A SURVEY ON TRUST MANAGEMENT IN CLOUD COMPUTING

A Paper
Submitted to the Graduate Faculty
of the
North Dakota State University
of Agriculture and Applied Science

By

Maryam Roodaki

In Partial Fulfillment of the Requirements
for the Degree of
MASTER OF SCIENCE

Major Department:
Computer Science

March 2016

Fargo, North Dakota

North Dakota State University

Graduate School

Title

A SURVEY ON TRUST MANAGEMENT IN CLOUD COMPUTING

By

Maryam Roodaki

The Supervisory Committee certifies that this ***disquisition*** complies with North Dakota State University's regulations and meets the accepted standards for the degree of

MASTER OF SCIENCE

SUPERVISORY COMMITTEE:

Dr. Juan Li

Chair

Dr. Hyunsook Do

Dr. Na Gong

Approved:

4/12/2016

Date

Dr. Brian M. Slator

Department Chair

ABSTRACT

Nowadays the number of people that outsource their data to the cloud increases dramatically. Cloud computing offer cost-effective dynamic, scalable and shared services for enterprises from remote data centre. However, the problem of trusting cloud computing is a paramount concern for most enterprises in such a way that trust is widely regarded as one of the top obstacles for the adoption and growth of cloud computing. There are a lot of methods proposed by researchers to help the consumers identify the cloud service provider who seems to be more reliable. These trust-aided unified evaluation framework help in measuring the trustworthiness of cloud service providers. This work provides a generic analytical framework to compare different trust models based on a set of assessment criteria to help cloud consumers find the trust model that best satisfies their trust concerns in cloud computing.

ACKNOWLEDGMENTS

I would also like to thank Professor Jen (Juan) Li from North Dakota State University for her comments and suggestions throughout the thesis, as she has considerably helped shape and structure this work, especially when it comes to methodological aspects.

Also, I would like to thank my family for their unconditional support and the love of my life, Farshid for his endless help and support.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Figure Page

