes that provide the malicious ratings for IP1. In the example shown in Figure 5.2, SP1-SP70 passes successfully through the filter and the feedback from SP71-SP100 are not considered for computing the reputation of IP1. The feedbacks provided by SP1-SP70 are weighted by the corresponding *credibility* which the CBR have for each of the feedback providers i.e. CR1 refers to the credibility that the CBR has for the node SP1. CBR forms Opinions on the feedbacks by considering the credibility of the feedback provider i.e. OPF1 refers to the opinion on the feedback (OP1) provided by SP1 weighted by credibility CR1. The weighted ratings OPF1 – OPF70 are used by the CBR to compute the reputation score of IP1. The consensus opinion OPF forms the reputation score for IP1.

5.3 Trust Framework

As briefed in Section 5.1, the trustworthiness of the IP is modelled based on the cloud characteristics (Mell and Grance, 2011) to have dimensions as: *on-demand self-service(os)*, *resource pooling(rp)*, *rapid elasticity(re)* and *measured service(ms)*.

The *on-demand self-service* characteristics, enables the consumer to unilaterally provision computing resources without requiring any human interaction. The *rapid elasticity* characteristic of the cloud provider enables the consumer to scale resources rapidly up and down based on demand. The *resource pooling* characteristics of the cloud environment enables cloud service providers to use multi-tenant model, dynamically assigning physical and virtual resources with location independence. The *measured service* characteristic of cloud enables it to control and optimize the resources by metering capability at certain level of abstraction such as storage, bandwidth, processing etc. The resources can be controlled based on the agreement between the consumer and the provider. The resource usage can be monitored, controlled and reported by providing transparency to the provider and the consumer.

Each dimension that represents a cloud characteristic contains a list of features to specify the context. The *on-demand self-service* dimension includes the following features: *availability_d* and *timely_d*. The feature *availability_d* contributes to the dimension by capturing the availability of resources in the event of an on-demand resource provisioning request. The feature *timely_d* contributes to the dimension with the provider’s capability to provision the resource within a suitable time. The *rapid elasticity* dimension includes the features: *availability_e* and *timely_e*. The *availability_e* and *timely_e* features contribute to the *rapid elasticity* dimension during the occurrence of the event that triggers elasticity. The *resource pooling* dimension includes the features: *affinity* and *legal*. The *affinity* feature and the *legal* feature capture the provider's capability/violations towards the provisioning of resources with the given affinity constraints and within the location boundaries respectively. The *measured service* dimension takes into account the features related to resource usage that includes: *viewable*, *controllable* and *reportable*. The features *viewable*, *controllable* and *reportable* provides the capability of the infrastructure provider to view, control and report resource usage.

Each SP and CBR on the direct interaction with the IP will have a certain amount of trust for an IP. Whenever an SP wants to use IP, the SP ranks the IPs based on the trustworthiness value of IPs which are calculated based on its direct experience in the past as well as the feedback received from other SPs. Each SP and CBR has evidences of its direct experiences with the IPs, stored along the dimensions and features of the dimensions. The work in this chapter is mainly focused on the CBR assessing the trust and reputation of the IPs.

5.3.1 Trust Model

Trust is computed considering direct experiences, indirect experience and a balancing factor. Direct experiences are used to compute the reliability trust and indirect experiences in terms of feedbacks are used to compute reputation trust. Confidence in reliability trust and reputation trust is considered as the balancing factor to assign weight to each of this trust. The trust model comprising of *reliability* trust and *reputation* trust is defined as follows:

$$Trust = confidence * Reliability + (1 - confidence) * Reputation \quad (5.1)$$

Where *confidence* is the weight trustee assigns to the reliability trust that is evaluated through direct interaction. The *confidence* value ranges between [0-1]. Reputation trust is based on the feedback received.

5.3.2 Reliability trust

The reliability of an entity such as IP is based on the direct interaction between the SP and IP. Let us denote R(i, j): dimension as the reliability of entity j from the perspective of entity i for the given dimension. The SP updates its rating and reliability for each feature of the dimension. The overall reliability of entity j from the perspective of entity i, is given as weighted average for all dimension which is as follows:

$$\begin{aligned} &R(i,j): on-demand, Elasticity, Resource\ pooling\ , Measured\ services\ = \\ &\quad R(i,j): on-demand * W1 + R(i,j): elasticity * W2\ + \\ &\quad R(i,j): resourcePooling * W3 + R(i,j): measuredServices\ * W4 \qquad (5.2) \end{aligned}$$

Where *W1*, *W2*, *W3*, *W4* are weights with $W1 + W2 + W3 + W4 = 1$and R(i,j): on-demand, R(i, j):elasticity, R(i, j):resourcePooling, R(i,j):measuredService are the dimension considered in the trust model based on the cloud characteristics.

Reliability of a single dimension is given as:

$$R(i,j): on-demand = R(i,j): availability\_d\ * W11 + R(i,j): timely\_d * W12 \qquad (5.3)$$

$$R(i,j): Elasticity = R(i,j): availability\_e *\ W21 +\ R(i,j): timely\_e *\ W22 \qquad (5.4)$$

$$R(i,j): Resource\ pooling = R(i,j): afinity\ \ *\ W31 +\ R(i,j): legal\ *\ W32 \qquad (5.5)$$

$$R(i,j): Measured\ Services = \begin{matrix} R(i,j): viewable\ *\ W41 \\ + \\ R(i,j): controllable\ *\ W42 \\ + \\ R(i,j): reportable\ *\ W43 \end{matrix} \qquad (5.6)$$

Where *W11*, *W12*, *W21*, *W22*, *W31*, *W32*, *W41*, *W42*, *W43* are weights assigned such that *W11 + W12=1, W21+W22=1*, *W31+ W32=1* and *W41+ W42+ W43=1*.

Reliability of a single feature can be given as the expectation of the opinion. For example below the reliability of the available on demand resource is given as:

$$R(i,j):availability\_d= Exp(W^{i}_{availability\_d}) \tag{5.7}$$

Where $W^{i}_{\text{availability\_d}}$ is the opinion of entity *i* for the feature *availability_d,* for its direct interaction with entity *j*. $W^{i}_{\text{availability\_d}}= (b^{i}_{\text{availability\_d}}, d^{i}_{\text{availability\_d}}, u^{i}_{\text{availability\_d}}, a^{i}_{\text{availability\_d}})$, where $b^{i}_{\text{availability_d}}$is the belief in the proposition, $d^{i}_{\text{availability\_d}}$ is the disbelief in the proposition, $u^{i}_{\text{availability\_d}}$ is the uncertainty of the proposition, $a^{i}_{\text{availability\_d}}$ is base rate that provides the weight of uncertainty that contributes to the probability expectation (P.S. Pawar et al., 2012).

In circumstances where data related to any of the dimension or the feature is not available, the corresponding weights can be readjusted such that zero weight is assigned to the dimension or feature for which data is not available and the overall weight can be distributed amongst the dimensions or features for which data is available. This enables the trust model to cope with the missing data.

5.3.3 Reputation Trust

The reputation trust is calculated based on the feedbacks received from the other entities in the system. Rep (i,j) is the reputation trust of entity *j* from the perspective of entity *i*. The CBR (entity *i*) receives feedback from all SPs their reliability trust about entity *j* for each feature of the dimension and computes the reputation trust Rep (i,j) for each feature. The overall Reputation trust of entity *j* from the perspective of entity *i* for all the dimensions is given as the weighted average:

$$\begin{aligned} Rep(i,j)&: on-demand, Elasticity, Resource\ pooling\ , Measured\ services = \\ & Rep(i,j): on-demand * W1\ + \\ & Rep(i,j): elasticity * W2\ + \\ & Rep(i,j): resourcePooling * W3 + \\ & Rep(i,j): measuredServices\ * W4 \end{aligned} \qquad (5.8)$$

Where $W1$, $W2$, $W3$, $W4$ are weights with $W1 + W2 + W3 + W4 = 1$ and *Rep(i,j):on-demand,Rep(i,j):elasticity,Rep(i,j):resourcePooling, Rep(i,j):measuredService* are the dimension considered in the trust model based on the cloud characteristics.

The Reputation trust for each dimension is based on the features available for the dimension and is given as follows:

$$Rep(i,j): on-demand = Rep(i,j): availibility_d * W11 + Rep(i,j): timely\_d * W12 \qquad (5.9)$$

$$Rep(i,j): Elasticity = Rep(i,j): availibility_e * W21 + Rep(i,j): timely\_e * W22 \qquad (5.10)$$

$$\mathrm{Rep(i,j): Resource\ pooling} = Rep(i,j): \mathrm{afinity}\ * \mathrm{W31} + Rep(i,j): \mathrm{legal}\ * \mathrm{W32} \qquad (5.11)$$

$$\mathrm{Rep(i,j): Measured\ Services} = + \begin{matrix} Rep(i,j): \mathrm{viewable}\ * \mathrm{W41} \\ Rep(i,j): \mathrm{controllable}\ * \mathrm{W42} + \\ Rep(i,j): \mathrm{reportable}\ * \mathrm{W43} \end{matrix} \qquad (5.12)$$

Where *W11*, *W12*, *W21*, *W22*, *W31*, *W32*, *W41*, *W42 andW43* are weights assigned such that *W11+ W12=1*, *W21+W22=1*, *W31+ W32=1* and *W41+ W42+ W43=1.*

The reputation trust for each feature identified for the dimension is given by first discounting or weighing the feedback with the credibility for the feedback provider and then taking consensus view of all the discounted opinion. For example the reputation trust for the availability feature of on-demand dimension is given as:

$$Rep(i,j):availability\_d = Exp((W^{k1}{}_{(availability\_d} \otimes W^{k1}{}_{(credibility)}) \oplus$$

$$(W^{k2}{}_{(availability\_d} \otimes W^{k2}{}_{(credibility)}) \oplus \dots \oplus$$

$$(W^{kn}{}_{(availability\_d} \otimes W^{kn}{}_{(credibility)})) \qquad (5.13)$$

Where $W^{kl}{}_{\text{availability\_d}}$, is an opinion of entity *k1,* based on its direct interaction with entity *j,* for the feature *availability_d*. The symbol $\oplus$ is the consensus operator is given in (Jøsang, 2001). $W^{k1}{}_{\text{credibility}}$ is credibility opinion for entity *k1*, as built by entity *i*, based on the trueness of feedback received.

5.3.4 Credibility

The credibility is the trust in the feedback provider from the trustor's perspective. This enables the trustor to weight the information provided by the feedback provider about the trustee. The credibility is given as follows:

$$W^{k}{}_{new\ credibility} = W^{k}{}_{current\ credibility} \underline{\otimes} W^{k}{}_{previous\ credibility} \qquad (5.14)$$

$$cv = 1 - |F_{kj} - Q_j| \qquad (5.15)$$

$$W^{i}{}_{current\ credibility} = f(cv) \qquad (5.16)$$

Where $\underline{\otimes}$ is a consensus operator to combine dependent trust as defined by Jøsang (Jøsang et al., 2006) and *cv* is credibility value which is used to build the current credibility opinion. The *cv* forms the positive evidence and *(1-cv)* provides the negative evidence to build the current credibility opinion $W^{k}{}_{\text{current credibility}}$. F_{kj} is the feedback response provided by witness k about trust j and the Q_j is the real QoS by trustee j. The initial value of the credibility is set to a high belief of 1.0.

Let $W^{A_i}_x = (b^{A_i}_x, d^{A_i}_x, u^{A_i}_x, a^{A_i}_x)$ where i€ [1,n], be n dependent opinions respectively held by agents A$_1$, …,A$_n$ about the same proposition x. The depended consensus is then $W^{A_1 \ldots A_n}_x = (b^{A_1 \ldots A_n}_x, d^{A_1 \ldots A_n}_x, u^{A_1 \ldots A_n}_x, a^{A_1 \ldots A_n}_x)$ where (Jøsang et al., 2006):

$$b^{A_1 \ldots A_n}_x = \sum^n_1 (b^{A_i}_x / u^{A_i}_x) / (\sum^n_1 (b^{A_i}_x / u^{A_i}_x) + \sum^n_1 (d^{A_i}_x / u^{A_i}_x) + n) \qquad (5.17)$$

$$d^{A_1 \ldots A_n}_x = \sum^n_1 (d^{A_i}_x / u^{A_i}_x) / (\sum^n_1 (b^{A_i}_x / u^{A_i}_x) + \sum^n_1 (d^{A_i}_x / u^{A_i}_x) + n) \qquad (5.18)$$

$$u^{A_1 \ldots A_n}_x = \mathrm{n} / (\sum^n_1 (b^{A_i}_x / u^{A_i}_x) + \sum^n_1 (d^{A_i}_x / u^{A_i}_x) + n) \qquad (5.19)$$

$$a^{A_1 \ldots A_n}_x = \sum^n_1 a^{A_i}_x \; / n \qquad (5.20)$$

As the initial credibility is set to 1, this credibility is transformed to an opinion with a function *f*. The function *f* first converts the credibility value *cv* into positive evidences (s) and negative evidences (r) given as: $s=cv * n$ and $r= (1-cv) *n$. And later the function *f* computes the opinion based on the positive and negative evidences as in section 4.3.1. The value *n* signifies the total amount of evidence and any high value of *n* will create lower uncertainty. For practical purpose, n=100 can be considered.

Let us consider an example to compute a new credibility of a feedback provider *k*. Consider that the feedbacks are provided as 1-Excelent, 0.75- good, 0.5-average, 0.25-bad and 0-worse. If feedback from consumer *k* for a service provider *j* is obtained as F_{kj} =0.75 and after consuming the service it is observed that the Quality of service obtained from the service provider is bad which is given as Q_j=0.25. This results in the reduction of the credibility for the feedback provider as the feedback provided is deviating from the actual service received. The *cv* value is computed as cv=1-|0.75-0.25| = 0.5. The *cv* value of cv=0.5 is used to compute the current credibility opinion of the feedback provider. Considering n=100, we obtain the positive and negative evidences as *s=50* and *r=50*. Using the opinion representation as in section 4.3.1 we obtain the current credibility opinion as $W_{\text{curent credibility}}$ = (0.48, 0.48, 0.039). Based on the previous *cv* that is cv=1, the previous credibility opinion is given as $W_{\text{previous credibility}}$ = (0.99, 0, 0.009). The new credibility opinion is computed as dependent consensus which is given as;

$$W_{\text{new credibility}} = W_{\text{previous credibility}} \underline{\otimes} W_{\text{current credibility}}$$

$$= (0.99, 0, 0.009) \underline{\otimes} (0.48, 0.48, 0.039)$$

$$= (0.887, 0.096, 0.015)$$

It can be observed that the belief in the new credibility is reduced compared to the previous credibility due to the high difference in the feedback provided and the QoS obtained after the service is consumed. For the example we use the feedbacks as five discrete values, but any values can be used in the range of 0-1.

5.3.5 Filtering Unfair Ratings

The Reputation trust depends mainly on the feedbacks provided by the providers. The feedbacks have significant impact on the trust computation of the trustee, especially when the confidence level of the trustor is low. In systems with large number of feedback providers, the malicious groups of feedback providers may significantly impact the reputation and the trust value computed for the trustee. Many studies (Dellarocas, 2000; Jia et al., 2012; Whitby et al., 2004; Yang et al., 2009) exists to show how to reduce the effect of the malicious feedback providers. The study in this chapter uses three categorized groups of malicious feedback provider as considered in (Jia et al., 2012). The malicious groups are: complementary, exaggerated positive and exaggerated negative.

$$\begin{aligned} \mathrm{R}'^{(\mathrm{i,j})} &= 1 - \mathrm{R}(\mathrm{i,j}) && \text{complementary} \\ &= \alpha + \mathrm{R}(\mathrm{i,j})(1-\alpha) && \text{exaggerated positive} \\ &= \mathrm{R}(\mathrm{i,j})[1 - \alpha/(1-\alpha)] && \text{exaggerated negative} \qquad (5.21) \end{aligned}$$

Where R(i,j) is the reliability trust of entity j, α is the degree of exaggeration coefficient. In this chapter we focused mainly to demonstrate a case where the filtering of the malicious feedback providers significantly improves the robustness of the trust model. This improvement is complementary to the robustness achieved using the credibility metrics. Though any techniques of excluding malicious feedback providers are applicable, we demonstrate our model using the outlier method to filter the exceptions in the feedback (Arning et al., 1996). In this approach, the outlier is defined as the feedbacks that are inconsistent with majority of the feedbacks and has low probability that it originated from the same statistical distribution as other feedbacks in the overall set of feedback. This work has been initially discussed in the context of detecting of outliers in large databases (Arning et al., 1996). The work in this chapter uses the basic optimal algorithm (Zhang and Feng, 2009) defined to find the subset with maximum smoothing factor which primarily is dependent on the outlier detection algorithm(Arning et al., 1996) in large databases.

5.4 Conclusion

This chapter presents an extended trust model that comprehensively captures the cloud characteristics and enables the trust evaluation for a cloud infrastructure service provider. The trust model considers the cloud characteristics as dimensions and identifies several features associated with the dimensions. The trust model primarily uses the opinion model for creating context specific features. The trust framework proposes to consider an early malicious filter which along with the credibility defined in the trust model helps in enhancing the robustness of the model against malicious feedbacks.

In this chapter we proposed the use of cloud broker that is used to evaluate the trustworthiness of the cloud service provider. In the next Chapter 6, we propose a detailed architecture of the cloud broker.

Chapter 7 provides the evaluation of this extended trust model. The extended trust model proposed in this chapter is evaluated using experiments to verify the robustness of this model against malicious feedback providers.

Chapter 6 Cloud Broker Based Security Reputation

This chapter describes the proposed cloud broker architecture that provisions the trustframework with the surrounding capabilities such as monitoring, SLA framework etc for performing trust evaluations of the cloud service providers. The chapter describes the cloud broker architecture and the four modes of operation: *recommender*, *intermediary*, *aggregation* and *arbitration*.

In addition to the detailed cloud broker architecture, this chapter also presents the use of cloud broker architecture model that enables us to build a security reputation framework for cloud service providers, capturing comprehensive evidence of security information to build its trust and security reputation.

6.1 Introduction

Cloud computing provides flexible and dynamic access to virtualised computing and network resources that can be provisioned in real-time with minimum management effort and service provider interactions (Mell and Grance, 2011). Due to the desirable properties of low maintenance costs, flexibility, scalability, and virtualisation, cloud computing has become ‘the’ platform of choice for deploying all sorts of applications and software solutions. As a result, an end-user can encounter many cloud service providers offering a multitude of services, with each cloud provider offering its own application programming interface (API), specialised services, billing utility, and security functionalities, in order to satisfy various user requirements.

Current cloud market consists of large number of cloud service providers, with a variety of cloud services. This offers the cloud service consumers with flexibility and choice for selecting the cloud service providers for their service. However, selecting the cloud service providers pose challenges such as (Sundareswaran et al., 2012) : 1) No standard representation of cloud properties 2) No standard adopted by cloud providers for negotiating and agreeing the requirements. SLA of cloud providers may vary in format and content. 3) The cloud user may have service requirements that cannot be fulfilled by a single cloud provider.

In situations, when consumers/users want to deploy applications on multiple cloud platforms, they face considerable challenges due to the interface diversity and architectural differences in these cloud platforms. Hence it is important that the cloud platforms are able to interoperate to provide the best tailor-made services as requested by the users. However, due to the current lack of comprehensive cloud interoperation standards or the lack of implementation in the few cases where such standards exist, the interoperability in the cloud is usually very difficult to obtain for the end customers.

To overcome this difficulty, there is a need to have an additional computation layer that enables discovery, mediation, monitoring, interoperability and also management of the services. The cloud brokerage provides this additional layer that eases the use of cloud services and also provides value additions for the services deployed in the cloud. Recent studies in cloud computing environment have advocated that the use of cloud brokers provide several advantages (Nair et al., 2010) (Li et al., 2012). In (Nair et al., 2010) the authors affirm the use of cloud broker as cloud service intermediation, cloud service aggregation and cloud service arbitrage. On the other hand (Li et al., 2012) suggested the use of cloud brokers to handle the complexity of prioritization and selection of cloud infrastructure as service provider.

6.1.1 Cloud Broker Value Chain

Similar to any brokerage system, the cloud broker is an intermediary between the consumers/user and the cloud service providers. In (Nair et al., 2010)(Gartner, n.d.), the proposed use of cloud broker is as 1) *cloud service intermediation*: intermediation for multiple services to add value-additions like identity management or access control, 2) *cloud service aggregation*: bringing together two or more fixed cloud based services, and 3) *cloud service arbitrage*: similar to cloud service aggregation, but providing a more dynamic aggregation to support flexibility. Gartner (Gartner, n.d.) provides clear descriptions of these categories of cloud broker while in (Nair et al., 2010) provides a high level architecture of the cloud broker that fulfils the requirements of these categories. The work in (Nair et al., 2010) proposes the use of cloud broker at an abstract level without any concrete architecture to realize the proposed functionalities.

In this chapter we propose the architecture of a Cloud Broker (CBR) that simplifies the complexity of provisioning, integrating, and administering a cloud service on multiple cloud platforms.

This chapter proposes the cloud broker architecture design and implementation to use the cloud broker as (1) *cloud service recommendation* (2) *cloud service intermediation* (3) *cloud service aggregation* and (4) *cloud service arbitrage*.

CBR used in *cloud service recommendation* enables the consumer/user to get recommendations from the CBR about the most suitable cloud infrastructure provider for hosting their service, based on the degree of Trust, Risk, Eco-efficiency and Cost (TREC) (Ferrer et al., 2012; "OPTIMIS Toolkit," n.d.). The CBR as a *recommender* reduces the effort of the consumer to identify the suitable cloud service provider for its service, but the actual deployment of the service to the cloud infrastructure is performed by the consumer after obtaining deployment solution from the CBR.

CBR used as *cloud service intermediation* provides management functionalities like Value Added Services (VAS) that are cloud provider specific and which may be essential for the consumer's service that is deployed in the cloud provider environment. This chapter considers the two security services Intelligent Protection System (IPS) and Secure Storage Service offered as value additions on top of the services delivered. As an *intermediary*, the CBR also takes complete responsibility of the consumer's/user's services to identify the most suitable Infrastructure Provider (IP) based on TREC, performs the deployment on the selected IP, and manages smooth functioning of the service at its operational stage.

The use of CBR as *cloud service aggregation* provides management functionalities for multi-cloud deployment and operation of a service by combining the multiple cloud infrastructure provider services. The CBR also provides VASs that are independent of cloud providers. In this chapter the CBR that acts as virtual cloud provider and also provisions Virtual Private Network (VPN) overlay as a VAS that is established dynamically between the service components deployed across multiple cloud providers.

	Deployment Solution	Deployment of Service	Provider specific VAS	Provider Independent VAS	Static Multi-cloud deployment	Dynamic multi-cloud
Recommender	X					X
Intermediary		X	X			
Aggregator	X	X	X	X	X	
Arbitrage	X	X	X	X	X	X

Table 6.1 : Features for cloud broker used in different modes

CBR used as cloud service arbitrage can be considered as dynamic aggregation wherein the multi-cloud deployment of consumer service is dynamically decided based on the service requirements. In this chapter, the CBR decompose the service requirements at component level and negotiates with multiple cloud providers for each of the service components to formulate an optimized deployment solution taking into account the basic service requirements as well as additional requirements such as trust, risk, eco-efficiency, cost, compliance and security.

Table 6.1 summarizes the features for CBR used in different modes.

The remaining chapter is structured as follows. Section 6.2 provides a cloud broker scenario which considers a Genomic application and its requirement to be deployed in the cloud environment using a cloud broker. Section 6.3 describes the Cloud broker architecture and its implementation. Section 6.4 describes the cloud broker used as recommender. Section 6.5 describes the Cloud used as arbitrage along with the deployment scenario of the Genomic application. Section 6.6 introduces the security reputation and Section 6.7 provides the Cloud broker architecture enabled for security reputation. Section 6.8 describes our approach of the reputation modelling to build the security reputation of the cloud service provider. Section 6.9 provides concluding remarks.

6.2 Cloud Broker Scenario

6.2.1 Genomic Application

To illustrate the cloud broker architecture and implementation, we consider a Genomic application that is used across the deployment scenarios described in this chapter (Royo et al., 2008).

The genetic information of patients is a key for an efficient treatment of several diseases and a genomic application considered here helps in the identification of genes which cause a disease. The successful identification of genes in an automatic way provides scientists with valuable information that allows them to perform functional analysis at all levels. The genomic application implements a combination of different existing genomic services with sequence comparison algorithm to help on gene detection from genomic DNA sequence. A composition of these services is invoked to obtain the reference data and prepare the DNA sequence in the suitable format for the computation. This computation calculates the comparison of the pre-processed DNA sequence with the reference data which identifies the most relevant genes. For each of these genes a deep analysis is performed and its results are post-processed with other genomic services producing the final report delivered to the researchers.