LIST OF ABBREVIATIONS

AC ..... Attribute Certificate

AA ..... Attribute Authority

AAA ..... Attribute Assertion Authority Certificate

AAC ..... Authentication and Authorization Center

ACA ..... Attribute Certification Authority

AICPA ..... American Institute of Certified Public Accountants

AIDB ..... Application Information Database

AIK ..... Attestation Identity Key

AMRep ..... Accurate and Multi-faceted Reputation

AP ..... Attestation Provider

API ..... Application Programming Interfaces

AR ..... Attestation Requestor

ARVTM ..... Application-Oriented Remote Verification Trust Model

BTC ..... Build Trust on Cloud

CA ..... Certification Authority

CAIQ ..... Consensus Assessments Initiative Questionnaire

CC ..... Cluster Consumer

CCM ..... Cloud Controls Matrix

CDB ..... Credential Database

CDN ..... Content Delivery Network

DFET ..... Data-driven and Feedback-Enhanced Trust

CARDB ..... Credit Adjustment Rule Database

CRL ..... Certificate Revocation List

CSA ..... Cloud Security Alliance

CSCA .....Cloud Service Connection and Adaptation

CSP .....Cloud Service Providers

CARD .....Cloud Service Registry and Discovery Trust

CTP .....CloudTrust Protocol

DAA .....Direct Anonymous Attestation

DAPM .....Distributed Agent Publish and Management

DHT .....Distributed-Hash-Table

DITT .....Domain Inside Trust Table

DOTT .....Domain Outside Trust Table

DTM.....Dynamic Trust Monitor

DTP .....Distributed Trust Protocol

EDB.....Evidence Database

En .....Entropy

Ex .....Expected value

FBCT.....Family-gene Based model for Cloud Trust

FHRM .....Feedback-enhanced and Hierarchical Rating Mechanism

FIM .....Federated Identity Management

FISMA .....Federal Information Security Management Act

FRTM.....Fuzzy Reputation-Based Trust Model

HASBE .....Hierarchical Attribute Set Based Encryption

He.....Hyperentropy

IaaS .....Infrastructure as a Service

IOWA.....Induced Ordered Weighted Averaging

JVM.....Java Virtual Machines

MCMP.....Multiple Cloud Management Platform

MM .....Master Module

MTBAC .....Mutual Trust Based Access Control

MTCEM.....Multi-Tenancy Trusted Computing Environment Model

NIST.....National Institute of Standards and Technology

OM .....Operation Monitor

OS .....Operating System

OTDA .....Overall Trust Degree Aggregation

PaaS.....Platform as a Service

PCR.....Platform Configuration Registers

PII.....Personally Identifiable Information

PKI .....Public Key Infrastructure

QoS .....Quality of Service

RM .....Registration Manager

RU .....Recommending Users

RVT.....Risk-Value Table

TC .....Trusted Coordinator

TCE .....Trust Computation Engine

TCG.....Trusted Computing

TCCP.....Trusted Cloud Computing Platform Model

TCP .....Trusted Computing Platform

TCS .....TSS Core Services

TDDL.....TCG Device Driver Library

TEMRV.....Trust Evaluation Management based on Remote Verification Group

TFCC.....Trust Feedback Compliance Checker

TI.....Trust Information

TM.....Trust Manager

TN .....Trusted Node

TPM .....Trusted Platform Module

TP SLA monitor.....Third-Party Service Level Agreement monitor

TREASURE.....Trust Enhanced Security for Cloud Environments

TT.....Trust Table

TSE .....Trust Semantics Engine

TSP.....TSS Service Provider

TSS.....Trusted Platform Software Stack

TUE.....Trust Update Engine

TVMCM .....Trusted VM Clone Model in Cloud Computing

UAP.....Unified Access Portal

RWFS.....Trust Work Flow Scheduling

S3 .....Amazon Simple Storage Service

SaaS.....Software as a Service

SLA .....Service Level Agreement

SOA.....Service Oriented Architecture

SSH .....Secure Shell

SSL.....Secure Sockets Layer

STAR .....Security, Trust & Assurance Registry

VPN.....Virtual Private Network

VM .....Virtual Machine

WSN.....Wireless Sensor Network

ZKC2P .....Zero-Knowledge Credibility Proof Protocol

CHAPTER 1. INTRODUCTION

Recently, cloud computing has been receiving much attention as a new computing paradigm for providing flexible and on-demand infrastructures, platforms, and software as services. According to National Institute of Standards and Technology (NIST) [1], "Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g. networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction". Cloud computing offers service dynamism, elasticity and wide variety of choices to enterprises. In today's competitive environment, enterprises cannot ignore these services. Flexible cloud computing services require one party (i.e. Cloud Consumer (CC)) rely on the actions of other party (i.e. Cloud Service Provider (CSP)), therefore, trust has become a vital component of such services.

Trust is a complex social phenomenon. Based on the concepts of trust developed in social sciences [2, 3], trust is a mental state comprising: (1) expectancy in which the trustor expects a specific behavior from the trustee such as providing valid information or effectively performing cooperative actions; (2) belief in which the trustor believes that the expected behavior occurs, based on the evidence of the trustee's competence, integrity, and goodwill; (3) willingness to take risk –in which the trustor is willing to take risk for that belief" [4]. Although intuitively easy to comprehend, the notion of trust has not been scholarly defined [5].

In order to use cloud services, an enterprise needs to give up control of its assets (i.e. data) to the CSP. Loss of control on stored data in cloud triggers uncertainty about data confidentiality, privacy, integrity and availability for CCs which adversely affects adoptability of cloud computing services. Enterprises have to remember that as compelling as cloud services are, it isn't without

potential problems. Amazon Simple Storage Service (S3), as an example of cloud services, had suffered an outage for several hours in February 2008 that resulted in numerous customer Web applications going offline [6, 7]. Enterprises also must consider the possibility that data could be stolen or viewed by people who are not authorized to see it. CSPs take all user critical information and put on virtual servers while users may never know if stored information will be used against their consent. CSP can be forced by government agencies to reveal stored data. For example, based on the U.S.A. Patriot Act [8] and other national security related laws, the U.S. government has the unfettered ability to obtain access to data stored inside and outside the United States by U.S. CSPs or their foreign subsidiaries. Same rules and concerns can be applied for other countries. For individual users of famous CSPs like Microsoft, Amazon, Apple and etc., the bigger risk is to lose the access to their online accounts that store numerous amount of personal data such as pictures and videos, email correspondences and banking information because of accusation of terms of service violation. Although, most of such cases can be resolved, it may take long time for communication with CSP and meanwhile users do not have access to their personal data. According to [9], trust management is ranked among the top 10 obstacles for adopting cloud computing. Adoptability of cloud services depends on establishment of trust on CSP to assure data security and guarantee cloud performance and behavior.

CSPs and consumers can take advantage of the benefits of cloud computing technologies when an effective trust management system is designed and implemented for cloud computing services, properly. To guarantee effectiveness of trust management in offered cloud computing services, on-going and state-of-the-art evaluation techniques are required. Evaluation of trust in cloud computing is a challenging issue since trust is subjective and case sensitive. In order to evaluate trust in cloud services, different methods and techniques have been proposed which are

commonly known as the “Trust Models” in literature [10-15]. For simplicity, existing trust models can be categorized into five trust mechanisms based on the common elements between literatures in this area [2, 14-19]. First category is reputation based trust mechanism which uses an aggregated opinion of a community in defining the trust worthiness of a cloud service. Second category is authentication based trust mechanism in which a CSP is trustworthy if it complies with a trusted policy. Third category is Service Level Agreement (SLA) based trust which is based on contracts and agreements signed by CSPs for the delivery of their services to CCs. Fourth category is domain based trust which divides the cloud into autonomous number of domains and uses authentication mechanisms for evaluating the trust worthiness of the entities inside or outside of the domain. The last category is platform based trust which uses a security assurance policy for a platform.

Trust in cloud computing has attracted lots of attention in recent years. There have been many trust models (such as [20 - 34]) proposed to study trust in cloud computing. Each trust model may be limited to certain features of cloud services. Therefore, it is vital for enterprises to be able to evaluate effectiveness of trust models based on a holistic view of most important issues in offered cloud services. People have realized the importance of trust in cloud computing towards vast adaptation of cloud service. There are researches focusing on the analysis and evaluation of existing trust models. For example, Kanwal and et al. [14] have studied twelve trust models and classified these trust models into five mechanisms. They have evaluated seven parameters for twelve studied trust models and quantified each parameter strength as low, medium or high. Corradini and et al. [16] have investigated fourteen trust models. They have categorized these trust models into three mechanisms to search for the major causes of lack of trust in each mechanism. They have summarized that lack of reliable and efficient evaluation system is a major barrier against adoption of cloud computing services. Rathi and et al. [16, 17] have investigated trust from

two different angles including trustworthiness of cloud user (CU) and trustworthiness of CSP. They have studied five different mechanisms to determine their disadvantages for trust on CSP and have concluded that lack of transparency is the major issue for establishing trust. However, they did not mention their evaluation method. Huang and et al. [18] have defined five different mechanisms and defined role of four cloud entities including CSP, cloud service, cloud auditor and cloud broker. They have examined each entity considering domain of expectancy and source of trust in the five mechanisms. Firdhous and et al. [19] have studied trust in the distributed systems. They have evaluated thirteen trust models in cloud computing using six evaluating parameters. Although they evaluated trust models based on different parameters, trust mechanisms are not included in their evaluation.

In spite of the existing surveys on trust in cloud, there is a need for a more comprehensive and up-to-date survey to investigate the most recent trust models, as more and more new trust models and techniques have been proposed and existing surveys did not cover them Moreover, previous surveys use their own ad hoc parameters to do the evaluation and comparison. This would cause confusion for enterprises to understand the criteria of trust evaluation and the main concerns about trust. This survey tries to address the aforementioned problems and help people understand trust issues in cloud computing. In this survey, we have systematically studied various trust models including the most up-to-date models to provide a holistic view and evaluation method for trust management in cloud computing. We have categorized fourteen assessment criteria widely used in existing research into six groups that evaluate the main concerns in trust management. Selected assessment criteria provide measureable criterions to compare trust models. We have identified possible risks and threats for trust management in cloud computing. Finally, we categorize existing trust models into five different mechanisms.