This genomic application is implemented as a service using the programming model and IDE. The OPTIMIS Programming Model (PM) simplifies cloud enablement of new applications by offering a run-time programming model which can be optionally used with Eclipse-based IDE ("OPTIMIS Programming Model plugin," n.d.). By introducing an abstraction layer, the OPTIMIS PM makes application development generic and independent of the underlying cloud infrastructure interfaces. It simplifies cloud application development by a simpler programming model based on sequential specifications of data and performance compliance described in the OPTIMIS service manifest. A run-time model provides optimal parallelism and multi-cloud distribution and performs run-time scheduling and optimization during application execution. PM IDE supports creation of a service manifest. It also interacts with the image creation service for the creation of composite services according to the service manifest.

The genomic application contains five different components. 1) Genome Formatting 2) Similar Sequence Retrieval and Format 3) Relevant Gene Search 4) GeneWise 5) Genome Application GUI.

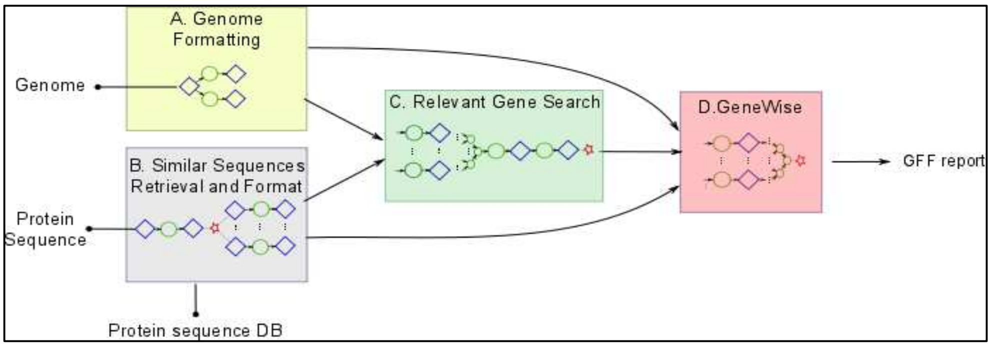


Figure 6.1 : Components of Genomic Application

The consumers have the following requirements for deploying the genomic application in the cloud: **requirement-1** : For privacy reasons, the consumer requires that some of the crucial components are distributed across multiple cloud providers; **requirement-2** : The application architecture demands that some components need high affinity and are required to be deployed in the same IP and at the same time requires anti-affinity between some components so that these components should never be in the same cloud provider; **requirement-3** : Apart from the resource requirements of each component, the consumer expects capabilities such as high level of trust and eco-efficiency and low risk and cost from the cloud service providers; **requirement-4**: The data provided to the application needs to be protected against the data protection law and hence legal requirements related to IPR and provider location are posed by the consumer; **requirement-5**: The consumer is also concerned of the protection of the service and the data, as it will be deployed in the cloud environment and have security requirements for the genomic application components; **requirement-6**: The genomic application components require communication and expects secure communication between the application components. This requirement is very generic and independent of any cloud provider while the requirement-4 and requirement-5 are very specific to the cloud providers; **requirement-7**: Since the application may be required to be deployed in multiple providers, the selection of cloud providers should be dynamic, based on the fulfilment of the resource requirements and on the capabilities.

Considering the multi-dimensional requirements of the consumer for the genomic application, the consumer may approach the CBR to be used as either *recommender* to get recommendations of deployment or as intermediary/aggregator/arbitrage to offload the responsibility of deployment to the CBR.

CBR used as cloud service recommendation: In this mode of the CBR operation, the consumer gets recommendation on the deployment solution from the CBR and performs the deployment and operational management of the application itself. In this case the CBR can take into account the **requirement-1**, **requirement-2, requirement-3, requirement-4** and **requirement-7** devising the deployment solution but **requirement-5** and **requirement-6** (security) are required to be realized by either the consumer or by the cloud service provider. Also the deployment and the operation management of the application is the responsibility of the consumer.

CBR used as cloud service intermediation: The CBR used as service intermediation provisions the VAS that is provider specific and hence it can fulfil **requirement-5** concerned with the security of service and data. Multi-cloud functionality is beyond the scope of intermediation which disallows the fulfilment of the **requirement-6** that expects secure communication across the components.

CBR used as cloud service aggregation: The CBR used as cloud service aggregation is capable of multi-cloud deployments and also contains all the capabilities of CBR as recommendation service. This allows the fulfilment of all the requirements except for **requirement-7** that expects the dynamic selection of the cloud providers. Due to the multi-cloud capabilities, selected VAS can be applied to the appropriate provider. For the **requirement-6** a virtual overlay can be created across all providers for secure communication between application components.

CBR used as cloud service arbitrage: CBR used as cloud service arbitrage includes the aggregation capabilities and also contains functionality of dynamic selection and management of multi-cloud services. CBR in this mode of operation is capable of fulfilling all the requirements of the consumer for the Genomic application.

Table 6.1 Table 6.2 lists the requirements that can be fulfilled by the CBR in different modes of operation.

	Recommendation	Intermediation	Aggregation	Arbitrage
requirement-1	X		X	X
requirement-2	X		X	X
requirement-3	X		X	X
requirement-4	X		X	X
requirement-5		X	X	X
requirement-6		X	X	X
requirement-7	X			X

Table 6.2 : Requirements of Genomic application fulfilled by cloud broker used in different modes

6.3 Cloud Broker Architecture

The cloud broker (CBR) architecture proposed in this section has multiple benefits. The key aspects of the CBR include: 1) Maximization of the user choice; 2) Multi-tier reseller model and user driven customization; 3) Provision of services on multi-tier reseller model; 4) Harmonization of high-value enhancements.

The maximization of the user choice (case (1)) is provided by a) having multi-cloud deployment model support that enables cloud site selection, b) broker intermediated agreements that avoid vendor lock-in, and c) broker-based optimization for the selection of cloud site.

The CBR support for multi-tier reseller model and user driven customization (case (2)) is achieved by supporting a) broker arbitrage architecture, b) service manifest decomposition, and c) Trust, Risk, Eco-efficiency and Cost (TREC) based provider selection.

The CBR provides services on multi-tier reseller model (case (3)) by incorporating into the architecture that support for a) automated value-added services integration and b) secure overlay across the infrastructure providers.

The CBR harmonizes high-value enhancements (case (4)) by integrating into the consumer services that support for (a) security, (b) data protection, (c) TREC based optimization and (d) broker enabled horizontal elasticity.

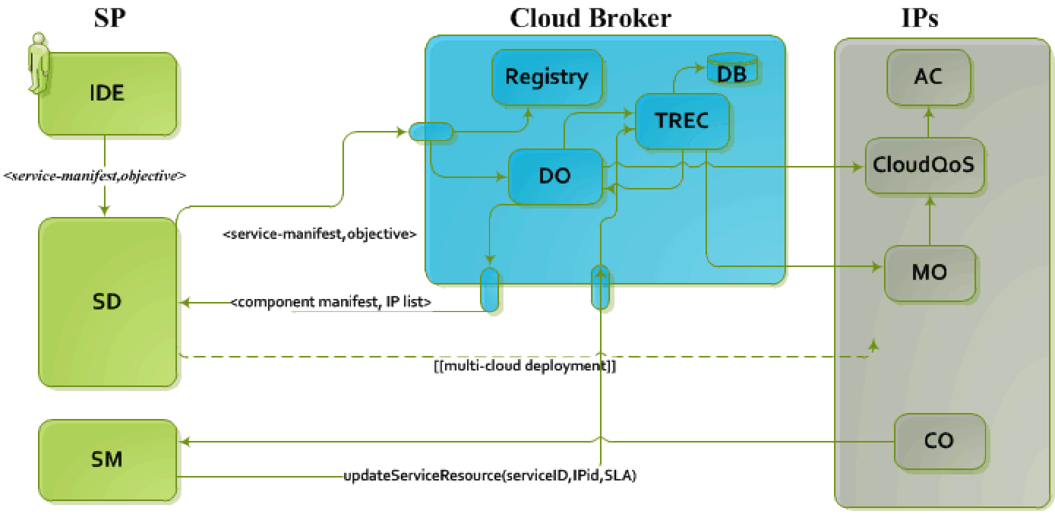
The CBR enables the use of multiple infrastructure services by integrating them so as to implement a singular cloud service or process. The CBR architecture supports value-added services and serves as new business and deployment model for a "virtual infrastructure provider" where it can offer value-add on top of assembly of wholesale offerings from different cloud providers.

Inline with the different modes of CBR as discussed in section 6.1.1, the cloud broker architecture is developed using OPTIMIS toolkit components ("OPTIMIS Toolkit," n.d.), which can be used as *recommendation*, *intermediation, aggregation or as arbitrage*. Section 6.4 describes the architecture for a cloud broker as a recommender. The functionality of intermediation and aggregation is included in the cloud broker as abitrage which is enabled with capabilities of static/dynamic aggregation and provisioning of VAS. Hence no separate section on intermediation and aggreation is provided however these capabilities are discussed in section 6.5 which describes the cloud broker as arbitrage.

To describe the CBR architecture, a deployment scenario of Genomic application discussed in section 6.2.1 is considered. Specifically for the scenario, the IP registry contains multiple cloud providers, but only two cloud providers (Provider-1 and Provider-2) meet the requirements of the application and the application is deployed in these two providers. The the CBR service is hosted in the BT (British Telecom) Cloud.

6.4 Cloud Broker Used as a Recommender

Figure 6.2 shows the high level sequence of operations perfomed by CBR as a recommender.


SP
Cloud Broker
IPs
IDE
Registry
DB
TREC
DO
AC
CloudQoS
MO
<service-manifest,objective>
<service-manifest,objective>
SD
<component manifest, IP list>
[[multi-cloud deployment]]
SM
CO
updateServiceResource(serviceID,IPid,SLA)

IDE – Integrated Development Environment
SM – Service Manager
DB – Database
AC – Admission Control
CloudQoS – Cloud Quality of Service
SD – Service Deployer
DO – Deployment Optimizer
TREC – Trust, Risk, Eco-efficiency, Cost
CO – Cloud Optimizer
MO – Monitoring

Figure 6.2: **High level sequence diagram for broker as recommender**

The advantage of this architecture is that the consumer interacts with CBR only for getting the best choice of cloud providers to deploy the Genomic application. As the Genomic application have multiple components, the solution recommeded by the broker may be a multi-cloud deployment of the service components. Based on the solution obtained, the consumer may do the multi-cloud deployment to the cloud providers.

The CBR as recommender will perform the following high level steps

- Create Service manifest and construct the application
- Pass the service manifest to the CBR
- Perform Legal Check
- Get deployment solution

The consumer takes the responsibility to interact with the cloud providers based on the solution provided and perform service contextualization and upload data and finally create agreement. The consumer may either approach the cloud provider for any value addtions required for the service or take self responsibilty to provision these value additions.

In the following we describe the sequence of steps performed by the CBR. More details of the components used can be obtained from the Chapter 1Appendix C: and from the OPTIMIS toolkit(“OPTIMIS Toolkit,” n.d.).

Step 1: The SP creates the Genomic service using IDE (Integrated Development Environment) which in turn invokes the *ICS* (Image Creation Service) for service creation (shown in Figure 6.3). The IDE also supports creation of service manifest. The Genomic service created is in the form of VM images for each of the components; i.e., five VM images are created.

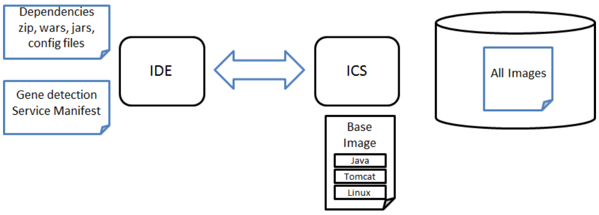

Dependencies zip, wars, jars, config files
Gene detection Service Manifest
IDE
ICS
Base Image
Java
Tomcat
Linux
All Images

Figure 6.3: **Image Creation Service**

Step 2: The IDE passes the *service manifest* and the optimization objective (trust, risk, eco-efficiency or cost) to the SD for deployment of the service.

Step 3: The SD uses the cloud broker interface to submit the *service manifest* and the optimization objective.

Step 4: The CBR has a Registry where all SPs and IPs register for using the CBR services.

Step 5: The CBR after receiving a request for deployment of a service, gets the list of IPs from the Registry.

Step 6: The TREC component of the broker contains the historical assessments of all SPs and IPs stored in the DB. The DO individually interacts with the TREC components to get the TREC assessment for each of the IPs in the IP registry.

Step 7: The DO also decomposes the *service manifest* received and evaluates, for each component, the suitability of the IPs based on the TREC levels expected by the components and the historical TREC assessment of IPS.

Step 8: The IPs that do not meet the TREC criteria specified in the service manifest are filtered.

Step 9: For the filtered list of IPs, the DO initiates SLA negotiations to receive offers from the IPs for the application to be deployed.

Step 10: In the process of negotiation, the CBR interacts with the AC which checks its current infrastructure status and the requirements of the Genomic application based on which it provides the offers.

Step 11: Once all the offers for all the components of the service is received the CBR applies the optimization algorithm to provide the SP with the ranked list of IPs for each of its service components based on the TREC.

Step 12: The SP deploys all its components considering the ranked list.

Step 13: The service is deployed using the CO at the IP side. The CO provides all VM related information to the SP, which in turn is forward to the CBR.

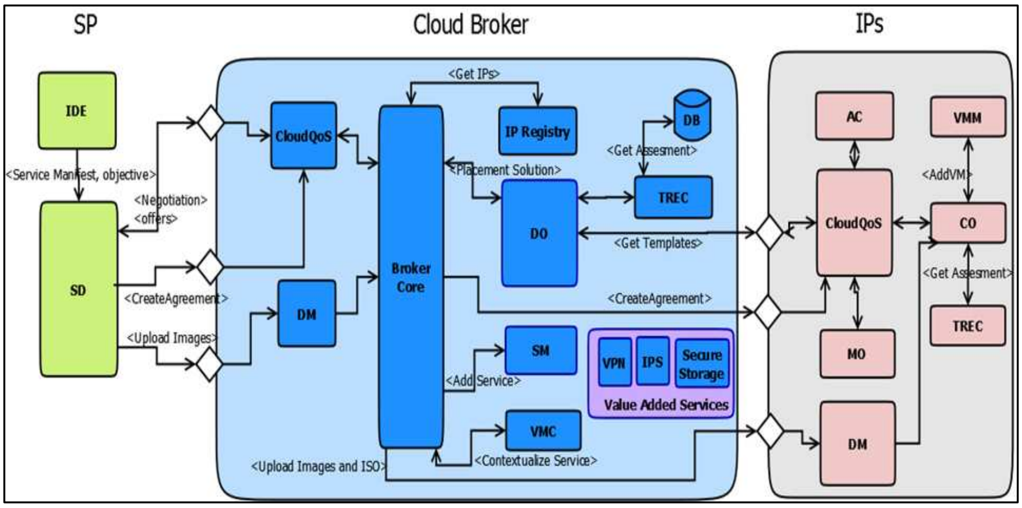
Step 13: The CBR passes the VM information to the TREC components to receive monitoring events for these service components.

The advantage of this architecture is that the SP interacts with CBR only for getting the best choice of IPs to deploy the Genomic application. As the Genomic application have multiple components, the solution recommeded by the cloud broker may be a multi-cloud deployment of the service components. Based on the solution obtained, the SP may carry out the multi-cloud deploymen to the IPs.

6.5 Cloud Broker Used as an Arbitrage

The CBR used as arbitrage provides an outlook of a cloud provider, to the consumer that wants to deploy its service. CBR used as arbitrage not only takes the responsibility of finding the optimal solution for the service (as recommender), but also performs the multi-cloud deployment (aggregation) and provides Value Added

Services (intermediation). The CBR also takes charge of the service during operational mode to monitor its performance and take critical decisions such as scale up/down, start/stop, or relocate the service components, reducing the complete workload of the consumer to monitor its service. The architecture of the cloud broker shown in Figure 6.4 used as arbitrage can be well explained with the deployment scenario of the Genomic application described in section 6.2.1.



IDE – Integrated Development Environment
SM – Service Manager
DM – Data Manager
DB – Database
VPN – Virtual Private Network
CloudQoS – Cloud Quality of Service
CO – Cloud Optimizer
VMM – Virtual Machine Manager

SD – Service Deployer
DO – Deployment Optimizer
VMC- Virtual Machine Contextualizer
TREC – Trust, Risk, Eco-efficiency, Cost
IPS – Intelligent Protection System
MO – Monitoring
AC – Admission Control

Figure 6.4 : High level component architecture of the Cloud Broker

In the following we describe the service deployment steps performed by the CBR used as arbitrage:

Step 1: Create Service manifest and construct the application: The PM-IDE ("OPTIMIS Programming Model plugin," n.d.) allow to specify the application requirements such as cpu, memory, disk requirements, TREC(Trust, Risk, Ecoefficience and Cost), elasticity, affinity, anti-affinity, security and legal constraints, to create the service manifest.

The consumer creates the Genomic service using the *ICS* (Image Creation Service) using the service manifest, as shown in Figure 6.3.

Step 2: Initiate negotiation with CBR

The programming model IDE ("OPTIMIS Programming Model plugin," n.d.) send the *service manifest* to the cloud broker to negotiate the service terms with the cloud broker.

Step 3: Perform Legal Check

The IP registry at the CBR contains a list of cloud providers. Each of the cloud infrastructures in the IP registry is verfied against location constraint specified in the service manifest, using the Data Manager (Kousiouris et al., 2011) service to perform legal check.

Step 4: Get deployment solution

The CBR provides list of legally compliant cloud providers and the service manifest, to the DO (Deployment Optimizer) (Li et al., 2012) for getting optimal solution for deploying the Genomic service. To obtain an optimal solution, the DO mainly performs two steps: 1) Decompose the service manifest using the constraints in the service manifest 2) Negotiate the decomposed manifest with the cloud providers and check for TREC (Trust, Risk, Eco-efficiency and Cost) constraints specified by the consumer.

Step 5 :Data upload and VM contextualization

The CBR uses the decomposed manifest and uploads the service images to the DM of the respective cloud providers. The CBR further performs VM contextualization (Armstrong et al., 2011) which bundles all the necessary configuration scripts in the form of an ISO files which also includes value added security services.

Step 6: Agreement creation

The final stage of the deployment process is the creation of the agreement. The SP initiates a *create agreement request* to the CBR. The CBR gets the context of this request and further follows with creating agreements with the multiple-cloud providers. The successful agreement creation of CBR with the cloud providers, start the Genomic service VMs in the running mode.

6.5.1 Use of Trust Model in Cloud Broker:

Service deployment: The SP places all the infrastructure requirements of a service in the service manifest which also includes additional trust requirements for the infrastructure provider. The SP sets minimum trust level expected for an IP that should be fulfilled including other resource requirements. This minimum expected trust level is utilized by DO (Deployment Optimizer) component of the cloud broker to negotiate the service level agreement with the IPs. The IPs that meet all resource requirements and the minimum trust level requirement, are shortlisted by the DO as a possible solutions for deployment of the service of the SP.

Service operation: During the service operation, the trust module on cloud broker continually monitors and records the trust level of the IP. Any failure to meet the resource requirements as per the agreed SLA for any of the SP, may lead to reduction of trust level for the IP. At regular intervals the cloud broker examines the expected minimum trust level against the recorded trust level. If the trust level for IP decreases below minimum expected trust level, the cloud broker prepares for an alternative solution such as redeployment of the service to another IP with high trust level and that may fulfil all the resource requirements.

6.6 Security Reputation

Since security remains a major concern in the use of cloud services, an individual or an enterprise expects a high level of confidence and trust in the cloud service provider it would like to use. The enterprise needs a process to identify and decide on the most suitable service provider to fulfil its security requirements for its service to be deployed. Reputation systems have been effectively used in making such decisions, however it is highly challenging to apply the concept to the cloud ecosystem, with a security context. This is challenging mainly due to the reluctance of the cloud service providers to publicize their security related information to the internet community or even to a selected group of customers. Relevant information may include events or incidence recorded due to security activities like firewall filtering, intrusion detection/prevention systems, security policies, authentication/authorization, identity and key management.

However one also need to keep in mind the fact that IT service providers have been providing details of their security systems and associated processes to third party (security) auditors for obtaining security certifications and legal compliance status. These certifications are often essential requirements of the service provider to gain confidence of their customers and the industry as a whole. In order to obtain security certification the service provider needs to share, among other details, the security event related information to the third party auditors. The higher the level of security certification required, the more critical is the security events information and process details expected by the auditors. In order to avoid security leakage it is a common practice to obtain non-disclosure agreements with auditors before this critical security information are shared. An enterprise needing cloud services have to rely on the security certifications of the cloud service providers to establish trust in the providers. This approach however constraint the enterprise to match their security requirements based only on the certification information published by the service providers and the associated minimum requirements that needs to be met by the service provider for obtaining the certification, due to unavailability of other detailed information.

As a way of breaking this impasse we propose the use of a Cloud Broker (CBR) that inherits and expands on the role of the security auditor, enabling the broker to obtain access to the security events due to the high trust placed by the service providers, which may not be possible with the wider community. The CBR provisions the enterprises with security reputation of the cloud service providers based on their security requirements as specified to the CBR. The registration with the broker allows the cloud service providers to highlight their security strengths without exposing their internal security details like event information to the wider customer base and at the same time also benefited by CBR's potentially wider customer base. The cloud service consumers benefit from the service that provides a closest match between their security requirements and the security reputation of the cloud service providers.

6.7 Cloud Broker Architecture Enabled for Security Reputation

We introduce a Cloud Broker architecture that enables building of security reputation of individual service provider and sharing the same with its customers. The proposed broker architecture is shown in Figure 6.5 that includes various components namely: (i) *Infrastructure Provider Interface (IPI)* (ii) *Service Provider Interface (SPI) (*iii) *Monitors (M)* and (iv) *Trust Engine (TE)*. The entities involved in the architecture are Infrastructure Providers (IP) and Service Providers (SP). The IP and the SP register with broker. The registration of the IP at the broker includes the agreement with the broker to share security related information with the broker and in turn the broker has a non-disclosure agreement with the infrastructure provider.

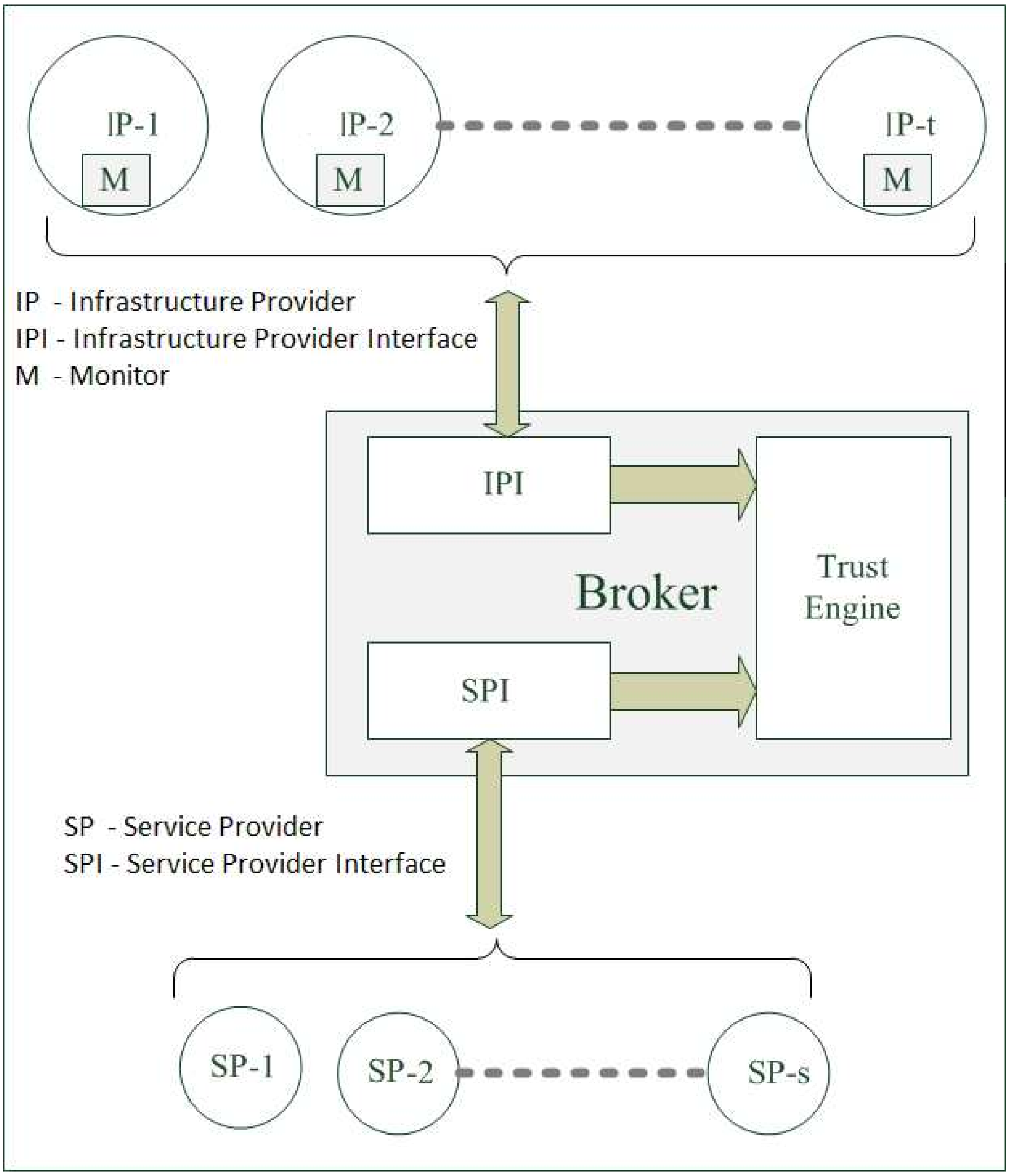

IP-1
M
IP-2
M
IP-t
M
IP - Infrastructure Provider
IPI - Infrastructure Provider Interface
M - Monitor
IPI
Broker
Trust Engine
SPI
SP - Service Provider
SPI - Service Provider Interface
SP-1
SP-2
SP-s

Figure 6.5 : Cloud Broker Architecture for Security Reputation

6.7.1 Infrastructure Provider Interface (IPI)

This interface enables the infrastructure provider to provide details of its security practices and security measures in place, allowing advertising its security strengths. In our experience, we find infrastructure providers try to provide the following security measures as a basic step towards securing their customers environment: (*i) Protecting individual virtual environment* (*ii) Filter traffic between each virtual instances iii) Hardening the hypervisor iv) Protecting the network infrastructure v) Protecting the data stored at each individual virtual instance vi) Policy enforcement for authentication and access management to individual virtual instances vii) Patch management.*

6.7.2 Service Provider Interface (SPI)

This interface allows the service providers to input their security requirements, select most appropriate cloud infrastructure provider for their security needs, provide feedback on the services and also log complaints. The requirements associated with a service and the security features expected, are encoded in the *service manifest* as discussed in (Ferrer et al., 2012). The feedback and the complaints form a vital piece of evidence to model the cloud infrastructure providers reputation based on its security strength.