This thesis is organized into 5 chapters. The first chapter contains an introduction to the work completed. It also presents the motivation for the research work, and the objective of the thesis. Chapter 2 explains the theoretical baselines. Reviews the definition of trust and the trust mechanisms. Chapter 3 includes research strategy. Explains how we conducted the research and what parameters are we using to analyze the trust models. Chapter 4 reviews some of the latest and known trust models and discussed the characteristics of the models and how they help in calculating trust in cloud computing. Also, categorize the trust models into six categories and analyze them based on three set of parameters, system characteristics parameters, security evaluating parameters and trust evaluating parameters. Chapter 5 provides a discussion on how to use the suggested analytical framework. Chapter 6 contains the thesis conclusion. In this chapter we compared all the trust models discussed in chapter 4 based on system characteristics, security and trust evaluating parameters.

CHAPTER 2. OVERVIEW OF TRUST AND CLOUD COMPUTING

2.1. Cloud Computing

Cloud computing is a paradigm that provides flexible and on-demand infrastructures, platform and software as services. Cloud computing has emerged as a result of combining the benefits of grid computing [35] and virtualization with those of service-oriented computing [36] to utilize computer resources (data centers) and deliver computer resources as services. Cloud computing uses virtualization techniques to design and govern the services it offers to automate business logics. Cloud environments promise several benefits such as reduced expenses and simplicity to service providers and service requesters [35, 37]. For instance, it only took 24 hours, at the cost of merely $240, for the New York Times to archive its 11 million articles using a cloud service named Amazon Web Services [38].

Cloud services are established based on five essential characteristics [1].The first characteristic is the on demand self-service, which enables consumers to provision computing power, storage, networks and software in a simple and flexible way. Second is broad network access in which cloud service consumers can access available computing resources over the network. Third is resource pooling where computing resources are pooled to serve multiple cloud service consumers based on a multitenant model where physical and virtual computing resources are dynamically reassigned on demand, forth is rapid elasticity where computing resources are elastically provisioned to scale rapidly based on the cloud service consumers need, and the last one is measured service where computing resources usage is monitored, metered, controlled, and reported to provide transparency for both CSPs and consumers [1].

2.1.1. Cloud Service Models

Cloud services have three different models. One of the service models is Infrastructure as a Service (IaaS) which provides raw storage space, computing, or network resources for the customers to run and execute any software that they choose. The other service model is Platform as a Service (PaaS) which the CSP provides the hardware and a toolkit and a number of supported programming languages to build higher level services. The users who are typically software developers host their applications on the platform and provide these applications to the end-users. The third service model is Software as a Service (SaaS) which the CC is the end-user who just have access to the complete applications running on a cloud infrastructure and offered on a platform on-demand.

2.1.2. Infrastructure Deployment Models of Cloud

The different infrastructure deployment models are distinguishing by their architecture, the location of the datacenter where the cloud is realized, and the needs of the CC. Public clouds is one of the deployment models which runs applications from different CCs who share this infrastructure and pay for their resource utilization on a utility computing basis. Private clouds is another deployment model which is built for the exclusive use of one CC, who owns and fully controls this cloud. The third deployment model in cloud computing is Community clouds in which CCs who have similar requirements, can share an infrastructure and configuration and management of the cloud. The last deployment models is Hybrid clouds which consist of any composition of other deployment models.

2.1.3. Possible Risks and Threats of Cloud Computing

Trust is one of the main concerns for the consumers to adopt Cloud computing [39]. Based on the common elements between literatures in this area [39-42], cloud computing risks and threats are:

- **Lack of Confidentiality**

According to standard computing literature [43], the IT Security is depend on the Confidentiality of data. In cloud computing confidentiality is achieved by encryption. To achieve confidentiality the encryption schemes need to be secure for long term. Also, confidentiality is threatened by decrypting data while using it. Furthermore, information leakage vulnerability in third-party compute Clouds pose threats to Confidentiality too [44].

- **Lack of Reliability**

Availability of resources in Cloud Computing is one of the biggest concerns for the consumers [45, 46]. Availability is not just consist of reachability but also success rate of transactions. However, the CSPs look at availability as a way to represent the level of reliability of the cloud services. The availability that the CSPs claim they provide to their CCs which is more like %99, is not clearly defined as it is the availability of a single server or the availability of the servers resides in data centers in different locations of the world [46].