6.7.3 Monitors

The broker receives security violation events of the infrastructure provider by registering to the pub-sub (Srivatsa and Liu, 2007) monitors in the provider's infrastructure. The threats that prevent organizations from adoption of the cloud infrastructure services and the areas for gathering metrics are identified as follows: (*i) Insecure Authentication or Authorization:* Interface allowing customers to manage cloud services in order to perform provisioning, management, orchestration, and monitoring their virtual instances (*ii) Insider Attack:* An insider from cloud infrastructure provider could have privileged access to confidential data or gain control over the cloud service with no or little risk of detection (*iii) Multitenant Attack:* Cloud environment is meant to allow multiple users share resources (CPU, network, memory, storage, etc.) and an improper isolation of the multi-tenant architecture may lead to have access to any other tenant's data (*iv) Data Leakage:* Customers data on the cloud could be compromised, deleted or modified (*v) Malware Propagation:* Any malware that infects a virtual instance could propagate over the shared host or to hypervisor, spreading rapidly, giving ability to eavesdrop on customer's transactions.

6.7.4 Trust Engine

The trust engine contained in the cloud broker is the core part of the architecture that performs the *trustworthiness* calculation for the cloud infrastructure providers. Figure 6.6 shows the internal work flow used for computing the reputation of cloud infrastructure provider based on the inputs received from the interfaces of the broker.

i. *Evidence:* The evidences provided to the opinion model are gathered from monitors, cloud infrastructure provider interface and service provider interface.

ii. *Opinion Model:* The evidences received from different monitors are used to form an opinion about a cloud infrastructure provider based on the opinion model proposed in Chapter 4 (P.S. Pawar et al., 2012). The opinion of a proposition *x*, represented as *w(x)* or w_x is defined in terms of *belief b(x)* or b_x, *disbelief d(x)* or d_x and *uncertainty u(x)* or u_x where $b(x)+d(x)+u(x)=1$. The opinion model in (P.S. Pawar et al., 2012) also described in Chapter 4 is given as follows:

$$W_x = (b_x, d_x, u_x, a_x) \tag{6.1}$$

$$b_x = c\, r / t \tag{6.2}$$

$$d_x = c\ s / t \qquad (6.3)$$

$$u_x = t / (r\ s + f^2 + 1) \qquad (6.4)$$

$$c = 1 - u_x \qquad (6.5)$$

where: *r* is amount of positive evidence; *s* is amount of negative evidence; *t* is total evidence given as *t=r+s*; *c* or *c(t) or c(r,s)* is certainty as a function of total evidence; and *f* is distance of focus to the centre of an ellipse formed by mapping the positive and negative evidence to major and minor semi-axes of an ellipse.

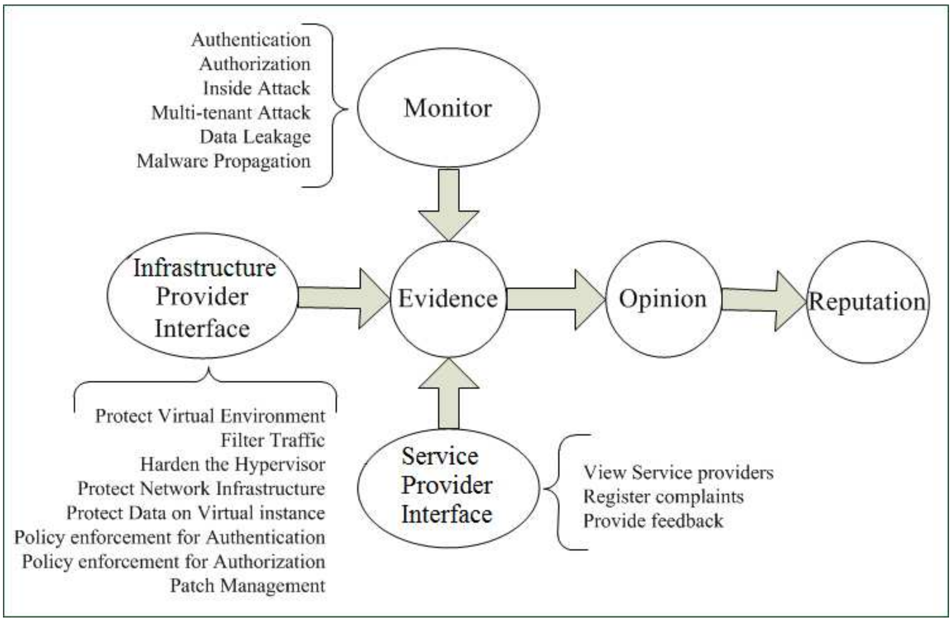


Figure 6.6 : Trust Engine

The opinion formed by the monitors is combined with the opinion formed based on the SPs feedback and complaints. The subjective logic in (Jøsang, 2001) is used to combine multiple opinions to form a single opinion using the operators such as conjunction, consensus that allows performing logical operations on opinions. This work uses the opinion model proposed in (P.S. Pawar et al., 2012) and the subjective logic operators (Jøsang, 2001). The conjunction operator is standard logic "AND" operating on the opinions. The consensus operator enables combining the opinions of entity A and entity B representing an imaginary entity [A, B]'s opinion about proposition *x*.

iii. *Reputation:* The probability expectation of an opinion is used to provide the reputation rating. The expectation of an opinion is given as *E(w(x))=b+au* where *E(w(x)) Є [0,1]* and *a(x)* is base rate that provides the weight of uncertainty that contributes to the probability expectation.

Figure 6.6 shows process of modelling the security reputation by broker. The first step is the broker getting evidential information from two sources a) Monitor and b) Service Provider interface. The second step is to convert the evidence obtained to compute an opinion. The third step is to calculate the reputation of an infrastructure provider based on the opinion formed. The details of reputation calculation are given in section 6.8.

6.8 Reputation System

The reputation of a cloud infrastructure provider is calculated in terms of its *trustworthiness (T)* using opinion obtained from computations, namely i) *Incidence Monitoring (M)*: Security incidence events received from monitoring ii) *Service Provider Rating (SPR)*: Ratings provided by the Service Provider for satisfaction of the security features provided by IP. The *trustworthiness(T)* is given by applying the conjunction operator of subjective logic on the opinions obtained from each of these computation and then calculating the expectation of the combined opinion.

$$T=\text{Expectation}\ (W_{M \wedge SPR}) \qquad (6.6)$$

Where W_M is the opinion obtained from the monitoring (M) as well as the W_{SPR} is the opinion obtained from the service provider rating (SPR). The symbol Λ is the *conjunction operator* used to combine the two opinions.

6.8.1 Incidence Monitoring

The incidence monitoring records evidence about the incidences related to parameters such as authentication, authorization, inside attacks, multi-tenent attack, data leakage and malware propogation. These incidents can either be identified by the cloud infrastructure provider and sent to the broker or the broker after receiving the security events carries further analysis to identify the incidences from the data received. Both approaches have their own advantages and disadvantages.

For each monitoring parameter, the number of incidents occuring within a time window *w* are observed. Every incident identified, adds to the negative evidence and absence of incidents increases the positive evidence. Based on the positive and negative evidences, opinions are formed for each of the parameters. Let $W_{AT,}$ $W_{AR,}$ $W_{IA,}$ $W_{MT,}$ $W_{DL,}$ and W_{MP} be opinions formed for IP based on the monitoring parameter of authentication, authorization, inside attacks, multi-tenent attack, data leakage and malware propogation respectively. Consider for example that there are *n* monitors associated with monitoring of authentication incidence at IP-1. Then the opinion W_{AT} for IP-1 is given as the *consensus* of all *n* monitors. Considering all monitoring parameters, the overall opinion W_M for IP-1 is given by applying *conjunction* operator over the *consensus* opinion, which is as follows:

$$W_M = W_{AT}^{M1,\ldots,Mn} \Lambda W_{AR}^{M1,\ldots,Mn} \Lambda W_{IA}^{M1,\ldots,Mn} \Lambda$$

$$W_{MT}^{M1,\ldots,Mn} \Lambda W_{DL}^{M1,\ldots,Mn} \Lambda W_{MP}^{M1,\ldots,Mn} \qquad (6.7)$$

Where $W_{AT}^{M1,.,Mn}$ is consensus opinion by monitors M1 to Mn regarding authentication. Similarly consensus opinions for other parameters are obtained.

6.8.2 Service Provider Rating

For every usage of the services from the IP, the service provider rates the satisfaction of security features and capabilities provided by the IP corresponding to the requirements set forward initially by the SP. Consider *q* SPs registered with the broker and provide ratings to the IP for each of the monitoring parameters. The overall opinion W_{SPR} for IP-1 based on the service provider rating is given by applying the *conjunction* operator over the consensus opinion, as follows:

$$W_{SPR} = W_{AT}^{SP1,SP2\ldots,SPq} \wedge W_{AR}^{EU1,EU2\ldots,EUq} \wedge W_{IA}^{SP1,SP2\ldots,SPq} \wedge W_{MT}^{SP1,SP2\ldots,SPq} \wedge W_{DL}^{SP1,SP2\ldots,SPq} \wedge W_{MP}^{SP1,SP2\ldots,SPq} \qquad (6.8)$$

Where $W_{AT}^{SP1,SP2\ldots,SPq}$ is consensus opinion for IP-1 given by service provider SP1 to SPq based on the authentication. Similarly $W_{AR}^{SP1,SP2\ldots,SPq}$, $W_{IA}^{SP1,SP2\ldots,SPq}$, $W_{MT}^{SP1,SP2\ldots,SPq}$, $W_{DL}^{SP1,SP2\ldots,SPq}$ and $W_{MP}^{SP1,SP2\ldots,SPq}$ are the consensus opinion for IP-1 by SP1 to SPq based on authorization, insider attacks, multi-tenent attacks, data leakage and malware propogation respectively.

6.8.3 Trust of Cloud Service Provider

The *trustworthiness (T)* of the cloud infrastructure provider is given by calculating the expectation of the opinions W_M and W_{SPR} given by incidence *monitoring* and the *Service Provider* respectively. The *trustworthiness (T)* can be represented as:

$$T = \text{Expectation } (W_M \wedge W_{SPR}) = \text{Expectation } (W_{M \wedge SPR}) \qquad (6.9)$$

Where $W_{M \wedge SPR} = (b_{M \wedge SPR}, d_{M \wedge SPR}, u_{M \wedge SPR}, a_{M \wedge SPR})$ and the expectation of the opinion $W_{M \wedge SPR}$ is given as :

$$E(W_{M \wedge SPR}) = b_{M \wedge SPR} + (a_{M \wedge SPR})(u_{M \wedge SPR}) \qquad (6.10)$$

6.9 Conclusion

The cloud broker is designed to simplify and optimize the life-cycle of a cloud service as its components operate, interact, and communicate across multiple cloud platforms. The CBR architecture negotiates and creates agreements with multiple providers, also assists consumers by providing optimal solution for multi-cloud deployment, relieve the consumers from the complexities of application to infrastructure mapping and handle the requirement of non-trivial networking among all resource providers.

The cloud broker architecture provides capabilities such as monitoring, SLA negotiation, service construction and deployment which are essential surrounding capabilities required by the trust framework to perform trust assessments of cloud service providers.

The CBR architecture with different modes of operations significantly reduces the SPs/consumers efforts to select and deploy its services in appropriate cloud infrastructures. The trust framework can utilize the modes of cloud broker to perform various kind of trust assessment. Chapter 3 outlines the various trust assessments performed using the different modes of cloud broker whereas Chapter 4 and Chapter 5 describe in detail the trust evaluation of cloud service provider using the cloud broker.

In this chapter we also proposed security reputation systems using broker architecture for cloud infrastructure providers, allowing service providers to achieve a level of expectation from cloud infrastructure providers about their deployed security systems. Trust and reputation systems, when combined with suitable cyber security assessment, governance, risk and compliance frameworks, can provide a means of reducing the potential risk of using cloud infrastructure services while increasing consumer confidence and offer additional incentives for cloud infrastructure providers to increase their level of compliance to cyber security. This chapter provides a high level architecture for reputation of cloud infrastructure providers in conjunction with cyber security. At present, the governance, cyber security and compliance frameworks for the cloud providers lack such a support.

Chapter 7 Evaluation

7.1 Introduction

This chapter presents the evaluation of the cloud broker architecture and trust models that have been introduced in the previous chapters. This chapter is organized as follows:

The first experiment is for the cloud broker architecture that is used to evaluate the trust model. Section 7.2 details on the performance due to the mediation layer of cloud broker.

Section 7.3 provides details of the experiments performed for the evaluation of the Opinion based trust model for optimized cloud services that is proposed in Chapter 4.

Section 7.4 describes the evaluation of the extended trust model based on the cloud characteristics and credibility. The evaluation of this trust model uses the cloud broker case study described in section 5.2 and in Chapter 5.

Section 7.6 provides the concluding remarks based on the evaluations of the trust models.

7.2 Percentage of Deployment Time Overhead

The main objective of this experiment is to assess the overall deployment time overhead due to the intermediary functionality of the cloud broker and analyse the overhead due to individual components of the cloud broker.

To compute the percentage overhead due to deployment via the cloud broker, our experimental setup comprised of multi-cloud deployment via cloud broker and a multi-cloud deployment by consumer without a cloud broker as shown in Figure 7.1 and Figure 7.2.

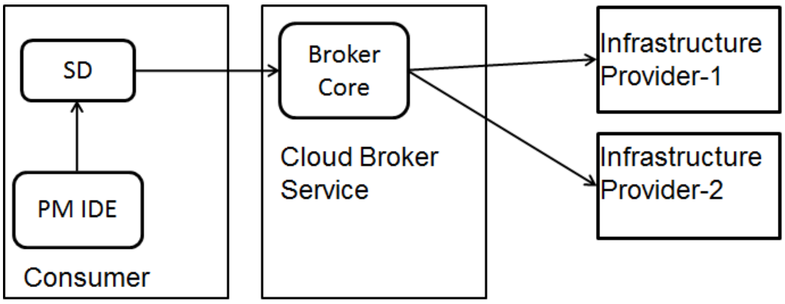


Figure 7.1 : Multi-cloud deployment via cloud broker

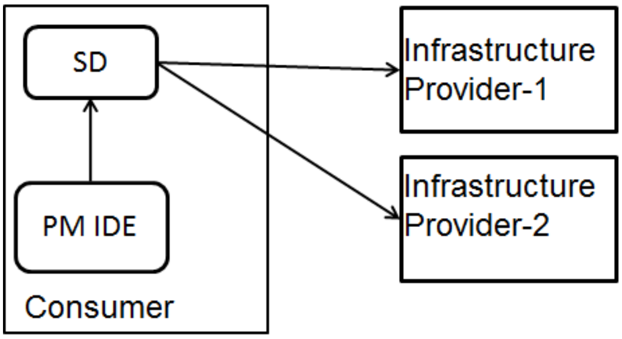


Figure 7.2 : Multi-cloud deployment without cloud broker

The multi-cloud deployment steps via CBR are shown in Table 7.1. The multi-cloud deployment by the consumer directly with cloud providers without involving CBR will have all the steps except for Step 1 & Step 8 that deal with SLA agreements with CBR and the Step 4 that deals with data initialization at the CBR.

	Deployment Steps	**Time (Broker multi-cloud)**	**Time (consumer multi-cloud)**	**Percentage Overhead due to deployment via CBR**
1.	Negotiation with CBR	16sec	-	4.8 %
2.	Perform legal check	6sec	6sec	0 %
3.	Get deployment solution	142sec	142sec	0 %
4.	Consumer to CBR data upload initialization	10sec	-	3.05 %
5.	CBR/Consumer to multiple providers data upload initialization	45sec	45sec	0 %
6.	Contextualize for provider settings and VAS of CBR	24sec	20 sec	1.22 %
7.	CBR/consumer upload contextualized ISO files	85sec	75sec	3.05 %
8.	Consumer creates Agreement with CBR	5 sec	-	1.52 %
9.	CBR/consumer create agreements with multiple Cloud providers	39sec	39sec	0 %
	Total time	**372 sec**	**327 sec**	
	Total overhead			**13.76 %**

Table 7.1 : Percentage overhead due to deployment via cloud broker used as arbitrage

This experiment uses the cloud broker implementation as described in Chapter 6 considers real deployment of the Genomic application described in section 6.2.1 which is in the form of a VM image of size 1 GB. The deployment of application components is performed over the OPTIMIS testbed("Optimis - Optimized Infrastructure Services," n.d.). Each component in the CBR is configured to log the start and end time of it's activity. The time required for each of the steps during the deployment of the application via the cloud broker is computed using the log. For the purpose of multi-cloud deployment by the consumer without the CBR, the step 2, step 3 and step 5 provides the same results considering the system with similar configurations for the consumer and the CBR. Step 6 that performs contextualization requires less time compared to the contextualization at the CBR. This is mainly due to the three VAS (Value Added Services) that the CBR included into the ISOs created by the contextualization component. Step 7 has a relation with step 6, wherein the ISOs created during contextualization at the CBR are slightly larger (due to inclusion of VAS) compared to the consumer contextualization, which impacts the upload time of ISOs to the cloud provider via the CBR.

Table 7.1 shows the amount of time taken for the completion of each step in the deployment process, for the deployment via the CBR and without CBR. We computed the overhead as the ratio of "additional time taken by the CBR for each step" to the "total deployment time of the consumer without CBR".

The experiment is executed 20 times and the Table 7.1 shows the average time recorded for each step. The observation of this experiment shows that the overall overhead due to the deployment via broker is 13.76% of which the overheads due step 1 (4.8% overhead due to negotiation), step 4 (3.05% overhead due to CBR upload initialization) and step 8 (1.52% overhead due to agreement creation with

CBR) is unavoidable due to CBR used as arbitrage. The overheads due to step 6 (1.22% due to contextualization of VAS) and step 7 (3.05% due to ISOs upload) are adjustable with the VAS requirements of the consumer.

The observation of this experiment shows that, the total overhead of 13.76% may account to few seconds of delay in the deployment of the service via the cloud broker. Considering the advantages and easiness that cloud broker provides the SP/Users for multi-cloud deployment, a few seconds of delay in the deployment process may be acceptable.

7.3 Evaluation of Opinion Based Trust Model for Cloud Services

In order to evaluate the proposed trust model in Chapter 4, we have developed a prototype tool. We used this tool to evaluate the model in three different experiments. More specifically, in the first set of experiments we provide a comparison of the proposed opinion model with other existing models using data set from Amazon marketplace (www.amazon.co.uk). In the second and third sets of experiments, we use the example of the cloud computing scenario described in Section 4.2 to evaluate the use of the various parameters considered in our model. In the second set of experiments we analyse the proposed model for each individual parameter, namely (a) SLA monitoring, (b) SP ratings, and (c) SP behaviour. In the third set of experiments, we analyse the model when considering combinations of the parameters in order to see if the use of more than one parameter provides better trust values.

7.3.1 Comparison of the Proposed Model

The objective of this experiment is to evaluate the performance of the trust model in terms of its accuracy, to assess the future behaviour and compare this performance with some of the existing trust models. To evaluate the accuracy of the trust models, we have used a real data set from the Amazon market place.

The dataset of Amazon marketplace used in this evaluation includes rating received by users, for four sellers and for the same music track CD. The seller1, seller2, seller3 and seller4 are rated by 618, 154, 422, and 314 distinct users respectively and these users rated the sellers at different times independent of each other. This data set contains ratings in the range of 1 to 5, for each seller, provided by the users. Table 7.2 provides the sample data for first 10 users amongst the 618 users who rated seller1.

ID	Author ID	Ratings	Review
1	1	5	"great seller thank you"
2	2	4	"happy with delivery "
3	3	5	"Wonderful transaction. Book in great condition, well packaged and received very quickly. A super seller! Thank you very much."
4	4	5	"Just as expected"
5	5	1	"Received damaged item. seller refused to refund sending a number of rude, angry E-mails accusing me of lying. Only when i submitted a claim and sent E-mail of complaint to Amazon did seller eventual agree to refund. Sent damaged item back and, unbelievably, seller further accused me of watching and damaging the whole box set! Seller was rude, arrogant and unco-operative throughout. STAY AWAY!!!"
6	6	5	"thank you"
7	7	2	"Order cancelled by seller and refund given. However, this was advised by Amazon. No communication from the seller and this is reason for 2 stars. I think that a quick note from the seller with an apology would have been appropriate in the circumstances. Hopefully I can still obtain elsewhere."
8	8	5	"Very happy "
9	9	5	"none"
10	10	5	"delighted with the cd, arrived promptly and was in excellent condition, thankyou very much"

Table 7.2 : **Sample dataset of 10 user ratings for seller1, on Amazon market place**

The rating is converted to the form <r:positive, s:negative> evidence such that r+s=1. More specifically, rating 1 maps to <0,1>, rating 2 maps to <0.25,0.75>, rating 3 maps to <0.5,0.5>, rating 4 maps to <0.75, 0.25>, and rating 5 maps to <1,0>. A user performing the (i+1)th transaction has access to all the previous *i* ratings.

We compared the proposed model with Jøsang's (Jøsang, 2001) and Wang's (Wang and Singh, 2010) approaches. For all the three models, the experiment takes previous *i* ratings to predict the (i+1)th rating and calculates the expectation $E=b+au$ to predict the (i+1)th rating. The belief is calculated using the *i* previous ratings and the base rate is considered as 0.5.

Figure 7.3 shows the experimental results for a single seller. One time step on the x-axis represent 25 transactions and the y-axis represents errors that are computed as the average of 25 prediction errors based on the ratings. The results of the experiment shows that our model has lower prediction error when compared to Jøsang's (Jøsang, 2001) and Wang's (Wang and Singh, 2010) approaches. Table 7.3 summarizes the experiment performed for four sellers for the same music track CD.

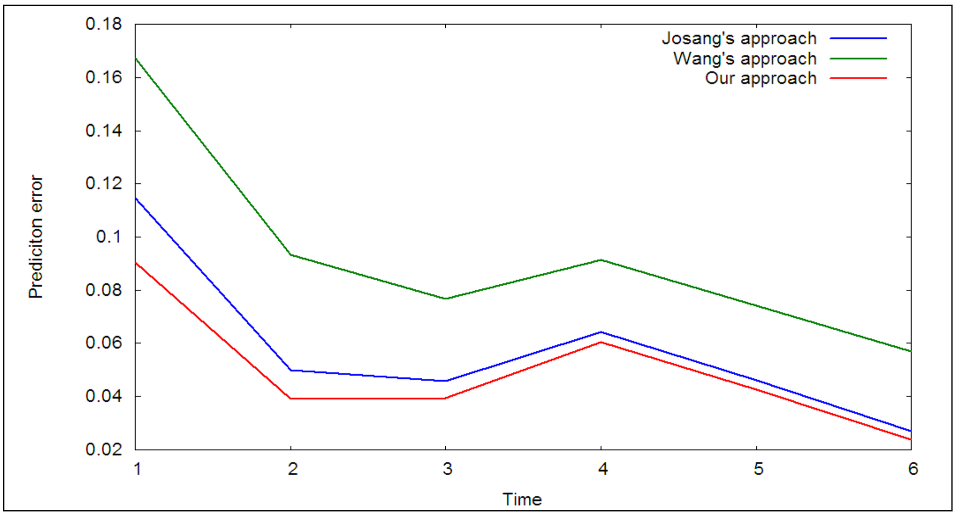


Figure 7.3 : **Average prediction error for a Seller based on the ratings [1,5] (x-axis: One time stamp represent 25 transaction; y-axis: Average of 25 prediction errors)**

Approach	Seller1	Seller2	Seller3	Seller4
Jøsang ’—	0.10619	0.05736	0.06219	0.10809
Wang’s	0.12753	0.09278	0.09415	0.14004
Our	0.10456	0.04878	0.05848	0.10449

Table 7.3 : **Average prediction error for 4 sellers based on the ratings [1,5]**

The observation and results of the experiment validates the improved performance of the proposed trust model compared to other well-known trust models, for predicting the future behaviour based on the evidence available. Predicting future behaviour with high accuracy is certainly an essential requirement for a trust model and the results of this experiment tend to show a positive support towards selection of our trust model.

7.3.2 Experiments Using Individual Parameters

7.3.2.1 SLA Monitoring

The aim of this experiment is to assess the impact of the SLA Monitoring parameter when only this parameter is available in the trust model. The impact of the SLA parameter is assessed in a simulated environment by varying this parameter from being compliant to non-compliant and observing the behaviour of the model.

In this experiment, we consider only the SLA monitoring parameters with four resources (CPU, memory, disk, VM) associated with IP1 as fixed. We considered that the resource demand requests are sent by all SPs with incremental resource requirements. When IP1 is able to provide the demanded resources, IP1 is considered compliant with the SLA and this increases the positive evidence maintained by the SPs for IP1. At a certain point the requested resources exceed the capacity of the IP1 resulting in SLA violations. The SLA violations, add to the negative evidence maintained by the SPs for IP1.

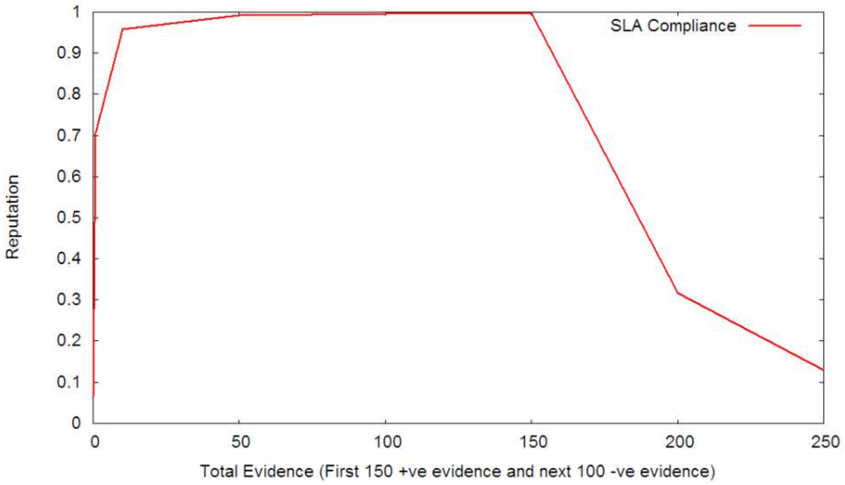


Figure 7.4 : **Reputation based on SLA monitoring only**

The result of this experiment in Figure 7.4 shows that the reputation increases when each of the SPs have positive evidence; a maximum reputation is achieved by IP1 when each of the SPs had positive evidence of 150. After this point, the SLA violations accumulate negative evidences causing a reduction on the reputation. The observation of this experiment validates the trust model with only SLA compliance parameter which shows any non-compliance results in reduction of the reputation score while being compliant increases the reputation score.

7.3.2.2 SP Rating

The aim of this experiment is to assess the impact of the SP Ratings when only this parameter is used in the trust model. The impact of the SP ratings is assessed in a simulated environment by varying the SP rating parameter with the mixture of positive and negative ratings and observing the behaviour of the trust model.

In this experiment we considered that all the SPs used IP1 and rated IP1 for its performance based on CPU, memory, disk and virtual machine indicators. These ratings are preserved by the SPs for evaluating the IPs. The experiment starts with IP1 receiving positive ratings from each of the SPs. Each time the ratings are provided to IP1, SP1 calculates the reputation of IP1 taking into account its own ratings as well as the ratings of the other SP2 to SP5 providers. When a degraded performance is observed (i.e.; there are SLA violations), the SPs rate IP1 with negative ratings. In this experiment, the SP1's positive and negative evidence is fixed as 200 positive and 50 negative evidences.

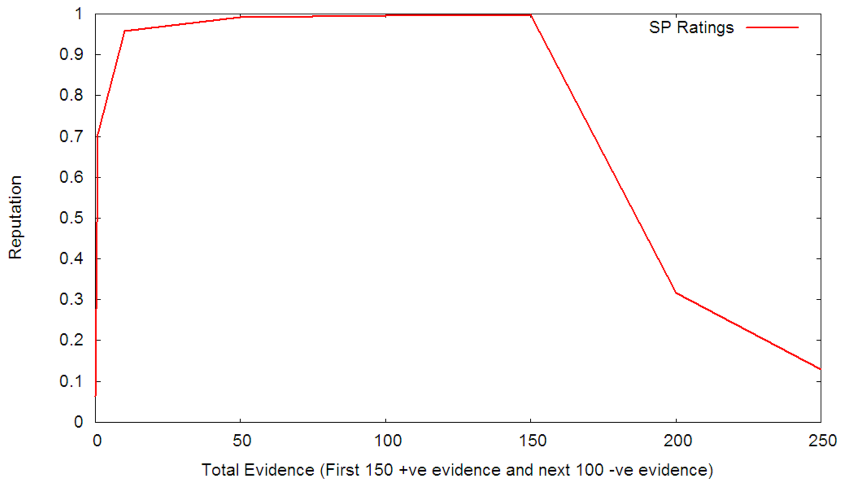


Figure 7.5 : **Reputation based on SP Ratings only**

The result of this experiment is as shown in Figure 7.5. From the results it can be observed that the increase in the positive ratings received by SP1 from other SPs, increase the reputation until the positive evidence reaches 150. As SP1 starts receiving negative ratings from other SPs, the reputation reduces. The observation of this experiment shows that SP rating parameter has sufficient impact on the reputation trust where in the positive rating increases the reputation score while negative rating decreases the reputation score.

7.3.2.3 SP Behavior

The aim of this experiment is to assess the impact of the SP behaviour when only the SP behaviour parameter is used in the trust model. The impact of the SP behaviour is assessed in a simulated environment by varying the SP behaviour parameter with the mixture of positive and negative ratings and observing the behaviour of the trust model.

In this case, the experiment begins with all SPs using only IP1 for all its resources (CPU, memory, disk space, and virtual machine). The positive behaviour of all SPs increases the positive evidence for all SPs, which increases the reputation of IP1 in terms of SPs behaving towards IP1. A degraded performance observed from IP1 may lead to SPs changing their infrastructure provider. This reduces the SPs positive behaviour towards IP1 and increases the negative evidence for all SPs, reducing the reputation of IP1.

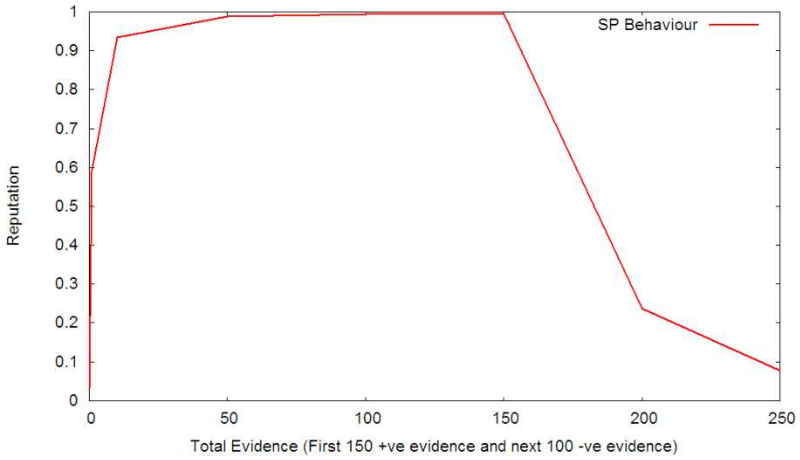


Figure 7.6 : **Reputation based on SP Behaviour only**

Figure 7.6 shows the results of this experiment. From the results it can be observed that the increase in the positive evidence, increase the reputation until the positive evidence reaches 150. As SPs stop using IP1 due to bad service, IP starts receiving negative evidence from SPs which reduces the reputation score. The observation of this experiment shows that SP behaviour parameter has sufficient impact on the reputation trust where in the positive evidence increases the reputation score while negative evidence decreases the reputation score.

In summary, the experiments with individual parameters considered show an increase in the reputation with SLA compliance evidence for SLA monitoring, and positive SP ratings and positive SP behaviour towards an IP. Also violations of SLA, negative SP rating values, and negative behaviour of an SP reduces the reputation of an IP.

7.3.3 Experiments Using Combination of Parameters

7.3.3.1 Combination of SP rating and SP Behavior

The aim of this experiment is to assess the impact of using combined parameters of SP rating and SP behavior in the trust model. The impact is assessed in a simulated environment by having a static value to SP rating and varying the SP behaviour parameter from no positive evidence to a high positive evidence. A similar behaviour can also be obtained by varying the SP behavior parameter from no negative to a very high negative evidence.

In this experiment, we consider IP1 with positive ratings from all the SPs. SP1 calculates the reputation of IP1 considering its own ratings as well as ratings of SP2, SP3, SP4 and SP5. The ratings provided by SP2, SP3, SP4 and SP5 are first discounted using SP2's, SP3's, SP4's and SP5's behavior respectively towards IP1. When maintaining constant SP ratings by all SPs, the SP behavior of SP2, SP3, SP4 and SP5 changes by increasing the positive behavior of these SPs for initially zero positive behavior to a very high value.

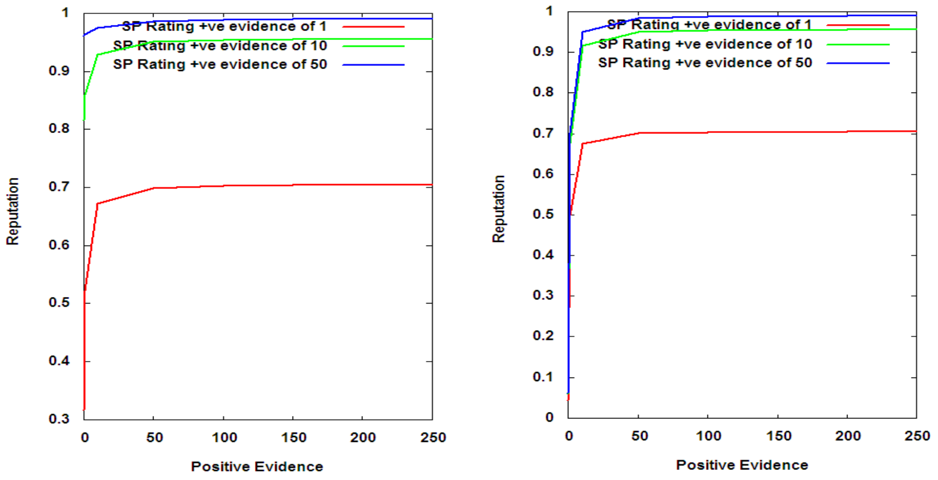
Figure 7.7(a) shows that (i) as the SP behavior becomes more positive, the reputation of IP1 increases; (ii) when SP1 has less evidence, there is a large variation, which causes a bigger impact of the other SP behavior and as the SP1's amount of evidence increases, the reputation has less impact of SP behavior. Intuitively this means that as the direct experience of SP towards the IP increases, the SP have more direct evidence and the influence of the other SP behavior on the trust computation reduces.

7.3.3.2 Combination of SP rating and SLA monitoring

The aim of this experiment is to assess the impact of using combined parameters of SP rating and SLA monitoring in the trust model. The impact is assessed in a simulated environment by having a static value to SP rating and varying the SLA parameter from no positive evidence (non-compliant) to high positive evidence (high compliance). A similar behaviour can also be obtained by varying the SLA parameter from no negative (compliant) to a very high negative (non-compliant) evidence.

In this experiment, to calculate the opinion of IP1 based on SP ratings, we consider all past provided SP ratings. We maintained constant opinions about IP1 and considered that the positive evidence of SLA compliance is varied from zero to a high amount of positive evidence for all SPs (SP1 to SP5).

From Figure 7.7 (b) it is observed that when the positive evidence from the SLA monitoring increases, the reputation of IP1 also increases. The observation of this experiment reveals that as the direct experience of SP towards the IP increases, the SP have more direct evidence and the influence of the SLA monitoring on the trust computation reduces.



(a)

(b)

Figure 7.7 : Reputation based on (a) SP ratings and SP behaviour, (b) SP ratings and SLA monitoring

7.3.3.3 Combination of SP Rating, Behavior and SLA monitoring

In these experiments we calculated the reputation using all parameters. We considered the values of two of the parameters fixed and varied the third parameter, as explained below.

Effect of SP behavior

The aim of this experiment is to assess the influence of the SP behaviour on the trust model in the presence of other parameters. The influence of the SP behaviour is assessed in a simulated environment by varying this parameter only and keeping static all the other parameters of the trust model.

The SP rating is fixed at a total of 10 positive evidences by each of the SPs. The SLA monitoring is fixed at 50 positive evidences as total evidence by each SP towards IP1. The SP behavior for SP1 to SP5 is varied from zero positive to a positive evidence of 250 in a total evidence of 250.

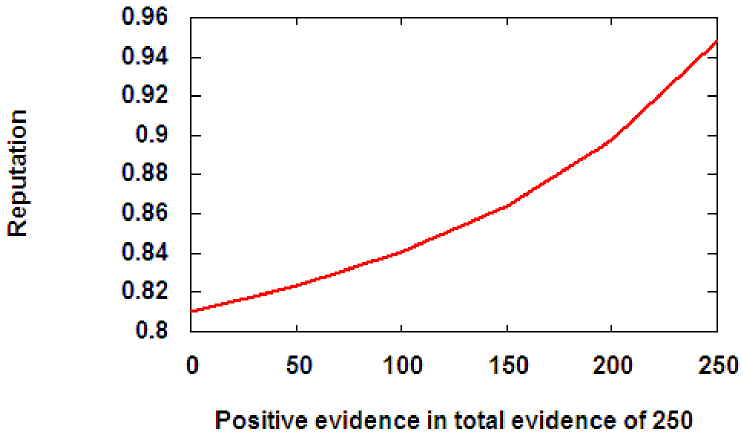


Figure 7.8 : Effect of SP behaviour

Figure 7.8 shows that with the increase in the positive evidence of SP behaviour the reputation of IP1 increases. The result of this experiment shows that changes in the evidence of SP behaviour has significant impact on the reputation of the IP even in the trust model with other parameters.

Effect of SLA monitoring

The aim of this experiment is to assess the influence of the SLA Monitoring parameter on the trust model. The influence of the SLA parameter is assessed in a simulated environment by varying this parameter only and keeping static all the other parameters of the trust model.

The SP ratings provided by all SPs for IP1 and the SP behavior for all SPs are fixed. The total evidence consists of only positive evidence obtained from SLA monitoring, which is varied from zero to 250. Figure 7.9 (a) shows that the reputation of IP1 increases with the increase in positive evidence obtained.

The effect of SLA monitoring information is important to evaluate reputation of an IP during the operational phase. In a cloud environment, when the SPs deploy their services on a particular IP, the services are retained for significantly longer duration. This results in less frequent updates of SP ratings and SP behaviour. The provision of updates of compliance/non-compliance SLA monitoring information at regular intervals may have significant impact on the reputation of an IP, as shown in Figure 7.9 (a). The observation of this experiment validates and shows that the SLA compliance has high impact on the reputation trust regardless of other parameters available in the trust model. Any non-compliance results in reduction of the reputation score while being compliant increases the reputation score in the trust model.

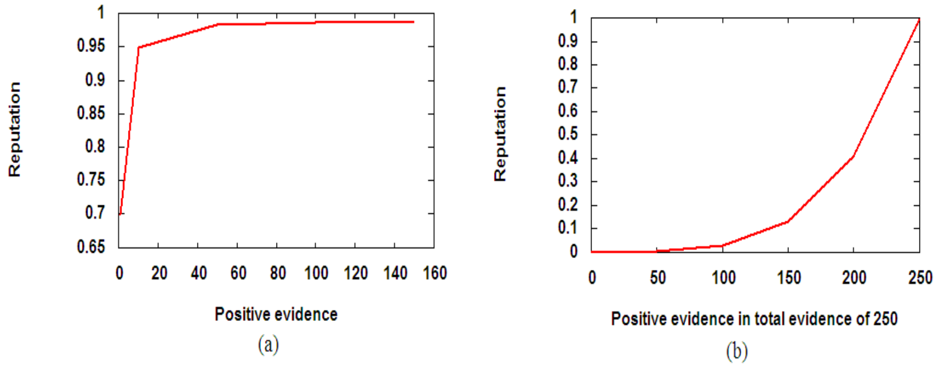


Figure 7.9 : **(a) Effect of SLA compliance; (b) Effect of SP rating**

Effect of SP ratings

The aim of this experiment is to assess the influence of the SP behaviour parameter on the trust model. The influence of the SP behaviour parameter is assessed in a simulated environment by varying this parameter only and keeping static all the other parameters of the trust model.

The SP behavior of all SPs towards an IP and the SLA violation for an IP provided by all SPs are fixed. The positive evidence from all SPs for IP1 is varied from zero to 250 in a total evidence of 250. Figure 7.9 (b) shows that as the positive evidence increases and the negative evidence reduces the reputation of IP1 will increase. The observation of this experiment validates and shows that the SP rating parameter has equally high impact on the reputation trust as the other parameters available in the trust model.

7.4 Evaluation of Trust Model for cloud Based on Cloud Characteristics

The Trust model is evaluated using a simulation of the cloud computing scenario discussed in Section 5.2. The detailed architecture of the cloud broker is also described in Chapter 6. A typical simulation is carried out for 250 iterations, with a total of 100 SP nodes, one CBR node trying to evaluate a single IP node. The SP nodes are tagged with one of the four categories which include: normal group (G1), exaggerated positive group (G2), exaggerated negative group (G3) and complementary group (G4). The experiments use the different ratios G1:G2:G3:G4 of the SP nodes. The remaining sections are organized as follows: Section 7.4.1 describes metrics to study the characteristics of the Trust model. Section 7.4.2 demonstrates the trust model robustness due to consideration of credibility in the trust model. Section 7.4.3 demonstrates sensitivity of the model to uncertainty. Section 7.4.4 and Section 7.4.5 mainly check the robustness of the model against malicious ratings. Section 7.4.6 shows the effect on trust due to positive and negative evidences. Section 7.4.7 demonstrates the enhancement to the trust model over the credibility due to the introduction of malicious filter.

7.4.1 Metrics

To evaluate the trust model defined in Chapter 5 we defined two metrics similar to the one defined in (Jia et al., 2012):

7.4.1.1 Average Credibility

Average credibility indicates the competence that the trustor receiving feedbacks excludes or reduces influence of the malicious nodes. It is defined as

$$W = \sum_{i=1}^{M} \frac{W_{ik}}{M} \qquad (7.1)$$

Where *M* denotes the number of feedback provider who regarded node *k* as the witness. W_{ik} is the opinion provided by node *k* to node *i*. For each of the feedback provided, the node *i* maintains credibility against the opinion W_{ik} provided by node k. For the evaluation, we assume that the ratio of nodes is known for the normal group, positive exaggerated group, negative exaggerated group and complementary group. A high average credibility value means that node k is an acceptable feedback provider and malicious peers should achieve low credibility value with time.

7.4.1.2 Difference between real QoS and Feedback

The trustor observes the real quality of service provide by the trustee after the transaction is completed and compute the difference between the real QoS and the feedback provided. The *diff* is given as follows:

$$diff \ = \sum_{j=1}^{n} \frac{|T_j - t_j|}{n} \qquad (7.2)$$

Where T_j is trustee's reputation for node j and t_j denotes real QoS of node j's trustee. A high *diff* indicates that there exists a large gap between the node reputation and its QoS. A well designed system will result in a lower *diff* value.

7.4.2 Average Credibility Decreases with Time

The aim of this experiment is to assess the credibility parameter used in the trust model. The credibility parameter is expected to ensure that the feedback provided by malicious nodes should be weighted less to reduce the influence of malicious nodes and correctly model the reputation of the trustee.

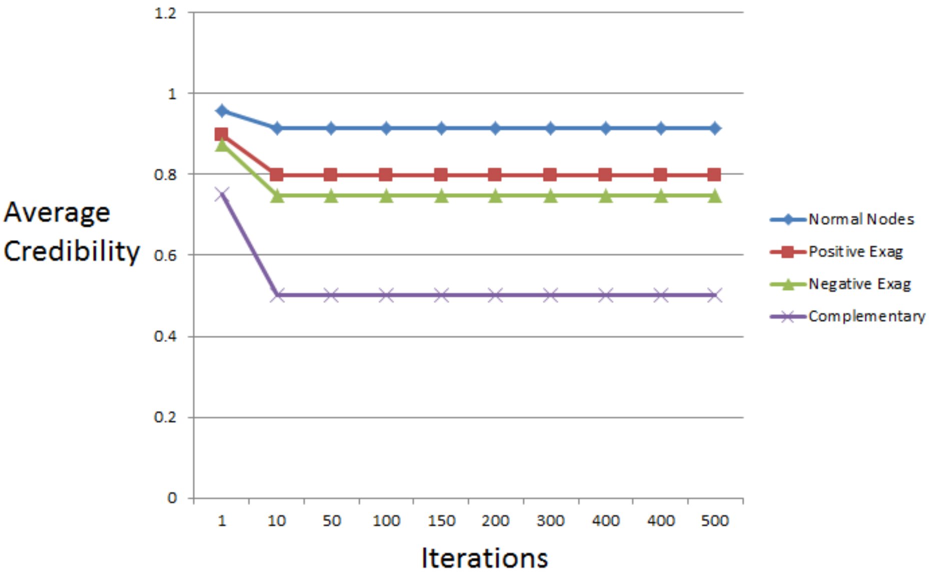


Figure 7.10 : **Average Credibility for different groups of SPs. G1:G2:G3:G4 is 70:10:10:10**

In this experiment, the ratio of nodes G1:G2:G3:G4 is given as 70:10:10:10. The CBR (cloud broker node) receives feedback from all the SPs about the IP (infrastructure provider node) and based on which it computes the reputation of IP. After the CBR performing transaction with the IP, it computes difference between the feedback provided and the real QoS provided by the IP. This enables the CBR to compute the current credibility of the feedback providers i.e. SPs. For iteration, credibility of the SPs is updated considering its previous credibility and the average credibility is computed for each group G1, G2, G3 and G4.

The results in Figure 7.10 shows that the average credibility for the malicious node groups G2, G3 and G4 decreases drastically within very few iterations and remains low throughout all the iterations. This result indicates that malicious nodes achieve low credibility with time and the feedbacks provided by these malicious nodes will have low influence on the reputation computation as the feedbacks provided by these malicious nodes will be weighted less.

7.4.2.1 Statistical analysis of the experiment results

Statistical analysis is carried out for the experiment to verify the results of the credibility model. Our goal is to show that the credibility model makes a difference. This goal can be achieved by showing that the credibility model helps differentiate the group of normal node from the exaggerated positive, exaggerated negative and complementary group of nodes.

The hypothesis for achieving this goal is: The average credibility of Normal group (G1) of nodes is significantly different than the average credibility of the exaggerated positive group (G2), exaggerated negative group (G3) and complementary group (G4). The null Hypothesis is: the average credibility of Normal group (G1) of nodes is same as the average credibility of the exaggerated positive group (G2), exaggerated negative group (G3) and complementary group (G4).

The result of the statistical t-tests are performed between the normal and positive exaggerated groups, normal and negative exaggerated groups, and normal and complimentary groups at a significance level of 0.01, is given in the table.

Groups	T- value	P- value	Result is significant at
Normal group and exaggerated positive group	9.98149	0.00001	$p < 0.01$
Normal group and exaggerated negative group	11.78306	0.00001	$p < 0.01$
Normal group (G1) and complementary group (G4)	15.47461	0.00001	$p < 0.01$

Table 7.4 : **Statistical analysis of the experiment result obtained for credibility model at degree of exaggeration α = 0.1 and a standard deviation of ±1σ**

The results in Table 7.4 signifies that there is a less than 0.001 % chance that the two sets of values come from the same group and the average credibility of Normal group (G1) of nodes is same as the average credibility of the exaggerated positive group (G2) and exaggerated negative group (G3). The result leads to acceptance of the hypothesis.