- **Lack of Identity Management**

Federated Identity Management (FIM) is an important terminology in the case of federated clouds. FIM provides tool for sharing resources and services among different enterprises while the directory services, authentication and authorization do not have same technologies. But it is possible that FIM have the enterprises to use authentication broker (a common trusted third party)

as identity management provider. However, this kind of application can consider as a threat for the security of the entire service inventory [39].

- **Lack of Privacy**

From the CCs perspective, privacy is an important concern in cloud computing and entails the protection and appropriate use of the personal information of CCs, and the meeting of expectations of CCs about its use. For organizations, privacy entails the application of laws, policies, standards and processes by which Personally Identifiable Information (PII) of individuals is managed. Context is very important as privacy threats differ according to the type of cloud scenario. Some cloud application areas and services might face a very low privacy threat, for example if the service is to process information that is public. It is only if the service handles personal information, in the sense of collecting, transferring, processing, sharing or storing it, that there could be a privacy risk and privacy needs to be taken into account [39]. Privacy becomes very important when multiple services need to be combined to enable a new service. For example, print on demand service in cloud which can be provided by combining a printing service with a storage service [40] can cause a privacy threat since the information regarding the services might need to flow across service providers' boundaries [40].

- **Lack of Reputation**

With the growing number of CSPs, the CCs are facing a challenge to select the best and most appropriate providers from numerous offers. In [42], the author points out a typical scenario, where a CSP can offer a secure service while another may not, if the latter charges half the price, the majority of organizations will ask for the latter one as there is no real way to explore the difference.

- **Lack of Service Level Agreements (SLAs)**

Standard SLAs in the present Cloud market are also one of the obstacles that the consumers face while adopting the services offered by the CSPs. Consumers might face problems that occur from vendor lock-in, insufficient security measures, data unavailability, hidden costs, and non-transparent infrastructure. In most cases, SLAs are created to protect the vendors/providers and not the CCs. Most of the above mentioned problems are overlooked in current SLAs offered by the CSPs [47].

- **Lack of transparency**

Providers of cloud computing technologies may be unlikely to share information about processes, operations, controls, and methodologies, especially related to IT general controls affecting the cloud environment. There are some transparent security principles help identify the types of information that should and should not be disclosed. Those that should be disclosed are: Common security features such as the use of firewalls and encryption of data in transmission or at rest should be disclosed because they are considered basic security features that most security people would expect to be in place anyway, performing disclosure when it is imperative due to a legal or regulatory requirement, Security architectural details that may either help or hinder security management should be disclosed, governance responsibilities of the CC versus those of the CSP should be clearly articulated so that CCs are clear on what they must do themselves to help protect their data. Also, there are some principals for which disclosure is not recommended which are: do not disclose anything that could create risk to the datacenter or to the integrity of data stored in the datacenter, if disclosure could create potential harm for a CC or partner, it should be avoided, avoid disclosures that could create undue liability for the CSP, if disclosure would result in breach of a legal or regulatory requirement, it should be avoided. [48]

2.2. Trust

The purpose of trusted computing is to solve some of today's security problems through hardware changes to personal computer. Trust is a crucial enabling factor in relations where there is uncertainty, interdependence, risk, and fear of opportunism [49]. [2, 3] has defined trust as a mental state which is consists of expectancy, belief and willingness to take risk. Based on this definition two types of trusts can be introduced [50]. One of them is trust in performance which is the trust about what the trustee performs. The other one is trust in belief which is the trust about what the trustee believes. The trustee's performance should be based on what he says or what he does. For example, if we consider x as what the trustee says then trust_p(d,e,x,k) represents that trustor d trusts trustee e regarding e's performance x in context k. This relationship means that if x is made by e in context k, then d believes x in that context [51].

$$* \, trust\_p(d, e, x, k) \equiv madeBy(x, e, k) \supset believe(d, k \supset x) \quad (2.1)$$

where $\supset$ is an operator used for reified propositions to mimic the logical operator for implication. A trust in belief relationship, $trust\_b(d, e, x, k)$, represents that trustor d trusts trustee e regarding e's belief (x) in context k. This trust relationship means that if e believes x in context k, then d also believes x in that context [51]:

$$trust\_b(d, e, x, k) \equiv believe(e, k \supset x) \supset believe(d, k \supset x) \quad (2.2)$$

McKnight et al. believes that trust develops gradually over time so it starts small and then gradually increases. They define initial trust as "trust in an unfamiliar trustee, a relationship in which the actors do not yet have credible, meaningful information about, or affective bonds with, each other". They also define trusting beliefs as the confident trustor perception that the trustee has attributes that are beneficial to the trustor. There are three types of trust beliefs. One of them is Competence which is the ability of the trustee to do what the trustor needs. The Second one is

benevolence and is defined as the trustee caring and motivation to act in the trustor's interests and the last one is integrity, which is the honesty of the trustee and promise keeping [48].