The t-test results presented in Table 7.4 are for the degree of exaggeration α = 0.1 and a standard deviation of ±1σ. Additional t-tests are performed for the degree of exaggeration with α = 0.1 and standard deviation of ±2σ and it is observed that the t-test results is significant at $p<0.01$.

However, t-tests results with degree of exaggeration with α = 0.05 and standard deviation of ±2σ are significant at $p<0.05$ but are not significant at $p<0.01$ for the positive and negative exaggerated groups, as shown in Table 7.5

Groups	T-value	P-value	Result
Normal group (G1) and exaggerated positive group (G2),	2.020435	0.044685	is significant at $p < 0.05$ is not significant at $p < 0.01$
Normal group (G1) and exaggerated negative group (G3)	2.452037	0.015071	is significant at $p < 0.05$ is not significant at $p < 0.01$
Normal group (G1) and complementary group (G4)	15.590359	0.00001	is significant at $p < 0.05$ and $p<0.01$

Table 7.5 : **Statistical analysis of the experiment result obtained for credibility model at degree of exaggeration $\alpha = 0.05$ and a standard deviation of ±2σ**

This signifies that as the degree of exaggeration becomes less, the chances of differentiating between the normal groups and the positive and negative exaggerated groups becomes difficult and may lead to errors. The t-test results in Table 7.5 signify that there are 4.4% and 1.5% chances that the positive and negative exaggerated groups respectively are considered same as normal groups, which directly impacts the average credibility of these groups.

7.4.3 Sensitivity to Uncertainty

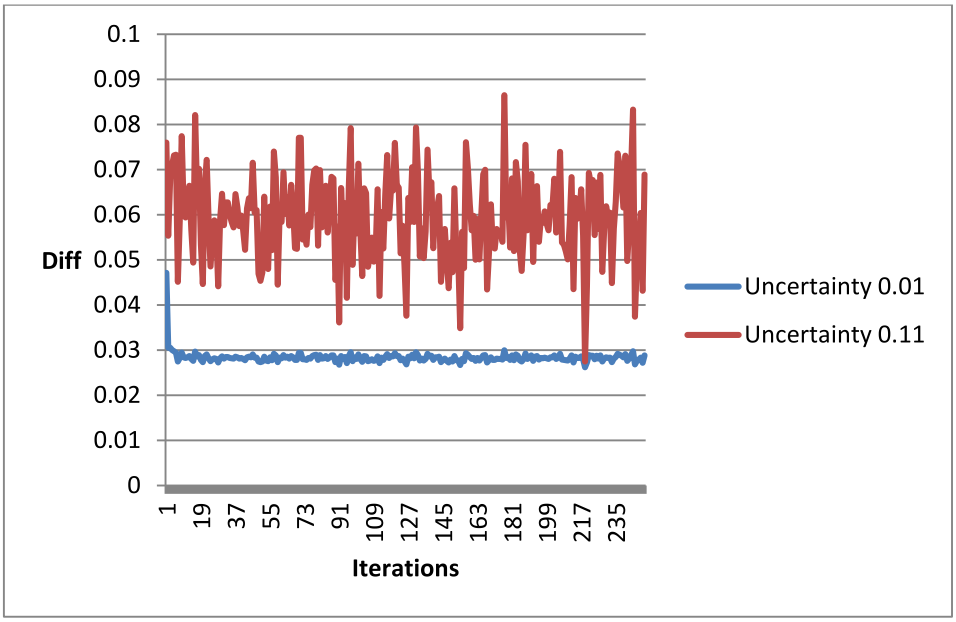


Figure 7.11 : **Diff for different levels of uncertainty by the feedback providers**

It is important to consider the feedback providers confidence in their feedback provided about the trustee. The aim of this experiment is to check if the confidences of the feedback provider have any impact on the robustness of the model. The feedback providers in the trust framework of section 5.3 provides feedback in the form of opinion *W = (b, d, u, a)*, about the trustee, which contains *belief*, *disbelief* and *uncertainty*. The feedback opinion is mainly the reliability trust R(i,j), for an entity *j* from the perspective of trustor *i*, which is given as the expectation of the opinion.

For this experiment, keeping the reliability trust provided by the feedback provider constant the experiment is executed for two cases of uncertainty for the feedback provided. In the first case a high uncertainty u=0.11 is maintained, while for the second case the uncertainty is reduced to 0.01. In both cases the malicious nodes ratio of 70:30:0:0 is considered for the experiment.

It is observed from Figure 7.11 that the trust model is sensitive to the uncertainty in the feedback value provided. The smaller the uncertainty, the *diff* value is correspondingly small. This result validates that with the increase in the evidence available, the uncertainty in the feedback value reduces and the system robustness increases.

7.4.4 Effect of Filtering With Mixed Category of Malicious Nodes

It is evident from the result obtained in Section 7.4.3 that due to the credibility parameter in the trust model, the *diff* value decreases with time. This signifies that the trust model is robust against: different ratios of malicious groups and different levels of uncertainty in the feedback provided.

As the *diff* value, which is the measure of difference between true QoS and the reputation computed, it is essential that this value reaches to its minimal as early as possible within the system. The result in Section 7.4.3 shows that the *diff* value reaches to a very low value after several iterations. In this experiment we aim to observe the impact on the trust model due to early filtering of the malicious nodes. For the purpose of this experiment the filtering technique described in (Zhang and Feng, 2009) is used for early filtering of malicious nodes. However, any other filtering techniques can also be adopted.

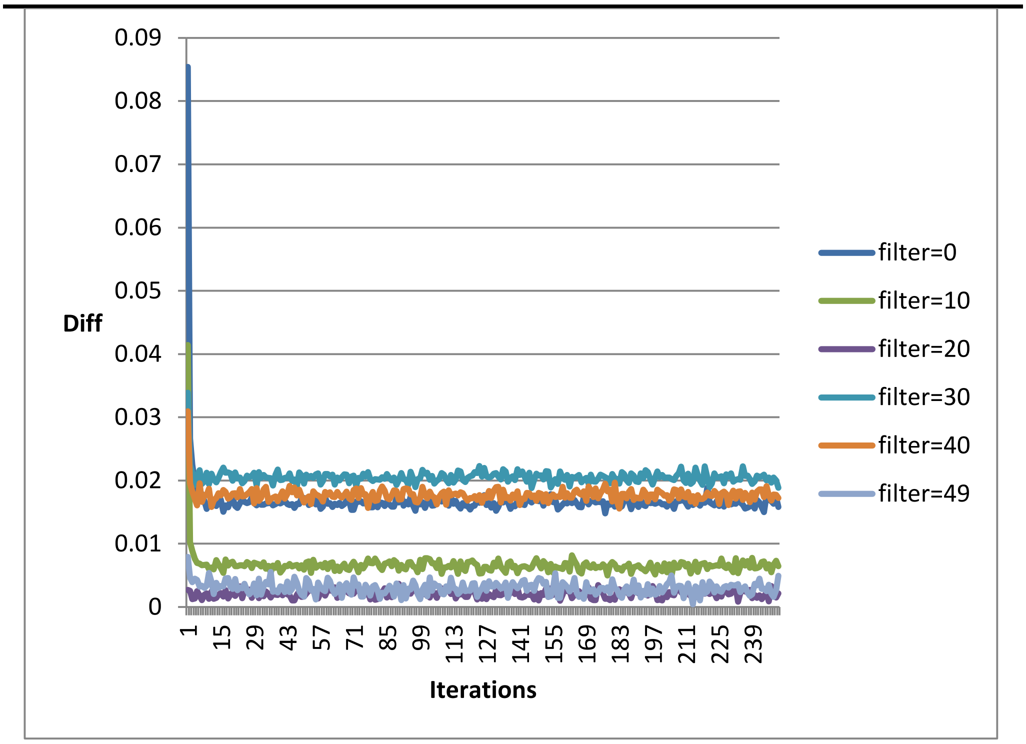


Figure 7.12 : Diff for different levels of filtering. SP node group ratio is 51:16:16:17

This experiment is performed with SP nodes having malicious node ratio considered as 51:16:16:17. The filtering of malicious nodes only restrict the use of ratings provided by these nodes for computation of reputation, but after the transaction is performed, the credibility of all the feedback providers is updated including the filtered nodes. This helps to appropriately discount the feedback of the nodes when it appears in the normal group category.

The results of this experiment shows that with no filtering or filter=0, the *diff* value is relatively higher when compared to filter = 49. Filter = 10 here signifies that 10% of the total nodes are filtered by the filtering mechanism to filter the unfair or malicious ratings. It is observed from the result that as the filter value increases the *diff* value reaches to a minimum, early in the iterations, which shows that the system becomes robust early on in time.

The result in Figure 7.12 shows that, as the filter level increases the *diff* value reduces, but a typical behaviour is obtained when the filter = 20. With further analysis it was observed that when filter = 20 is applied, it filtered all the complementary malicious node leaving with only positive and negative malicious group node along with the normal nodes. Due to the remaining positive as well as negative malicious group nodes, the effect of malicious ratings is neutralized to obtain a low *diff* value.

7.4.5 Effect of n/2 Filtering Even if There are Lesser Malicious Nodes

The experiment in Section 7.4.5 is executed with assumption that the system does not contain more than n/2 malicious nodes and the filtering mechanism is also applied until n/2 nodes are filtered. 'n' here is the total number of nodes in the system. The aim of this experiment is to verify the impact of filtering n/2 nodes even if the system may have lesser number of malicious nodes. For this experiment the SP node group ratio is considered as 70:10:10:10 where only 30% of the nodes are only malicious, but the filtering is applied until filtering of n/2 nodes is achieved.

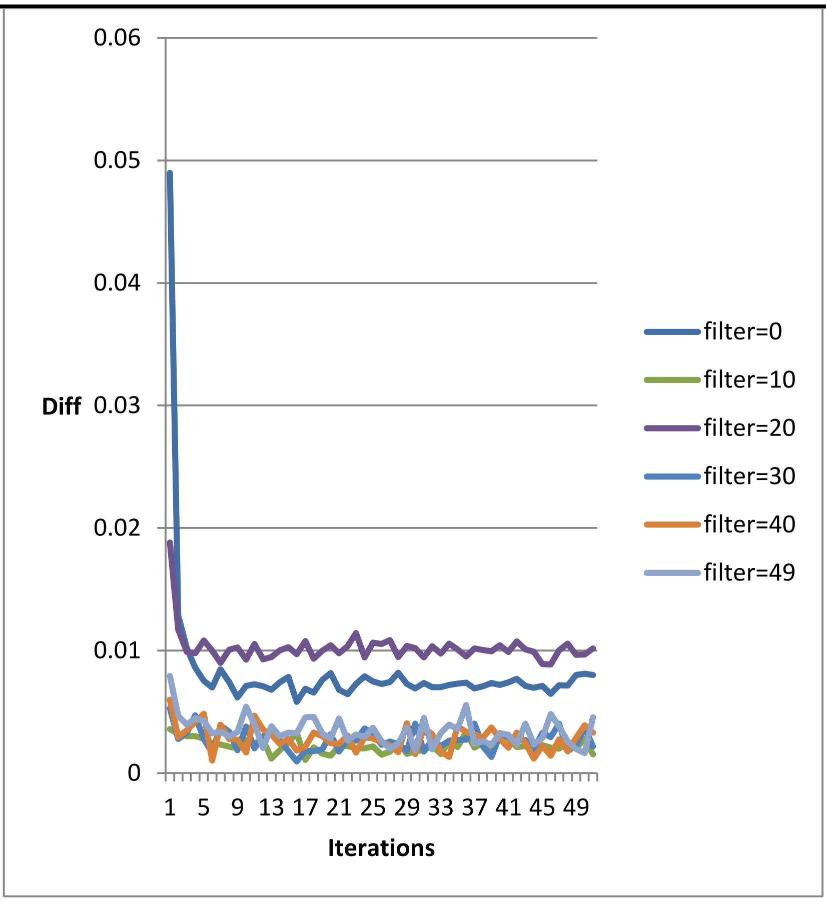


Figure 7.13 : Diff for different levels of filtering. SP node group ratio is 70:10:10:10

It is observed from the results in Figure 7.13 that there is no significant difference in the *diff* value when all 30% of the malicious nodes are filtered and when 49% of the nodes are filtered. In both cases the reputation computation is performed using the feedbacks from the normal group nodes.

This experimental result validates the applicability of the early filter to provide lower *diff*, in clean systems with no malicious nodes as well as in system where maximum of n/2 malicious nodes can exist.

7.4.6 Effect on Trust for Single and Multiple Context

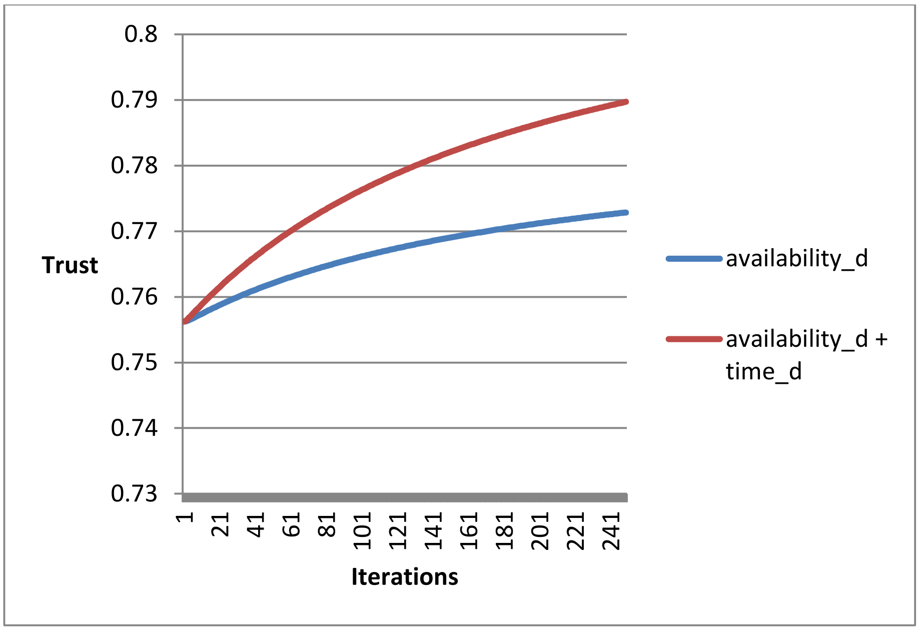


Figure 7.14 : Trust increases with increase in positive evidence and the rate of increase depends on number of contexts considered.

The aim of this experiment is to examine the behaviour of the Trust model defined in Section 5.3. This experiment is performed with all normal nodes i.e. no malicious nodes involved. The experiment is started with an initial value of trust for the IP. There are two cases considered for this experiment. In the first case only a single context i.e. only one parameter *availability_d* is varied. In the second case multiple context i.e. two parameter *availability_d* and *time_d* are considered for variation.

In the first case where the positive evidence for the parameter *availability_d* is increased it is observed that with the increase in the positive evidence for the *availability_d* parameter the overall Trust for the IP increases. In the second case the positive evidence for *availability_d* and *time_d* is simultaneously increased and it is observed that the rate of Trust increase is higher compared to the single context. This result is as shown in Figure 7.14. A similar effect is observed when the negative evidence is increased for one or more parameters.

7.4.7 Effect on Trust Due to Malicious Filtering

The aim of this experiment is to evaluate the trustworthiness computed by the model for the IP and ensure that it does not largely deviate due to the malicious nodes present in the system. This experiment is performed in two stages. In the first stage the trust value for the IP is computed without any malicious node present in the system i.e. node ratio of 100:0:0:0. In the second stage, malicious nodes with ratio of 70:30:0:0 is introduced and different filters are applied to observe the Trust value for the IP. The result of this experiment shows that the trust value obtained after introducing the positive exaggerated nodes with no filter (or filter=0) differs a lot from the original trust value with no malicious nodes. Due to the credibility defined in the trust model, the trust value does try to match the original trust value, but still there is a sizable difference between the two trust values. After introducing the malicious node filter of filter=30 and filter=n/2, the trust value nearly overlaps with the original trust that is obtained without the malicious node as shown in the Figure 7.16. For clarity, the Figure 7.16 shows the expanded view of the Figure 7.15.

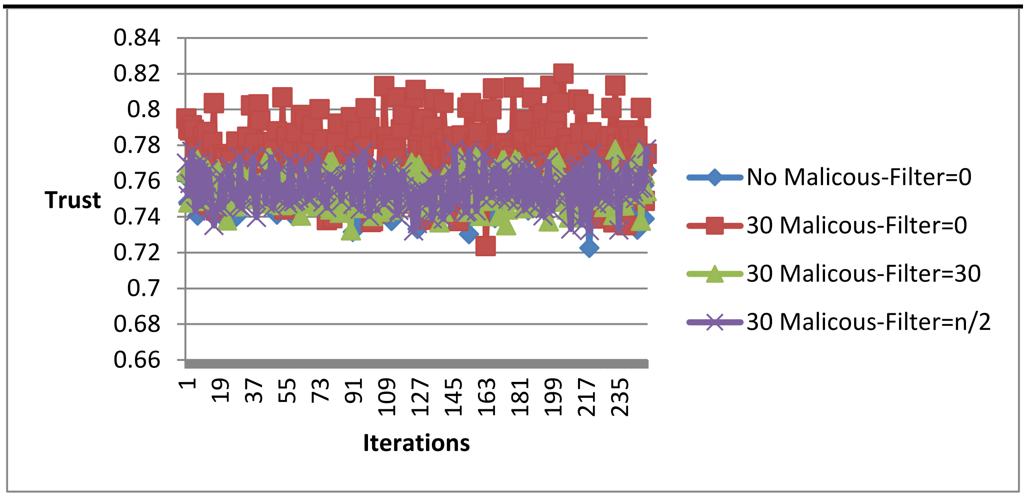


Figure 7.15 : Trust for different levels of filtering. SP node group ratio is 70:30:0:0

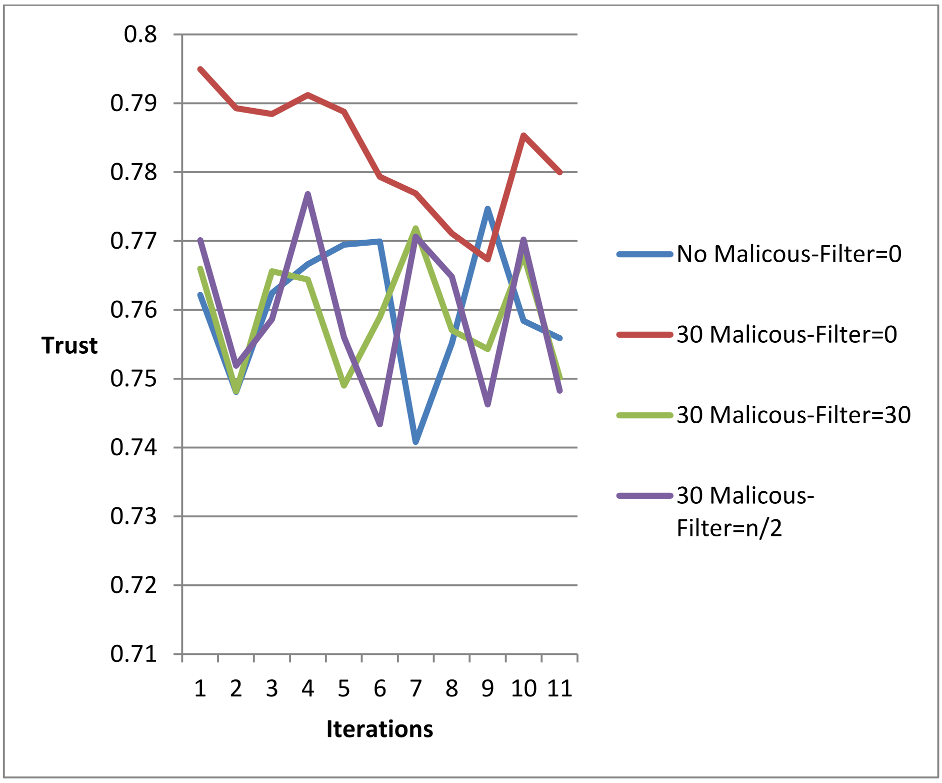


Figure 7.16 : Trust for different levels of filtering. SP node group ratio is 70:30:0:0 (expanded view of Figure 7.15)

7.5 Potential Threats to the experimental evaluation

In order to perform robustness measurement of the trust model against malicious feedback providers that are representative of real environments, parameters such as feedback parameter, the ratio malicious groups (normal group (G1), exaggerated positive group (G2), exaggerated negative group (G3) and complementary group (G4) and filter parameter are considered for the simulation and these parameters are varied from their lower bound to its higher bound. Also different combination of the parameters used in the simulation to evaluate the robustness of the trust model aligns the simulation to a real environment.

The simulation experiments performed in this thesis use statistical methods to generate and assign values to the parameters that allow to generalize the experiment results from statistical perspective.

The trust model proposed in this thesis is evaluated and shown for its robustness against unfair ratings and collusion attacks performed by malicious feedback providers. However, the trust model is not yet evaluated against other type of attacks such as Re-entry, Value imbalance and Sybil attacks and hence no assurance can be provided for the robustness of the trust model against these attacks.

7.6 Conclusion

This chapter evaluated the cloud broker architecture and the two trust models proposed that considered different aspects of the cloud environment.

- The evaluation of the cloud broker architecture demonstrates that the deployment time overhead due to the mediation layer of the cloud broker is significantly small.

- The evaluation of the opinion based trust model for optimized cloud services considered mainly SLA monitoring along with SP ratings and SP behaviour. This allows the SP to obtain trustworthiness of cloud service provider not only based on the historical information of transactions but also based on the performance of the on-going transactions.

- The trust model based on cloud characteristics comprises of dimensions that consider the essential cloud characteristics and encompasses the suitable parameters required to model the trustworthiness of the cloud service providers. The various evaluations performed for this model signifies the suitability, correctness and the robustness of this trust model in the cloud environment.

The trust model proposed in Chapter 4 & Chapter 5 and the evaluation performed on these trust models in this chapter signifies that the trust can be analysed for the cloud entities considering different features of the cloud service providers.

Chapter 8 Conclusion and Future Work

This thesis presented the trust assessment framework and trust models to assess the cloud service providers with the use of cloud broker architecture. This chapter summarizes achievements of the work described in this thesis and outlines the main contribution of the thesis. The thesis concludes with discussions on the open issues that may be addressed in the future work.

8.1 Achievement

Trust models defined for environments such as electronic market environments, peer-to-peer network, multi-agent systems are not suitable for the cloud computing environment. This thesis presented the trust assessment framework and trust models tailored for the cloud environments. The motivation of the work presented in this thesis arises from the study of cloud which reveals that cloud computing environment exhibiting the cloud characteristics are largely regulated with the SLAs which is the primary differentiator from other environment for performing trust assessments. In addition to this the various deployment architectures and service delivery models available in cloud computing environment makes the direct use of traditional trust framework and trust models unsuitable in the cloud.

Within the scope of this thesis we analysed the trust requirements in cloud computing environment and provided an approach to evaluate the trustworthiness of the cloud service providers. The investigation of trust framework requirements in cloud computing environment resulted in the need for a mediation to deal with challenges such as service level agreements, security measures and computation of trust and to provide a trustworthy environment. In this thesis, we presented a cloud broker as a mediation layer to deal with complex decision of selecting trustworthy cloud service providers. The cloud broker architecture presented in this thesis supports different modes of operation which assists in having a variety of trust assessments such as: a) Trust of cloud providers b) Security Reputation and c) Trust of group of providers for multi-cloud deployment. Chapter 3 describes the various modes operation of the cloud broker and also define the potential trust assessments within the cloud broker modes.

The investigation of trust requirements in cloud computing environment also resulted in the need for having an evaluation framework such that malicious providers cannot manipulate the evaluation process and that the cloud providers should be evaluated based on the fine grained QoS parameters together with feedbacks and recommendations. The trust models presented in this thesis incorporates cloud transactions information based on SLAs wherein historical transaction information as well as information of transaction in progress is considered for the trust assessment. The trust model incorporates parameters specific to the cloud characteristics for assessment of cloud service providers which ensures suitability of this trust model within cloud environments. Chapter 4 and Chapter 5 describe the trust model that considers SLA and parameters associated with cloud characteristics. The trust model presented in Chapter 4 and Chapter 5 is fundamentally based on the opinion representation. Chapter 4 presented an uncertainty model for the opinion representation in the trust model. The uncertainty model aids in increasing the accuracy of the trust assessment. An evaluation of the uncertainty model with a real data set from Amazon market place reveals that it reduces the prediction error in the trust model which is presented in the Chapter 7. The trust model presented in this thesis is extended with an early filtering mechanism and a credibility model, to resist reputation attacks. The early filtering mechanism that uses outlier method to filter exceptions in the feedback assists in excluding malicious feedback providers. The credibility model is primarily used to reduce the influence of the malicious feedback providers. Chapter 7 demonstrates cases of malicious groups providing feedback and it is observed that early filtering mechanism and credibility model jointly improves the robustness of the trust model providing resistance to reputation attacks.

Chapter 6 presents the detailed architecture of the cloud broker that supports the various modes of operation. The cloud architecture is implemented and validated in the EU funded OPTIMIS project using the components of OPTIMIS project. The cloud broker architecture supports modes of operation that enables provision to value added services such as security services. The support for security value added services in the cloud broker architecture encourages establishing a security trust/reputation framework. Chapter 6 also presents the use of cloud broker architecture for security reputation of the cloud service providers with the use of trust models defined in Chapter 4 and Chapter 5.

The trust model presented in this thesis is also implemented and used as one of the core components in the OPTIMIS project for providing TREC (Trust, Risk, Eco-efficiency and Cost) based optimized cloud service. The OPTIMIS base toolkit ("OPTIMIS Toolkit," n.d.) provides functionalities such as TREC assessment tools that allows a) *TREC based deployment solution*: generates optimized deployment solution based on TREC factors after negotiating with the cloud service providers; b) *TREC based optimisation of IP operation* : uses TREC assessment tools with various low level managers to create a self-managed cloud infrastructure, driven by cloud provider's BLOs (Business Level Objectives); and c) *TREC based VM management* : perform an efficient management of infrastructure resources by managing physical nodes and the VMs running on top of them.

8.2 Open Research Issues

This thesis presents a cloud broker architecture that aids in providing variety of trust assessments in its different modes of operation. In the cloud broker as cloud service recommendation, the cloud broker provides trust/reputation of the individual providers while in the cloud broker as cloud service intermediation the broker has capabilities for providing security reputation of the provider. The trust models presented in this thesis is used to assess the trustworthiness of individual cloud provider as well as the security reputation. However the proposed use of the cloud broker as cloud service aggregation/arbitration to assess multiple cloud providers requires a detailed analysis in order to devise a trust model for multi-cloud and federated cloud deployment architectures.

The trust models presented in this thesis is thoroughly evaluated with real data set and simulations for providing cloud service recommendations of individual cloud providers. However the security reputation framework proposed in this thesis, in its current stage is very abstract and requires an intense analysis and evaluation. Chapter 6 presents the high level architecture of a cloud broker, for assessing the security reputation of cloud providers, considering the various security assessments. Cyber security assessment, governance, risk and compliance frameworks, when combined with trust and reputation systems can provide a means of reducing potential risk of using cloud services while increasing consumer confidence and offer additional incentives for cloud provider to increase their level of compliance. However, at present, the governance, cyber security and compliance frameworks for the cloud providers lack such a support. Security reputation architecture as proposed in Chapter 6 in conjunction with widely acceptable cyber security and cloud compliance framework can validate the results in realistic conditions. This will also require several other research questions to be answered such as: a) What are the attributes by which trust is evaluated for cloud service providers and how to relate them with cyber security and compliance criteria b) how to evaluate providers while preserving user privacy and secrecy of recommendations.

Considering the widespread use of social networks, the current trust model can be extended to include recommendations about the cloud service providers from social trust networks. This may include work towards defining a social trust graph and providing trust based recommendations. The work will require definition of relationships between cloud entities that will enable to create social trust graph and define recommendation model that will enable to get trust based recommendations from the social trust graph. The trust based recommendations can be included in the trust models defined in Chapter 4 and Chapter 5 to evaluate the trustworthiness of the cloud service providers

.

Appendix

Appendix A: Papers Published

- **IFIPTM 2012**

Pawar, P.S., Rajarajan, M., Nair, S.K., Zisman, A., 2012. Trust Model for Optimized Cloud Services, in: Dimitrakos, T., Moona, R., Patel, D., McKnight, D.H. (Eds.), Trust Management VI, IFIP Advances in Information and Communication Technology. Springer Berlin Heidelberg, pp. 97–112.

- **Cloudcomp 2012**

Pawar, P. S., Nair, S. K., El-Mousaa, F., Dimitrakos, T., Rajarajan, M. & Zisman, A. (2012). *Opinion Model Based Security Reputation Enabling Cloud Broker Architecture*. Paper presented at the CloudComp 2012 - 3rd International Conference on Cloud Computing, 24 - 26 Sep 2012, Vienna, Austria.

- **IFIPTM 2013**

Pawar, P.S., Rajarajan, M., Dimitrakos, T., Zisman, A., 2013. Trust Model for Cloud Based On Cloud Characteristics , Trust Management VII, IFIP Advances in Information and Communication Technology.

- **IFIPTM 2014**

Pawar, P.S., Rajarajan, M., Dimitrakos, T., Zisman, A., 2014. Trust Assessment Using Cloud Broker, Trust Management VIII, IFIP Advances in Information and Communication Technology.

- **SIN 2014**

Rahulamathavan, Y., Pawar, P.S., Burnap, P., Rajarajan, M., Rana, O. F., Spanoudakis, G., 2014. Analysing Security requirements in Cloud-based Service Level Agreements, SIN 2014 – The 7th International Conference on Security of Information and Networks, 9-11th September 2014, Glasgow, UK. (To be published)

- **CloudCom 2014**

Daniel, J., Dimitrakos, T., El-Moussa, F., Ducatel, G., Pawar, P., Sajjad, A., 2014. Seamless Enablement of Intelligent Protection for Enterprise Cloud Applications through Service Store. IEEE, pp. 1021–1026. doi:10.1109/CloudCom.2014.92

- **IFIPTM 2015**

Pawar, P.S., Sajjad, A., Dimitrakos, T., Chadwick, D.W., 2015. Security-as-a-Service in Multi-cloud and Federated Cloud Environments, in: Damsgaard Jensen, C., Marsh, S., Dimitrakos, T., Murayama, Y. (Eds.), Trust Management IX. Springer International Publishing, Cham, pp. 251–261.

Daniel, J., El-Moussa, F., Ducatel, G., Pawar, P., Sajjad, A., Rowlingson, R., Dimitrakos, T., 2015. Integrating Security Services in Cloud Service Stores, in: Damsgaard Jensen, C., Marsh, S., Dimitrakos, T., Murayama,

Y. (Eds.), Trust Management IX. Springer International Publishing, Cham, pp. 226–239.

Appendix B: Patent Applications

- **Evaluating Software Compliance**

European patent application: WO2014202934 (A1)

- **Augmented Deployment Specification for Software Compliance**

European patent application: WO2014202933 (A1)

- **Enforcing Software Compliance**

European patent application: WO2014202932 (A1)

- **Model Based Enforcement of Software Compliance**

European patent application: WO2014202931 (A1)

- **Categorizing Software Application State**

European patent application: WO2014202930 (A1)

- **Application Broker for Multiple Virtualised Computing Environments**

European patent application: WO2014202929 (A1)

Appendix C: OPTIMIS (Optimized Infrastructure Services)

OPTIMIS is a FP7 EU-funded project with an aim of optimizing the cloud services by producing an architectural framework and a development toolkit. This will assist the cloud service providers to supply optimized services based on different aspect, such as trust, risk, eco-efficiency, cost (TREC) and legal. The optimization covers the full cloud service lifecycle that includes construction, deployment and operation of cloud services.

The OPTIMIS project also aims for three main use cases:

- Programming model validation through lifecycle management of on-demand services
- Extended elasticity via transparent cloud bursting
- Cloud brokerage and federation involving many cloud providers

In this Chapter, we propose a cloud broker architecture supporting different modes of operation that assists in the trust evaluation of the cloud service providers. The design and implementation of the cloud broker architecture is composed of several OPTIMIS components, integrated to coherently work to provide the cloud brokering functionality.

C.1 OPTIMIS Cloud broker components

The OPTIMIS (Ferrer et al., 2012) cloud broker is a component-based architecture formed by service composition, content delivery, service discovery and negotiation components to support inter-cloud operations. We describe below the major components used for the cloud broker (CBR) service:

C.1.1 Service Manifest

Service manifest is a document that describes the service and its components. The service consumers can specify the deployment and operational requirement for service which may include VM (Virtual Machine) images, thresholds for TREC levels, affinity constraints, location constraints, elasticity requirements, security requirements, and legal requirements. The cloud service providers can specify the billing plans, service provider endpoint reference, and resource definitions (Rasheed et al., 2012). OPTIMIS project includes standard approach to specify service manifest structure and APIs to manipulate the manifest. OPTIMIS also implements the Service Level Agreement (SLA) framework that evaluates the state of the parameters specified in the service manifest. OPTIMIS project uses existing WSAG4J framework, implementing WS-Agreement and WS-Agreement negotiation and defines new term languages in the OGF (Open Grid Forum) for Trust, Risk, Eco-efficiency, Cost, Data Security, Data Protection and Security. Since the service manifest is part of the SLA in the OPITMIs project, it creates contractual relationship between the consumer and cloud service provider. This allows the provider to plan its resource utilization and the commitments made to the consumer Figure a shows the abstract structure of the service manifest. In OPTIMIS, the service manifest is composed of an XML document.

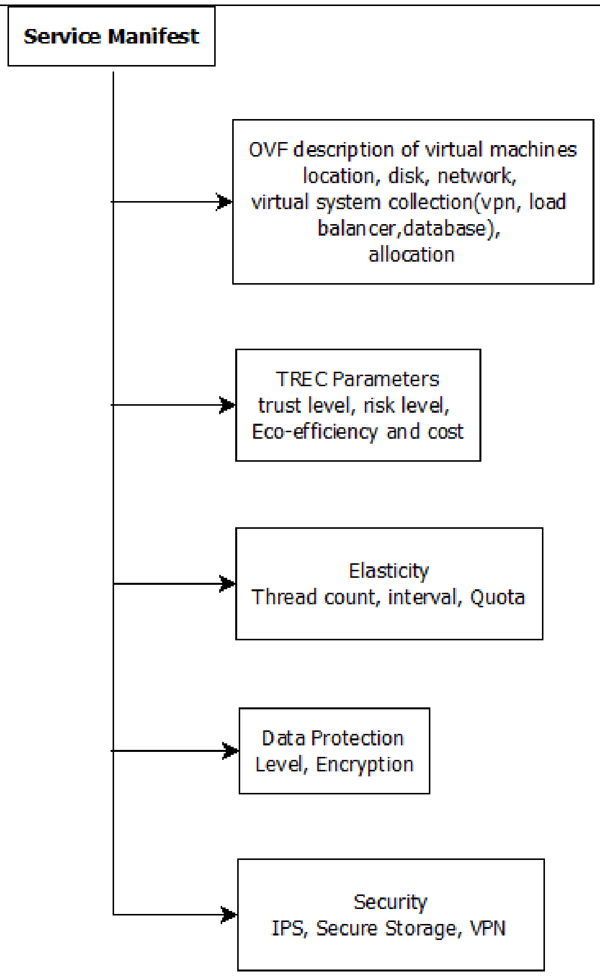


Figure a: Structure of service manifest

C.1.2 PM IDE (Programming Model - Integrated Development Environment)

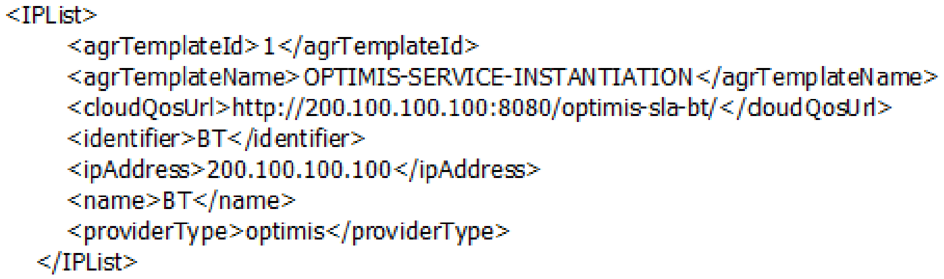
The OPTIMIS Programming Model (PM) simplifies cloud enablement of new applications by offering a run-time programming model which can be optionally used with Eclipse-based IDE ("OPTIMIS Programming Model plugin," n.d.). By introducing an abstraction layer, the OPTIMIS PM makes application development generic and independent of the underlying cloud infrastructure interfaces. It simplifies cloud application development by a simpler programming model based on sequential specifications of data and performance compliance described in the OPTIMIS service manifest. A run-time model provides optimal parallelism and multi-cloud distribution and performs run-time scheduling and optimization during application execution. In addition, a token-based software license management service supports elastic behavior. PM IDE supports creation of a service manifest. It also interacts with the image creation service for the creation of composite services according to the service manifest.

C.1.3 Image Creation Service (ICS)

The Image Creation Service allows the construction of VM images that embed the applications (developed with the programming model). It provides RESTFul web service for creating custom images. To create images, the requirements in the service manifest are matched with the image characteristics of the base image and the ICS maintains a list of base images.

C.1.4 IP Registry

The IP registry contains a list of all the cloud service providers. A consumer or cloud broker configures its own IP registry that contains the list of cloud service providers to be used. The structure of the registry contains the agrTemplateId Agreement Template Id), agrTemplateName (Agreement Template Name), cloudQosUrl (End point reference for negotiating and creating agreements), identifier (cloud provider identifier), ipAddress (cloud provider IP address), name (cloud provider name) and providerType (OPTIMIS or non-OPTIMIS cloud provider). Figure b provides a typical IP registry record that is stored in a form of an XML.



```
<IPList>
    <agrTemplateId>1</agrTemplateId>
    <agrTemplateName>OPTIMIS-SERVICE-INSTANTIATION</agrTemplateName>
    <cloudQosUrl>http://200.100.100.100:8080/optimis-sla-bt/</cloudQosUrl>
    <identifier>BT</identifier>
    <ipAddress>200.100.100.100</ipAddress>
    <name>BT</name>
    <providerType>optimis</providerType>
  </IPList>
```

Figure b : Example of an IP registry entry for a provider

C.1.5 SD (Service Deployer)

The SD (Li et al., 2012) is the core deployment manager of the consumer that interacts with the cloud broker for deployment of a service. As shown in Figure c, the SD performs the following steps to deploy the service: 1) Discovery of IPs: The cloud service providers are discovered by looking up the IP Registry. The consumer can configure the IP Registry to have the cloud broker as one of the cloud service provider 2) Negotiation: The consumer negotiates with the cloud service provider or the cloud broker to get offers for hosting the service 3) Service data transfer: The consumer uploads the service components to the cloud provider or broker 4) SLA creation: The consumer creates SLA with cloud service provider or broker based on the negotiated terms.

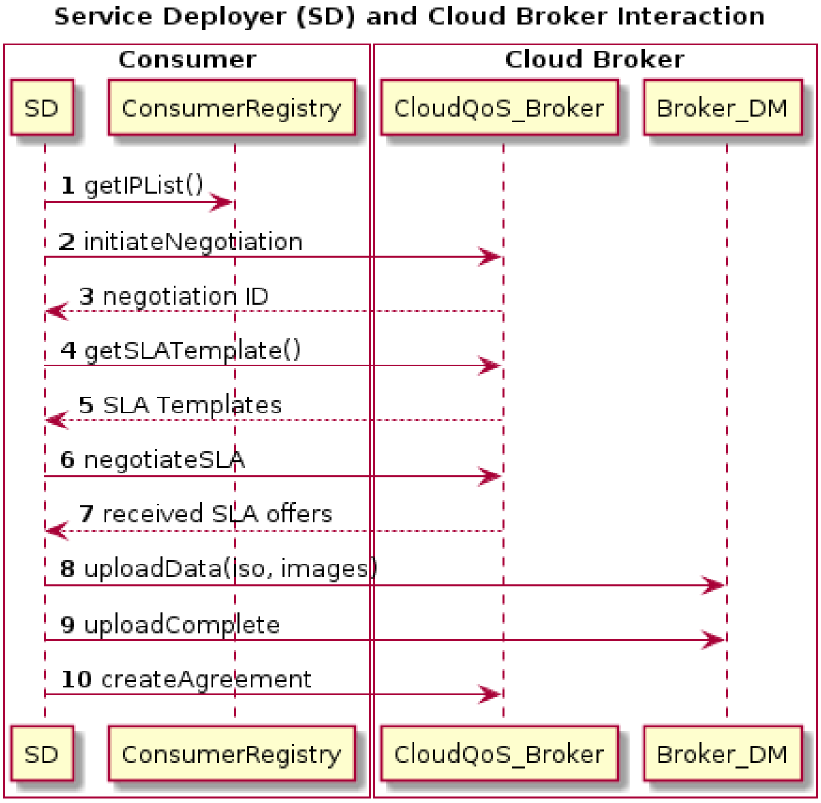


Figure c: SD and Cloud Broker interaction

C.1.6 SM (Service Manager)

The SM interface allows the consumer or cloud broker to deploy, un-deploy and redeploy services to cloud providers, by using the SD (Service Deployer). The SM keeps track of the deployed services and maintains the status of services during operation. Any changes to the service state such as stop, restart, scale-up/scale-down at the cloud service provider, during the operation are reported to the SM. The SM provides the service details that enables, fetching of the TREC values, monitoring values and getting the view of service data in the cloud providers.

C.1.7 TREC (Trust, Risk, Eco-efficiency and Cost)

TREC is a basic toolkit in OPTIMIS that essentially contains models to perform trust, risk eco-efficiency and cost evaluation of the consumers as well as cloud service providers.

Trust Framework: The trust framework determines trust levels for the consumers and the cloud service providers. The trustworthiness of the cloud service provider is determined based on the characteristics such as: runtime execution gap(reg), VM formation (vmf), IP reaction time(irt), SLA compliance(sc)(P. S. Pawar et al., 2012) and legal evaluations(le). The overall trustworthiness of the provider is computed by a fuzzy model that considers all the aspects.

Risk assessment: Risk assessment (Kiran et al., 2011) is performed at the following stages: 1) Provider risk assessment: the risk of dealing with a cloud provider before SLA request 2) Consumer risk assessment: the risk of dealing with a consumer before accepting SLA request 3) SLA request risk assessment: the cloud provider assesses the risk of accepting the SLA request 4) SLA offer risk assessment: the consumer assesses the risk of accepting the SLA offer 5) Risk assessment at operation: assess the risk of failure of physical hosts, VMs, services and the entire infrastructure of the cloud service providers

Eco-efficiency assessment: Eco-efficiency is defined as the ratio between the useful work performed and the energy consumed (Patterson, 1996). This definition is used to evaluate the relationship at the physical node level, virtual machine level and at service level. The cloud service provider computes energy efficiency (Katsaros et al., n.d.) at all levels which are further used by the placement algorithms to make decisions to optimize energy of the infrastructure of the cloud provider while at the same time fulfilling the SLAs of the consumers.

Cost assessment: Cost assessment (Molka and Byrne, 2013) is used to trace and predict the absolute cost of service operation. The resources that have been defined as having associated cost are: VCPU per unit, storage per GB, upstream network per GB, downstream network per GB, memory per MB and energy per kWHr. The cost model help understanding the cost of the cloud service provider for hosting a specific service. A weighted service-to-TCO-mapping model based on TCO influencing factors is used to determine the overall cost of a service for a cloud service provider during the service lifecycle.

C.1.8 DO (Deployment Optimizer)

The DO (Li et al., 2012) provides an optimal placement solution for each component of the service based on the TREC (Trust, Risk, Eco-efficiency, Cost) levels and ensuring the affinity constraints. The DO is provided with the service manifest document and the list of legally compliant providers. To device an optimal deployment solution, the DO splits the service manifest at a component level ensuring the affinity and anti-affinity constraints for the service components specified within the service manifest. The decomposed manifest is negotiated with list of cloud service providers for the service requirements and the TREC. The offers received for each decomposed manifest are evaluated to obtain an optimal solution for the entire service.

C.1.9 DM (Data Manager)

The OPTIMIS DM provides an OPTIMIS Distributed Files System (ODFS) that offers storage as a service spanning over different cloud providers. The OPTIMIS DM (Kousiouris et al., 2011) is a Hadoop based data management system as a front-end for cloud data management. The DM framework provides APIs and several components to extend Hadoops functionality beyond its backend. The DM supports data transfer and storage of the service components and the associated data of the service. The key features provided by the DM includes: 1) Location based data monitoring 2) Secure storage and key management 3) Seamless and interoperable exploitation of federated resources 4) Online predictions for future data activity 5) Validation of federated provider legal status. The legal constraints in the service manifest are checked against the DM of the cloud provider to validate the legal compliance of the cloud service provider. The shared storage functionality of the DM allows multi-cloud deployed service to share the data across its components.

C.1.10 VMC (Virtual Machine Contextualization)

The service components, which are mainly in the form of VM (virtual machine) images, may require additional information to launch the service. The OPTIMIS VMC (Armstrong et al., 2011)(Armstrong et al., 2013) supports embedding of various scripts that may be required for launching the service. The application is configured with all the general settings during the construction phase of the application and the VMC contextualizes with provider specific settings during the deployment phase. The contextualization is applicable to all aspects of configuring the service/application VM from virtual hardware to multi-tier software stacks, without the need to customize the guest VM. Contextualization of services is essential for interoperability, during the deployment of service across multiple providers and to support Value Added Services. The Contextualizer supports capabilities to either prepare VM images agnostic of operating system used or create ISO CDROM images that contain context data. The creation of VM images is essential for interoperable environment while the ISO images that contain context data and data processing scripts are mounted for the manipulation of data at runtime.

C.1.11 VMM (Virtual Machine Manager)

The VMM is responsible for performing efficient management of infrastructure resources by managing physical nodes and VM running on these physical nodes, during the whole lifecycle. It implements placement optimization policies based on TREC. The Placement Optimizer component of VMM re-organizes the mapping of VMs to physical resources for optimal placement while the Infrastructure Optimizer component of VMM is aimed to turn on/off physical resources depending on the load handled by the cloud infrastructure provider. VMM also provides interoperability with many cloud infrastructure provider solutions such as OpenNebula, EMOTIVE, OpenStack.

C.1.12 AC (Admission Control)

AC (Konstanteli et al., 2011)(Konstanteli et al., 2012) is responsible for checking whether a service or a set of services can be accepted in the OPTIMIS Cloud and to generate an optimal TREC-driven allocation pattern. The AC also considers the requirements of consumer’s new service, current work load of the cloud infrastructure provider, TREC values of the cloud providers, as well as the capacity planning. The admission control provides optimal allocation of elastic services on virtualized resources by incorporating a probabilistic approach in terms of availability guarantees.

C.1.13 Cloud QoS (Cloud Quality of Service)

This component (Rasheed et al., 2012) supports the negotiation and agreement creation between the consumers and the cloud service providers for the service to be deployed. This component implements the WS-Agreement (Andrieux et al., 2004) and WS-Negotiation and decouples the negotiation from agreement creation and service deployment. The separate negotiation and agreement layers allow both single step negotiation and multi-round negotiation. The service manifest in this section is used for the negotiation and agreement creation which contains the individual terms negotiated.

C.1.14 MO (Monitoring)

The Monitoring (Katsaros et al., n.d.) Infrastructure allows the runtime state of physical infrastructure, virtual infrastructure, and applications to be captured, stored, and analyzed. The monitoring component provides comprehensive monitoring of the service parameters which are essentially used by the TREC component. The collected information at different levels (physical, virtual and service) is stored and RESTFul web service APIs are provided to extract this information. It supports monitoring of OPTIMIS as well as non-OPTIMIS IP environments.

C.1.15 CO (Cloud Optimizer)

The CO is mainly responsible for the placement of the service component taking into consideration the non-functional constraints, legal requirements, and elasticity requirements of the service.

C.1.16 Broker Core

This component implements the core functions of the cloud broker service and uses the other OPTIMIS components to complete the cloud broker functionality. The core functions implemented within this component include: service manifest decomposition, multi-cloud service deployment functionality, and integrated framework for value-added services. For the cloud broker service, the components integrated with the Broker core component include DM for cloud broker data management service, VMC for interoperability and value added services, IP Registry for cloud provider discovery, TREC used for optimized deployment and operation of service and the VAS such as VPN, IPS and secure storage.

C.1.17 Value added services

The value added services primarily consists of services that the consumer may essentially require for its service deployed on the cloud providers via the cloud broker. Currently, the architecture is enabled with VPN, secure storage, and IPS as valued added services.

VPN (Virtual Private Network) Overlay: VPN overlay offers secure communication between the components of a service deployed on multiple cloud platforms. The VPN overlay is designed for a scalable and robust secure communication framework (Rajarajan et al., 2012). The VPN overlay employs the flexibility and scalability afforded by structured peer-to-peer overlays to join virtual machines running on different cloud IaaS providers with each other using IPSec tunnels, hence providing confidentiality, authentication, and integrity for all the data exchanged between different components of the service. This value-added service needs minimal manual configuration as peers automatically discover the information needed to perform their operations from a Universal Overlay of super-peers managing this service. The VPN overlay architecture also provides a distributed and scalable key management solution for the consumption of the virtual machines to set-up the secure communication channels.

IPS (Intelligent Protection System): The value-added service of Intelligent Protection System (IPS) is an integration of some of the traditional security services from the cloud broker for more efficient security management, faster provisioning in accordance of security policies and easier administration of common security tasks, as well as a visualisation of the security environment and its on-going status. It includes sub-services like firewall, intrusion prevention via deep packet inspection, security patching, anti-virus, and anti-malware. Each VM is contextualized by the VMC, which involves the installation and configuration of an IPS agent that can communicate securely with the cloud broker to take security actions according to the applicable policies.

Secure Storage: The Secure Storage value-added service provides the facilities of data protection for the multi-cloud environment. It enables the user to encrypt virtual storage volumes provided by the cloud platform and then mount that volume in a VM at run-time according to application requirements. The decryption keys are controlled by a Key Management Service (KMS) hosted at the CBR, which allows for policy-based key release and deny operations. Each component of Genome application in the form of VM uses the encrypted storage that ensures protection of application data in the cloud.

C.1.18 Genomic Application

To illustrate the cloud broker architecture and implementation, we consider a Genomic application that is used across the deployment scenarios described in this chapter (Royo et al., 2008).

The genetic information of patients is a key for an efficient treatment of several diseases and a genomic application considered here helps in the identification of genes which cause a disease. The successful identification of genes in an automatic way provides scientists with valuable information that allows them to perform functional analysis at all levels. The genomic application implements a combination of different existing genomic services with sequence comparison algorithm to help on gene detection from genomic DNA sequence. A composition of these services is invoked to obtain the reference data and prepare the DNA sequence in the suitable format for the computation. This computation calculates the comparison of the pre-processed DNA sequence with the reference data which identifies the most relevant genes. For each of these genes a deep analysis is performed and its results are post-processed with other genomic services producing the final report delivered to the researchers.

This genomic application is implemented as a service using the programming model and IDE.

The genomic application contains five different components. 1) Genome Formatting 2) Similar Sequence Retrieval and Format 3) Relevant Gene Search 4) GeneWise 5) Genome Application GUI.

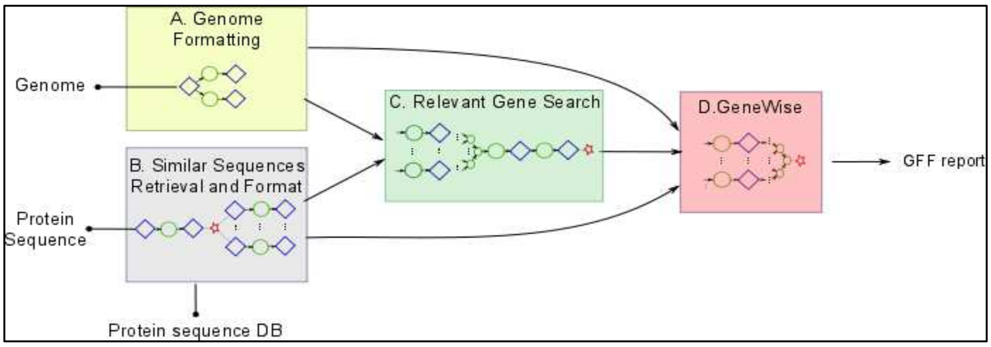
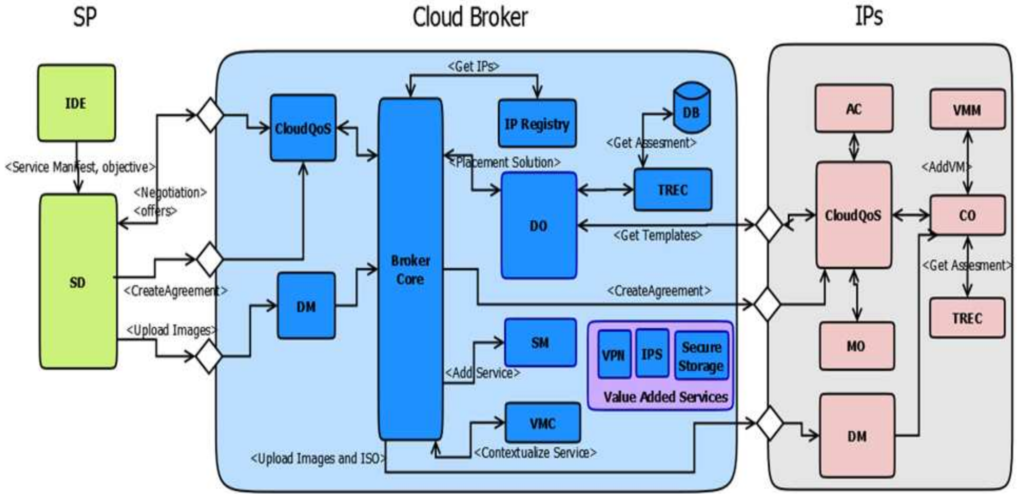


Figure d : Components of Genomic Application

C.1.19 Cloud Broker Used as an Arbitrage

The CBR used as arbitrage provides an outlook of a cloud provider, to the consumer that wants to deploy its service. CBR used as arbitrage not only takes the responsibility of finding the optimal solution for the service (as recommender), but also performs the multi-cloud deployment (aggregation) and provides Value Added Services (intermediation). The CBR also takes charge of the service during operational mode to monitor its performance and take critical decisions such as scale up/down, start/stop, or relocate the service components, reducing the complete workload of the consumer to monitor its service. The architecture of the cloud broker shown in Figure e used as arbitrage can be well explained with the deployment scenario of the Genomic application described in section 6.2.1.



IDE – Integrated Development Environment
SM – Service Manager
DM – Data Manager
DB – Database
VPN – Virtual Private Network
CloudQoS – Cloud Quality of Service
CO – Cloud Optimizer
VMM – Virtual Machine Manager
SD – Service Deployer
DO – Deployment Optimizer
VMC- Virtual Machine Contextualizer
TREC – Trust, Risk, Eco-efficiency, Cost
IPS – Intelligent Protection System
MO – Monitoring
AC – Admission Control

Figure e : High level component architecture of the Cloud Broker

In the following we describe the sequence of steps performed by the CBR used as arbitrage:

Step 1: Create Service manifest and construct the application: The PM-IDE provides a graphical interface which allows the consumer to specify the requirements of the Genomic application. The Genomic application consists of five components and the consumer specifies cpu, memory, disk requirements for each of the components. The consumer also specifies the TREC requirements, elasticity requirements, affinity and anti-affintiy constraints related to each of the components. The consumer further specifies the security requirement, the legal constraints and finally creates the manifest.

The consumer creates the Genomic service using PM-IDE (Programing Model - Integrated Development Environment) which in turn invokes the *ICS* (Image Creation Service) for service creation. The Genomic service created is in the form of VM images for each of the components; i.e., five VM images are created for the five components of this service as described in section 6.2.1.

Step 2: Initiate negotiation with CBR

The IDE passes the *service manifest* to the SD for deployment of the service. The SD is configured to interact with the cloud broker and uses *service manifest* to negotiate the terms with the cloud broker's CloudQoS component. The CBR receives the service manifest through the negotiation request sent by the SD.

Step 3: Perform Legal Check

The IP registry at the CBR may contain a huge list of cloud providers, but for this scenario, Figure f shows that the CBR has an IP Registry that is configured to have three cloud infrastructures (Provider-1, Provider-2 and Provider-3) that CBR uses for deployment of the service. The *service manifest* received by the CBR for the Genomic application contains location based legal constraint that specify that all the application components should be deployed within the European region. Each of the cloud infrastructures in the IP registry is checked for the location constraint specified in the manifest, using the Data Manager service to perform legal check. The cloud providers Provider-1 and Provider-2 are legally compliant but since the Provider-3 is legally non-compliant it is filtered out due to its presence being outside the Eurpean Union.

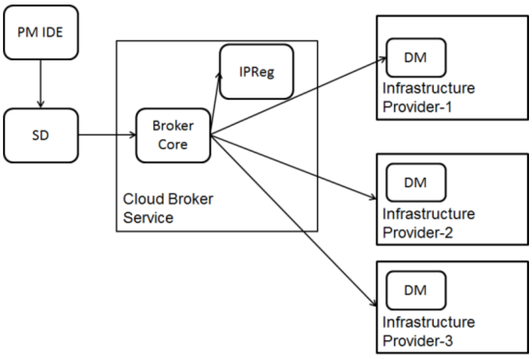

PM IDE
SD
IPReg
Broker Core
Cloud Broker Service
DM
Infrastructure Provider-1
DM
Infrastructure Provider-2
DM
Infrastructure Provider-3

Figure f : Legal compliant check

Step 4: Get deployment solution

The CBR now contains the two legally compliant cloud providers (Provider-1 and Provider-2) which are provided to the DO (Deployment Optimizer) along with the service manfiest for getting the optimal solution for deploying the Genomic service.

To obtain a optimal solution, the DO mainly performs two steps: 1) Decompose the service manifest 2) Negotiate the decomposed manifest with the cloud provider

- Decompose the service manifest

To decompose the service manfiest, the DO component uses several constraints that are specified in the service manifest. In this scenario, we show the decomposition of the service manifest mainly due to the two affinity and two anti-affinity constraints in the form of rules, mentioned for the Genomic service components in the manifest.

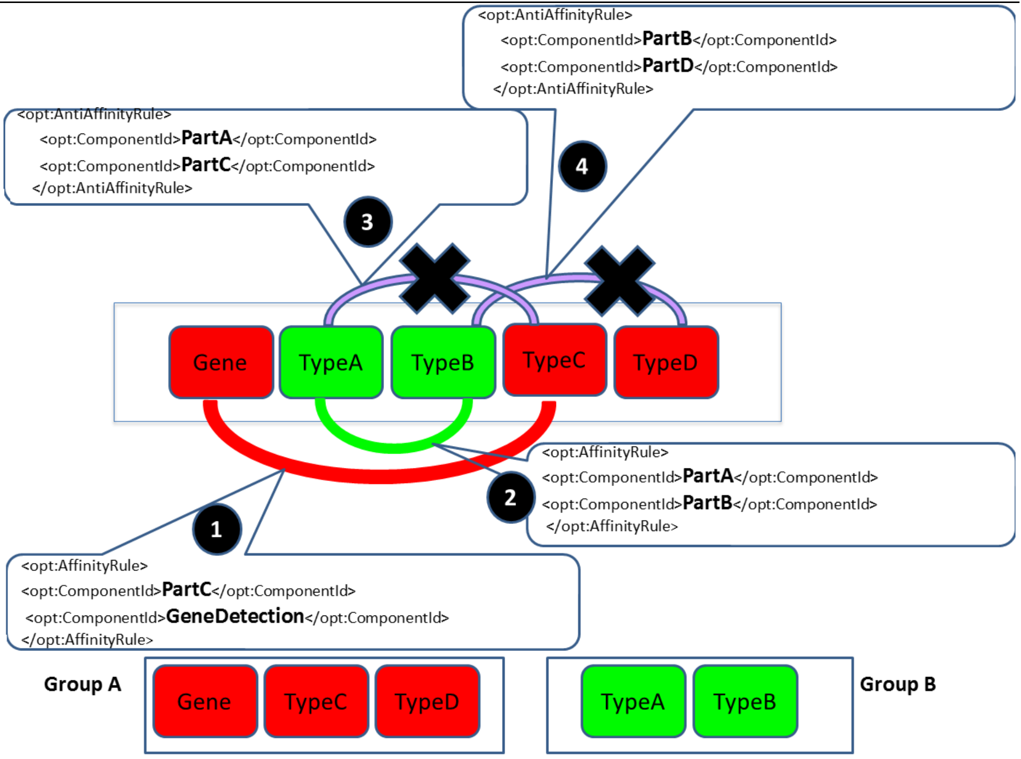

<opt:AntiAffinityRule>
<opt:ComponentId>PartB</opt:ComponentId>
<opt:ComponentId>PartD</opt:ComponentId>
</opt:AntiAffinityRule>
<opt:AntiAffinityRule>
<opt:ComponentId>PartA</opt:ComponentId>
<opt:ComponentId>PartC</opt:ComponentId>
</opt:AntiAffinityRule>
4
3
Gene
TypeA
TypeB
TypeC
TypeD
<opt:AffinityRule>
<opt:ComponentId>PartA</opt:ComponentId>
<opt:ComponentId>PartB</opt:ComponentId>
</opt:AffinityRule>
2
1
<opt:AffinityRule>
<opt:ComponentId>PartC</opt:ComponentId>
<opt:ComponentId>GeneDetection</opt:ComponentId>
</opt:AffinityRule>
Group A
Gene
TypeC
TypeD
TypeA
TypeB
Group B

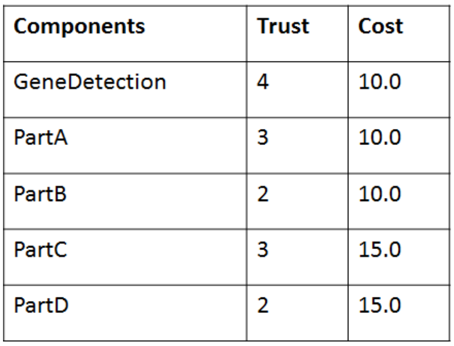
Figure g : Service manifest decomposition

As seen in Figure g, the first affinity rule requires the PartC and GeneDetection components to be available in the same cloud infrastructure provider environment. The second affinity rule requires PartA and PartB to be in the same cloud provider. The first anti-affinity rule requires that the PartA and PartC components should not be in the same cloud provider environment. This rule forms two logical groups of components ie. (GeneDetection, PartC) and (PartA, PartB). The second anti-affinity rule requires that PartB and PartD should not be in the same cloud provider environment. This rule separates the component PartD from the group (PartA, PartB). As there are only two cloud providers available that are compliant with Genomic service manifest, the manifest cannot be composed into more that two groups. The DO decomposes the manifest into two manifest. The first manifest contains the component grouped as (GeneDetection, PartC, PartD) and the second manifest contains the component grouped as (PartA, PartB).

- Negotiate the decomposed manifest with the cloud provider

The DO uses the decomposed manifest to negotiate with each of the legally compliant cloud providers. Apart from the resource requirements mentioned in the manifest, the DO additionally checks for the TREC constraints specified by the consumer.

The DO individually interacts with the TREC components to get the TREC assessment for each of the legally compliant cloud providers (Provider-1 and Provider-2) provided by the CBR. Figure h shows the minimum trust requirements and the maximum cost requirements per component and also the trust and cost evaluated for the cloud providers. The DO evaluates that for the decomposed manifest with group (GeneDetection, PartC, PartD) requires the provider to have trust level of atleast 4 and the maximum cost of 15.0 to fulfill the need of all the components in this manifest, which can be satisfied only by the cloud provider Provider-2. The manifest with component group (PartA, PartB) requires the trust level of atleast 3 and maximum cost of 10.0 which can be fulfilled only by the Provider-1.



Components	Trust	Cost
GeneDetection	4	10.0
PartA	3	10.0
PartB	2	10.0
PartC	3	15.0
PartD	2	15.0

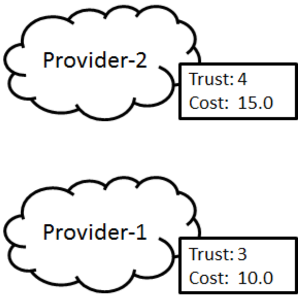


Figure h : Test-bed for deployment

The cloud providers that do not meet the TREC criteria specified in the service manifest are filtered by the DO. But for this scenario the DO identifies that the two cloud providers meet the TREC requirement of the service. The DO initiates SLA negotiations to receive offers from the cloud providers for the application to be deployed. In the process of negotiation, the CBR interacts with the AC which checks its current infrastructure status and the requirements of the Genomic application based on which it provides the offers.

The offers in the solution provided by DO are combined and customized by the Broker core component to provide a counter offer for the consumer.

Step 5 :Data upload and VM contextualization

If the offers are acceptable, the SD component of the consumer initiates the service data upload. The service manifest provided by the consumer contains the path of the service images that are created by using ICS (Image Creation Service). The CBR uses the decomposed manifest and uploads the service images to the DM of the respective cloud providers. The CBR further performs VM contextualization which bundles all the necessary configuration scripts essential for the component to start. The contextualization process creates the provider specific configurations in the form of an ISO files. The VMC creates the ISO files for the frond end component as well as for all other core components of the Genomic service. The ISOs created bundles for all the necessary configuration scripts essential for the components of the service to start.

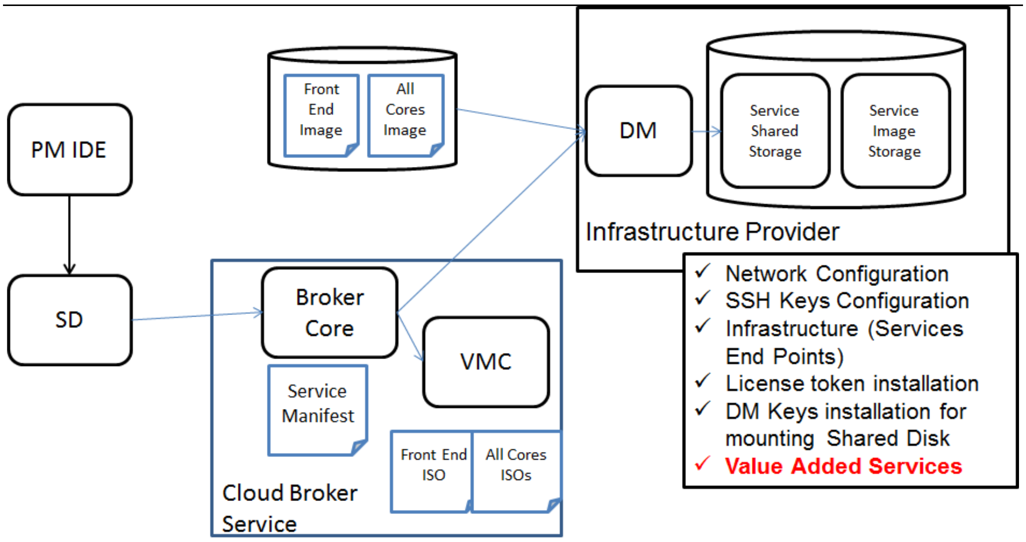


Figure i : Data upload Virtual Machine Contextualization

Figure i, the VMC contextualization process updates the service manifest with the configurations related to Network, SSH keys, Infrastructures, License token installations and DM keys installation for mounting Shared Disk. The VMC also performs contextualization for Value Added Services by including the IPS, VPN and Secure storage agents into the respective ISOs based on the specifications of the service manifest. The ISOs containing the contextualized information are uploaded to the respective cloud providers

Step 6: Agreement creation

The final stage of the deployment process is the creation of the agreements as shown in Figure j. The SD component of the consumer after SLA negotiation and uploading of service data initiates a *create agreement request* to the CBR. The CBR gets the context of this request and further follows with creating agreements with the multiple-cloud providers. The agreement creation with the cloud provider involves interaction with CloudQoS component of the cloud provider which in turn interacts with the AC (Admission Control) and CO (Cloud Optimizer) of the cloud provider to start the Genomic service components. The successful agreement creation with the cloud providers return the agreement IDs to the CBR and the CBR in turn returns the agreement ID of the agreement created with the consumer.

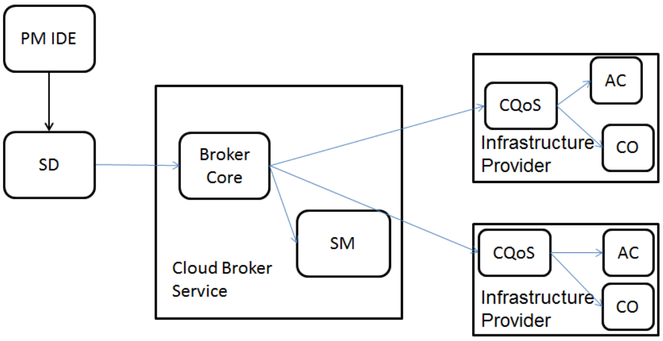


Figure j : Agreement creation

The successful agreement creation of the CBR with the cloud providers starts the Genomic service VMs in the cloud providers environment. The CO of the cloud at the provider extracts the end point reference of the CBRs SM (Service Manager) from the service manifest and registers the details of the service into the SM. The start of the service component or VM, mounts all the contextualised ISO files that contains necessary information for the component to launch successfully.

References

Abdul-Rahman, A., Hailes, S., 2000. Supporting Trust in Virtual Communities, in: Proceedings of the 33rd Hawaii International Conference on System Sciences-Volume 6 - Volume 6, HICSS ’00. IEEE Computer Society, Washington, DC, USA, p. 6007–.

Alhamad, M., Dillon, T., Chang, E., 2010. SLA-Based Trust Model for Cloud Computing, in: 2010 13th International Conference on Network-Based Information Systems (NBiS). Presented at the 2010 13th International Conference on Network-Based Information Systems (NBiS), pp. 321–324. doi:10.1109/NBiS.2010.67

Al-Shammari, S., Al-Yasiri, A., n.d. Defining a Metric for Measuring QoE of SaaS Cloud Computing. 15th Annu. Postgrad. Symp. Converg. Telecommun. Netw. Broadcast.

Amazon Web Services [WWW Document], n.d. URL http://aws.amazon.com/ (accessed 5.22.13).

Amour, B.S., 2014. A Subjective Logic Library Constructed Using Monadic Higher Order Functions. Tech. Rep.

Andrieux, A., Czajkowski, K., Dan, A., Keahey, K., Ludwig, H., Nakata, T., Pruyne, J., Rofrano, J., Tuecke, S., Xu, M., 2004. Web services agreement specification (WS-Agreement), in: Global Grid Forum.

Armstrong, D., Djemame, K., Nair, S., Tordsson, J., Ziegler, W., 2011. Towards a Contextualization Solution for Cloud Platform Services. IEEE, pp. 328–331. doi:10.1109/CloudCom.2011.51

Armstrong, D., Espling, D., Tordsson, J., Djemame, K., Elmroth, E., 2013. Runtime virtual machine recontextualization for clouds, in: Euro-Par 2012: Parallel Processing Workshops. pp. 567–576.

Arning, A., Agrawal, R., Raghavan, P., 1996. A linear method for deviation detection in large databases, in: International Conference on Knowledge Discovery and Data Mining. pp. 164–169.

Artz, D., Gil, Y., 2007. A survey of trust in computer science and the semantic web. Web Semant. Sci. Serv. Agents World Wide Web 5, 58–71.

Bardhan, S., Milojicic, D., 2012. A mechanism to measure quality-of-service in a federated cloud environment, in: Proceedings of the 2012 Workshop on Cloud Services, Federation, and the 8th Open Cirrus Summit. ACM, pp. 19–24.

Blaze, M., Feigenbaum, J., Ioannidis, J., Keromytis, A.D., 1999. The role of trust management in distributed systems security, in: Secure Internet Programming. Springer, pp. 185–210.

Bonatti, P.A., Shahmehri, N., Duma, C., Olmedilla, D., Nejdl, W., Baldoni, M., Baroglio, C., Martelli, A., Coraggio, P., Antoniou, G., Peer, J., Fuchs, N.E., 2004. Rule-based Policy Specification: State of the Art and Future Work.

Bonatti, P., Duma, C., Olmedilla, D., Shahmehri, N., 2005. An Integration of Reputation-based and Policy-based Trust Management, in: In Proceedings of the Semantic Web Policy Workshop.

Brandic, I., Dustdar, S., Anstett, T., Schumm, D., Leymann, F., Konrad, R., 2010. Compliant Cloud Computing (C3): Architecture and Language Support for User-Driven Compliance Management in Clouds. IEEE, pp. 244–251. doi:10.1109/CLOUD.2010.42

Bruneo, D., 2014. A Stochastic Model to Investigate Data Center Performance and QoS in IaaS Cloud Computing Systems. IEEE Trans. Parallel Distrib. Syst. 25, 560–569. doi:10.1109/TPDS.2013.67

Castelfranchi, C., Falcone, R., 1998. Principles of trust for MAS: Cognitive anatomy, social importance, and quantification, in: Multi Agent Systems, 1998. Proceedings. International Conference on. IEEE, pp. 72–79.

Comuzzi, M., Kotsokalis, C., Spanoudakis, G., Yahyapour, R., 2009. Establishing and Monitoring SLAs in Complex Service Based Systems. IEEE, pp. 783–790. doi:10.1109/ICWS.2009.47

Dalbert, C., 2009. Belief in a just world. Handb. Individ. Differ. Soc. Behav. 288–297.

Dellarocas, C., 2000. Immunizing online reputation reporting systems against unfair ratings and discriminatory behavior, in: Proceedings of the 2nd ACM Conference on Electronic Commerce. pp. 150–157.

De Meo, P., Nocera, A., Quattrone, G., Rosaci, D., Ursino, D., 2009. Finding reliable users and social networks in a social internetworking system, in: Proceedings of the 2009 International Database Engineering & Applications Symposium, IDEAS '09. ACM, New York, NY, USA, pp. 173–181. doi:10.1145/1620432.1620450

Demirkan, H., Goul, M., Soper, D.S., 2005. Service level agreement negotiation: A theory-based exploratory study as a starting point for identifying negotiation support system requirements, in: System Sciences, 2005. HICSS'05. Proceedings of the 38th Annual Hawaii International Conference on. IEEE, p. 37b–37b.

Denoeux, T., 2000. A neural network classifier based on Dempster-Shafer theory. Syst. Man Cybern. Part Syst. Hum. IEEE Trans. On 30, 131–150.

Di Modica, G., Tomarchio, O., Vita, L., 2009. A framework for the management of dynamic SLAs in composite service scenarios, in: Service-Oriented Computing-ICSOC 2007 Workshops. Springer, pp. 139–150.

Dumitrescu, C., Raicu, I., Foster, I., 2005. Di-gruber: A distributed approach to grid resource brokering, in: Proceedings of the 2005 ACM/IEEE Conference on Supercomputing. p. 38.

Ferrer, A.J., Hernández, F., Tordsson, J., Elmroth, E., Ali-Eldin, A., Zsigri, C., Sirvent, R., Guitart, J., Badia, R.M., Djemame, K., Ziegler, W., Dimitrakos, T., Nair, S.K., Kousiouris, G., Konstanteli, K., Varvarigou, T., Hudzia, B., Kipp, A., Wesner, S., Corrales, M., Forgó, N., Sharif, T., Sheridan, C., 2012. OPTIMIS: A holistic approach to cloud service provisioning. Future Gener. Comput. Syst. 28, 66–77. doi:10.1016/j.future.2011.05.022

Foster, H., Spanoudakis, G., 2011. Advanced service monitoring configurations with SLA decomposition and selection, in: Proceedings of the 2011 ACM Symposium on Applied Computing. ACM, pp. 1582–1589.

Gambetta, D., 2000. Can We Trust Trust? Gambetta Diego Ed Trust Mak. Break. Coop. Relat. Electron. Ed. Dep. Sociol. Univ. Oxf. Chapter 13, pp 213–237.

Garg, S.K., Versteeg, S., Buyya, R., 2013. A framework for ranking of cloud computing services. Future Gener. Comput. Syst. 29, 1012–1023. doi:10.1016/j.future.2012.06.006

Gartner, n.d. Cloud Services Brokerage: The Dawn of the Next Intermediation Age [WWW Document].

Gourlay, I., Djemame, K., Padgett, J., 2008. Reliability and risk in grid resource brokering, in: Digital Ecosystems and Technologies, 2008. DEST 2008. 2nd IEEE International Conference on. pp. 437–443.

Grandison, T., Sloman, M., 2000. A survey of trust in internet applications. Commun. Surv. Tutor. IEEE 3, 2–16.

Habib, S.M., Ries, S., Muhlhauser, M., 2011. Towards a Trust Management System for Cloud Computing. IEEE, pp. 933–939. doi:10.1109/TrustCom.2011.129

Habib, S.M., Ries, S., Muhlhauser, M., 2010. Cloud Computing Landscape and Research Challenges Regarding Trust and Reputation. IEEE, pp. 410–415. doi:10.1109/UIC-ATC.2010.48

Halberstadt, A., Mui, L., 2001. Group and reputation modeling in multi-agent systems, in: Proceedings of the Goddard/JPL Workshop on Radical Agents Concepts. NASA Goddard Space Flight Center.

Hasan, M.S., Huh, E.-N., 2013. Maximizing SLA and QoE in Heterogeneous Cloud Computing Environment, in: GCA'13 The 2013 International Conference on Grid Computing and Applications.

He, Q., Yan, J., Jin, H., Yang, Y., 2009. ServiceTrust: Supporting Reputation-Oriented Service Selection, in: Baresi, L., Chi, C.-H., Suzuki, J. (Eds.), Service-Oriented Computing, Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 269–284.

Hoffman, K., Zage, D., Nita-Rotaru, C., 2007. A Survey of attacks on Reputation Systems.

Hofmann, T., 1999. Probabilistic latent semantic analysis, in: Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence. pp. 289–296.

Hogan, M., Liu, F., Sokol, A., Tong, J., 2011. Nist cloud computing standards roadmap. NIST Spec. Publ. 35.

Hoover, W.E., Rockville, M.D., 1984. Algorithms For Confidence Circles and Ellipses (NOAA Techical Report No. 107 C&GS 3). US Department of

Commerce, National Oceanic and Atmospheric Administration, National Ocean Service, Washington, DC, USA.

Huang, J., Nicol, D.M., 2013. Trust mechanisms for cloud computing. J. Cloud Comput. Adv. Syst. Appl. 2, 9. doi:10.1186/2192-113X-2-9

Hwang, K., Kulkareni, S., Hu, Y., 2009. Cloud Security with Virtualized Defense and Reputation-Based Trust Mangement, in: Eighth IEEE International Conference on Dependable, Autonomic and Secure Computing, 2009. DASC '09. Presented at the Eighth IEEE International Conference on Dependable, Autonomic and Secure Computing, 2009. DASC '09, pp. 717–722. doi:10.1109/DASC.2009.149

Hwang, K., Li, D., 2010. Trusted cloud computing with secure resources and data coloring. Internet Comput. IEEE 14, 14–22.

IEEE SA - CPWG/2301 WG - Cloud Profiles WG (CPWG) Working Group [WWW Document], n.d. URL http://standards.ieee.org/develop/wg/CPWG-2301_WG.html (accessed 9.4.13).

IEEE SA - ICWG/2302 WG - Intercloud WG (ICWG) Working Group [WWW Document], n.d. URL http://standards.ieee.org/develop/wg//ICWG-2302_WG.html

Jia, C., Xie, L., Gan, X., Liu, W., Han, Z., 2012. A Trust and Reputation Model Considering Overall Peer Consulting Distribution. IEEE Trans. Syst. Man Cybern. Part Syst. Hum. 42, 164–177. doi:10.1109/TSMCA.2011.2162497

Jøsang, A., 2012. Robustness of Trust and Reputation Systems: Does It Matter?, in: Trust Management VI. Springer, pp. 253–262.

Jøsang, A., 2008. Abductive reasoning with uncertainty, in: The Proceedings of the International Conference on Information Processing and Management of Uncertainty (IPMU2008).

Jøsang, A., 2001. A logic for uncertain probabilities. Int. J. Uncertain. Fuzziness Knowl.-Based Syst. 9, 279–311.

Jøsang, A., 1997. Artificial reasoning with subjective logic, in: Proceedings of the Second Australian Workshop on Commonsense Reasoning. Perth, p. 34.

Jøsang, A., Bhuiyan, T., 2008. Optimal Trust Network Analysis with Subjective Logic. IEEE, pp. 179–184. doi:10.1109/SECURWARE.2008.64

Jøsang, A., Ismail, R., 2002. The beta reputation system. pp. 41–55.

Jøsang, A., Ismail, R., Boyd, C., 2007. A survey of trust and reputation systems for online service provision. Decis. Support Syst. 43, 618–644.

Jøsang, A., Keser, C., Dimitrakos, T., 2005. Can We Manage Trust?, in: Proceedings of the Third International Conference on Trust Management (iTrust), Versailes. Springer-Verlag, pp. 93–107.

Jøsang, A., Marsh, S., Pope, S., 2006. Exploring different types of trust propagation, in: Trust Management. Springer, pp. 179–192.

Jøsang, A., McAnally, D., 2005. Multiplication and comultiplication of beliefs. Int. J. Approx. Reason. 38, 19–51. doi:10.1016/j.ijar.2004.03.003

Jøsang, A., Pope, S., Diaz, J., Bouchon-Meunier, B., 2009. Dempster's rule as seen by little coloured balls. Manuscr. Submitt. J. Autom. Reason.

Jøsang, A., Sambo, F., 2014. Inverting Conditional Opinions in Subjective Logic.

Katsaros, G., Subirats, J., Fitó, J.O., Guitart, J., Gilet, P., Espling, D., n.d. A service framework for energy-aware monitoring and VM management in Clouds. Future Gener. Comput. Syst. doi:10.1016/j.future.2012.12.006

Kay, R.U., 2007. Fundamentals of the Dempster-Shafer theory and its applications to system safety and reliability modelling. Reliab. THEORY Appl.

Kerr, R., Cohen, R., 2009. Smart cheaters do prosper: defeating trust and reputation systems, in: Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems-Volume 2. pp. 993–1000.

Kiran, M., Jiang, M., Armstrong, D.J., Djemame, K., 2011. Towards a Service Lifecycle Based Methodology for Risk Assessment in Cloud Computing. IEEE, pp. 449–456. doi:10.1109/DASC.2011.89

Koehler, J.J., 1996. The base rate fallacy reconsidered: Descriptive, normative, and methodological challenges. Behav. Brain Sci. 19, 1–17.

Kokash, N., Birukou, A., D'Andrea, V., 2007. Web service discovery based on past user experience, in: Business Information Systems. Springer, pp. 95–107.

Konstanteli, K., Cucinotta, T., Psychas, K., Varvarigou, T., 2012. Admission Control for Elastic Cloud Services. IEEE, pp. 41–48. doi:10.1109/CLOUD.2012.63

Konstanteli, K., Varvarigou, T., Cucinotta, T., 2011. Probabilistic admission control for elastic cloud computing, in: Service-Oriented Computing and Applications (SOCA), 2011 IEEE International Conference on. pp. 1–4.

Koren, Y., Bell, R., Volinsky, C., 2009. Matrix Factorization Techniques for Recommender Systems. Computer 42, 30–37. doi:10.1109/MC.2009.263

Ko, R.K.L., Jagadpramana, P., Mowbray, M., Pearson, S., Kirchberg, M., Liang, Q., Lee, B.S., 2011. TrustCloud: A Framework for Accountability and Trust in Cloud Computing. IEEE, pp. 584–588. doi:10.1109/SERVICES.2011.91

Kousiouris, G., Vafiadis, G., Varvarigou, T., 2011. A Front-end, Hadoop-based Data Management Service for Efficient Federated Clouds. IEEE, pp. 511–516. doi:10.1109/CloudCom.2011.76

Koutrouli, E., Tsalgatidou, A., 2012. Taxonomy of attacks and defense mechanisms in P2P reputation systems—Lessons for reputation system designers. Comput. Sci. Rev. 6, 47–70. doi:10.1016/j.cosrev.2012.01.002

Krautheim, F.J., Phatak, D.S., Sherman, A.T., 2010. Introducing the trusted virtual environment module: a new mechanism for rooting trust in cloud computing, in: Trust and Trustworthy Computing. Springer, pp. 211–227.

Lei, J., Li, Y., 2014. Vector-Based Sensitive Information Protecting Scheme in Automatic Trust Negotiation. J. Netw. 9. doi:10.4304/jnw.9.4.927-931

Liu, Z., Squillante, M.S., Wolf, J.L., 2001. On maximizing service-level-agreement profits, in: Proceedings of the 3rd ACM Conference on Electronic Commerce. ACM, pp. 213–223.

Li, W., Svard, P., Tordsson, J., Elmroth, E., 2012. A general approach to service deployment in cloud environments, in: Cloud and Green Computing (CGC), 2012 Second International Conference on. pp. 17–24.

Li, X., Lyu, M.R., Liu, J., 2004. A trust model based routing protocol for secure ad hoc networks, in: Aerospace Conference, 2004. Proceedings. 2004 IEEE. IEEE, pp. 1286–1295.

Mahbub, K., Spanoudakis, G., 2011. Proactive SLA Negotiation for Service Based Systems: Initial Implementation and Evaluation Experience. IEEE, pp. 16–23. doi:10.1109/SCC.2011.34

Ma, H., King, I., Lyu, M.R., 2009. Learning to recommend with social trust ensemble, in: Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR ’09. ACM, New York, NY, USA, pp. 203–210. doi:10.1145/1571941.1571978

Mahmood, Z., 2011. Cloud Computing: Characteristics and Deployment Approaches. IEEE, pp. 121–126. doi:10.1109/CIT.2011.75

Ma, H., Yang, H., Lyu, M.R., King, I., 2008. Sorec: social recommendation using probabilistic matrix factorization, in: Proceedings of the 17th ACM Conference on Information and Knowledge Management. pp. 931–940.

Manuel, P.D., Thamarai Selvi, S., Barr, M.-E., 2009. Trust management system for grid and cloud resources, in: Advanced Computing, 2009. ICAC 2009. First International Conference on. IEEE, pp. 176–181.

Marilly, E., Martinot, O., Betgé-Brezetz, S., Delègue, G., 2002. Requirements for service level agreement management, in: IP Operations and Management, 2002 IEEE Workshop on. IEEE, pp. 57–62.

Massa, P., Avesani, P., 2007. Trust-aware recommender systems, in: Proceedings of the 2007 ACM Conference on Recommender Systems. pp. 17–24.

Massa, P., Avesani, P., 2004. Trust-aware collaborative filtering for recommender systems, in: On the Move to Meaningful Internet Systems 2004: CoopIS, DOA, and ODBASE. Springer, pp. 492–508.

Maximilien, E.M., Singh, M.P., 2004. Toward autonomic web services trust and selection, in: Proceedings of the 2nd International Conference on Service Oriented Computing. ACM, pp. 212–221.

Mcknight, D.H., Chervany, N.L., 1996. The Meanings of Trust.

Mell, P., Grance, T., 2011. The NIST Definition of Cloud Computing. Httpcsrcnistgovpublicationsnistpubs800-145SP800-145pdf.

Molka, K., Byrne, J., 2013. Towards predictive cost models for cloud ecosystems: Poster paper, in: Research Challenges in Information Science (RCIS), 2013 IEEE Seventh International Conference on. pp. 1–2.

Mui, L., Mohtashemi, M., 2002. A computational model of trust and reputation, in: In Proceedings of the 35th Hawaii International Conference on System Science (HICSS.

Mui, L., Mohtashemi, M., Ang, C., Szolovits, P., Halberstadt, A., 2001. Ratings in distributed systems: A bayesian approach, in: Proceedings of the Workshop on Information Technologies and Systems (WITS). pp. 1–7.

Nair, S.K., Porwal, S., Dimitrakos, T., Ferrer, A.J., Tordsson, J., Sharif, T., Sheridan, C., Rajarajan, M., Khan, A.U., 2010. Towards Secure Cloud Bursting, Brokerage and Aggregation, in: 2010 IEEE 8th European Conference on Web Services (ECOWS). Presented at the 2010 IEEE 8th European Conference on Web Services (ECOWS), pp. 189–196. doi:10.1109/ECOWS.2010.33

Nau, R.F., 2001. De Finetti was right: probability does not exist. Theory Decis. 51, 89–124.

Noor, T.H., Sheng, Q.Z., 2011. Trust as a service: a framework for trust management in cloud environments, in: Web Information System Engineering–WISE 2011. Springer, pp. 314–321.

Noor, T.H., Sheng, Q.Z., Alfazi, A., Law, J., Ngu, A.H., 2013a. Identifying fake feedback for effective trust management in cloud environments, in: Service-Oriented Computing-ICSOC 2012 Workshops. Springer, pp. 47–58.

Noor, T.H., Sheng, Q.Z., Zeadally, S., Yu, J., 2013b. Trust management of services in cloud environments: Obstacles and solutions. ACM Comput. Surv. 46, 1–30. doi:10.1145/2522968.2522980

Olmedilla, D., Rana, O., Matthews, B., Nejdl, W., 2005. Security and trust issues in semantic grids, in: Proceedings of the Dagsthul Seminar, Semantic Grid: The Convergence of Technologies. p. 10.

Optimis - Optimized Infrastructure Services [WWW Document], n.d. URL http://www.optimis-project.eu/ (accessed 2.19.14).

OPTIMIS Programming Model plugin [WWW Document], n.d. . Eclipse Plugins Bundles Prod. - Eclipse Marketpl. URL http://marketplace.eclipse.org/content/optimis-programming-model-plugin

OPTIMIS Toolkit [WWW Document], n.d. URL http://www.optimistoolkit.com/ (accessed 4.15.14).

Oren, N., Norman, T.J., Preece, A., 2007. Subjective logic and arguing with evidence. Artif. Intell. 171, 838–854. doi:10.1016/j.artint.2007.04.006

Palacios, M., Garcia-Fanjul, J., Tuya, J., Riva, C. de la, 2010. A Proactive Approach to Test Service Level Agreements. IEEE, pp. 453–458. doi:10.1109/ICSEA.2010.77

Palacios, M., Garcia-Fanjul, J., Tuya, J., Spanoudakis, G., 2015. Coverage-Based Testing for Service Level Agreements. IEEE Trans. Serv. Comput. 8, 299–313. doi:10.1109/TSC.2014.2300486

Patterson, M.G., 1996. What is energy efficiency?: Concepts, indicators and methodological issues. Energy Policy 24, 377–390. doi:10.1016/0301-4215(96)00017-1

Pawar, P.S., Nair, S.K., El-Mousaa, F., Dimitrakos, T., Rajarajan, M., Zisman, A., 2012. Opinion Model Based Security Reputation Enabling Cloud Broker Architecture. Presented at the CloudComp 2012 - 3rd International Conference on Cloud Computing, Vienna, Austria.

Pawar, P.S., Rajarajan, M., Nair, S.K., Zisman, A., 2012. Trust Model for Optimized Cloud Services, in: Dimitrakos, T., Moona, R., Patel, D., McKnight, D.H. (Eds.), Trust Management VI, IFIP Advances in Information and Communication Technology. Springer Berlin Heidelberg, pp. 97–112.

Pearson, S., 2013. Privacy, security and trust in cloud computing, in: Privacy and Security for Cloud Computing. Springer, pp. 3–42.

Pujol, J.M., Sangüesa, R., Delgado, J., 2002. Extracting reputation in multi agent systems by means of social network topology, in: Proceedings of the First International Joint Conference on Autonomous Agents and Multiagent Systems: Part 1, AAMAS ’02. ACM, New York, NY, USA, pp. 467–474. doi:10.1145/544741.544853

Rajarajan, M., Sajjad, A., Zisman, A., Nair, S.K., Dimitrakos, T., 2012. Secure communication using dynamic VPN provisioning in an Inter-Cloud environment. Presented at the ICON 2012: 18th IEEE International Conference on Networks, Singapore.

Rasheed, H., Rumpl, A., Waldrich, O., Ziegler, W., 2012. A Standards-Based Approach for Negotiating Service QoS with Cloud Infrastructure Providers. Presented at the eChallenges 2012 conference, IIMC - International Information Management Corporation, Lisbon, Portugal.

Rasmusson, L., Jansson, S., 1996. Simulated social control for secure Internet commerce, in: Proceedings of the 1996 Workshop on New Security Paradigms, NSPW ’96. ACM, New York, NY, USA, pp. 18–25. doi:10.1145/304851.304857

Resnick, P., Kuwabara, K., Zeckhauser, R., Friedman, E., 2000. Reputation systems. Commun ACM 43, 45–48. doi:10.1145/355112.355122

Resnick, P., Zeckhauser, R., 2002. Trust among strangers in Internet transactions: Empirical analysis of eBay’s reputation system. Adv. Appl. Microecon. 11, 127–157.

Royo, R., Lopez, J., Torrents, D., Gelpi, J., 2008. A BioMoby-Based Workflow for Gene Detection Using Sequence Homology. Int. Supercomput. Conf. ISC 08 Dresd. Ger.

Sabater, J., Sierra, C., 2002. Reputation and social network analysis in multi-agent systems, in: Proceedings of the First International Joint Conference on Autonomous Agents and Multiagent Systems: Part 1. pp. 475–482.

Sabater, J., Sierra, C., 2001a. REGRET: reputation in gregarious societies, in: Proceedings of the Fifth International Conference on Autonomous Agents. ACM, pp. 194–195.

Sabater, J., Sierra, C., 2001b. Social ReGreT, a reputation model based on social relations. ACM SIGecom Exch. 3, 44–56.

Salakhutdinov, R., Mnih, A., 2008. Probabilistic matrix factorization. Adv. Neural Inf. Process. Syst. 20, 1257–1264.

Santos, N., Gummadi, K.P., Rodrigues, R., 2009. Towards trusted cloud computing, in: Proceedings of the 2009 Conference on Hot Topics in Cloud Computing. San Diego, California, pp. 3–3.

Saravanan, M.M.K., Kantham, M.L., 2013. an Enhanced QOS Architecture Based Framework for Ranking of Cloud Services. Int. J. Eng. Trends Technol. IJETT 4, 1022–1031.

Shafer, G., 1992. The Dempster-Shafer theory. Encycl. Artif. Intell. 330–331.

Shao, J., Wang, Q., 2011. A Performance Guarantee Approach for Cloud Applications Based on Monitoring. IEEE, pp. 25–30. doi:10.1109/COMPSACW.2011.15

Spanoudakis, G., LoPresti, S., 2009. Web Service Trust: Towards a Dynamic Assessment Framework, in: Availability, Reliability and Security, 2009. ARES'09. International Conference. IEEE, pp. 33–40. doi:10.1109/ARES.2009.149

Srivatsa, M., Liu, L., 2007. Secure Event Dissemination in Publish-Subscribe Networks, in: 27th International Conference on Distributed Computing Systems, 2007. ICDCS '07. Presented at the 27th International Conference on Distributed Computing Systems, 2007. ICDCS '07, pp. 22–22. doi:10.1109/ICDCS.2007.136

Sundareswaran, S., Squicciarini, A., Lin, D., 2012. A Brokerage-Based Approach for Cloud Service Selection, in: 2012 IEEE 5th International Conference on Cloud Computing (CLOUD). Presented at the 2012 IEEE 5th International

Conference on Cloud Computing (CLOUD), pp. 558–565. doi:10.1109/CLOUD.2012.119

Venugopal, S., Buyya, R., Winton, L., 2006. A Grid service broker for scheduling e-Science applications on global data Grids: Research Articles. Concurr Comput Pr. Exper 18, 685–699. doi:10.1002/cpe.v18:6

Wang, Y., Singh, M.P., 2010. Evidence-based trust: A mathematical model geared for multiagent systems. ACM Trans. Auton. Adapt. Syst. TAAS 5, 14.

Whitby, A., Jøsang, A., Indulska, J., 2004. Filtering out unfair ratings in bayesian reputation systems, in: Proc. 7th Int. Workshop on Trust in Agent Societies.

Winsborough, W.H., Li, N., 2002. Towards practical automated trust negotiation, in: Policies for Distributed Systems and Networks, 2002. Proceedings. Third International Workshop on. IEEE, pp. 92–103.

Wu, H., Siegel, M., Stiefelhagen, R., Yang, J., 2002. Sensor fusion using Dempster-Shafer theory [for context-aware HCI], in: Instrumentation and Measurement Technology Conference, 2002. IMTC/2002. Proceedings of the 19th IEEE. IEEE, pp. 7–12.

Xianrong Zheng, Martin, P., Brohman, K., Li Da Xu, 2014. CLOUDQUAL: A Quality Model for Cloud Services. IEEE Trans. Ind. Inform. 10, 1527–1536. doi:10.1109/TII.2014.2306329

Yang, Y., Sun, Y.L., Kay, S., Yang, Q., 2009. Defending online reputation systems against collaborative unfair raters through signal modeling and trust, in: Proceedings of the 2009 ACM Symposium on Applied Computing. pp. 1308–1315.

Yao, J., Chen, S., Wang, C., Levy, D., Zic, J., 2010. Accountability as a Service for the Cloud. IEEE, pp. 81–88. doi:10.1109/SCC.2010.83

Yu, B., Singh, M.P., 2001. Towards Probabilistic Model of Distributed Reputation Management, in: In: Proceedings of the Fourth Workshop on Deception, Fraud and Trust in Agent Societies. Montreal, Canada.

Zadeh, L.A., 1986. A simple view of the Dempster-Shafer theory of evidence and its implication for the rule of combination. AI Mag. 7, 85.

Zhang, Z., Feng, X., 2009. New Methods for Deviation-Based Outlier Detection in Large Database, in: Sixth International Conference on Fuzzy Systems and Knowledge Discovery, 2009. FSKD ’09. Presented at the Sixth International Conference on Fuzzy Systems and Knowledge Discovery, 2009. FSKD ’09, pp. 495–499. doi:10.1109/FSKD.2009.303

Zhao, H., Yu, Z., Tiwari, S., Mao, X., Lee, K., Wolinsky, D., Li, X., Figueiredo, R., 2012. CloudBay: Enabling an Online Resource Market Place for Open Clouds. IEEE, pp. 135–142. doi:10.1109/UCC.2012.40

Zhou, R., Hwang, K., 2007. PowerTrust: A Robust and Scalable Reputation System for Trusted Peer-to-Peer Computing. IEEE Trans. Parallel Distrib. Syst. 18, 460–473. doi:10.1109/TPDS.2007.1021