2.2.1. Apply Trust in Cloud Computing Models

In order to gain trust on CSPs, transparency and accountability play important role. Security, Trust & Assurance Registry (STAR) is a free publicly accessible registry program which is launched by Cloud Security Alliance (CSA) to increase the cloud transparency. This program helps CSPs to publish self-assessment of their security controls, in either a Consensus Assessments Initiative Questionnaire (CAIQ) or a Cloud Controls Matrix (CCM). CAIQ contains over 140 frequent questions that is useful for cloud users or auditors. CCM is a framework describing how a CSP aligns with the CSA security guide [53]. STAR is a useful source for users who are seeking for cloud services and the information offered is a CSP's self -assessment. Cloud Trust Protocol (CTP) [54], is a request-response mechanism for a cloud user to obtain specific information about the elements of transparency which includes aspects of configuration, vulnerability, audit log, service management, service statistics, and so forth and are applied to a specific CSP. The primary purpose of the CTP and the elements of transparency is to generate evidence-based confidence that everything that is claimed to be happening in the cloud is indeed happening as described, and nothing else [8]. CTP helps user's internal observations of cloud service operations by providing an interesting channel between cloud users and CSPs. One of the weakness of STAR and CTP is that its information is provided by CSP itself and if CSP be dishonest it can filter out or change data which would violate data's reliability based on trust judgement.

Trust is an important concept in distributed computing environments and plays a critical role in ensuring and enhancing system security and adaptability. There are a number of challenges that affect trust in different distributed systems for example in social networks, hackers are a

serious threat since they try to access the user accounts and use them as a trusted source to spread malwares. Also, tools that help in managing, viewing, querying, transferring and displaying personal data in the system and provide searching and mining profile data can be considered as another trust issue in social networks. In P2P networks security implications arise from abusing trust between peers. One of the trust issues exists in peer to peer networks is the distributed denial of service (DoS). Also, attackers can make use of the querying nature of P2P networks to overload the network by sending a massive number of queries to peers, make the portions of the network inoperable. Since in P2P network the peers should contribute in resource distribution process, peer's data stream may be compromised by fellow peers who assist in transmitting the data in the system, and sometimes free to freeload off other peers.

As mentioned in section 2.1, Cloud Computing supports four deployment models which are public clouds, private clouds, community clouds and hybrid clouds. Kumar et al. [52] believe that in cloud computing, deployment models can affect trust. In a private cloud, trust is not applicable if the third party is not involved. However, public clouds can introduce many security risks since controlling data in this deployment model is very challenging. Trust in community cloud depends on the role of third party. If there is a third party involved, the trust risks are the same as corresponding case in private cloud. Otherwise, if the community cloud is managed by the organizations in the community, trust risks are limited to the trust relationships that are discussed and agreed between community members. In Hybrid clouds, since both the private clouds and public clouds are involved, all the trust issues related to public cloud shift to hybrid cloud, too.

2.2.2. Trust Mechanisms

Trust models are the techniques that are used for evaluating trust in cloud services. They can be categorized in certain categories named trust mechanisms [2]. This paper has categorized trust models into five common trust mechanisms based on [2, 55-60].

- **Reputation Based**

Trust and reputation are different from each other. Trust is the subjective expectation of one entity about another within a specific context at a given time [3-4, 61]. Reputation, on the other hand, is what is believed about an entity's standing by the community [61]. This belief can be derived from direct or indirect experiences collected in previous interactions between entities. It is important to note that trust can be used to determine the reputation of an entity, and vice versa [62].

Trust is consider between two entities however, reputation of an entity is the aggregated opinion of a community towards that entity. In other word, an entity that has high reputation is trusted by many entities in that community. An entity can use reputation to calculate the trust level of the trustee.

In cloud computing reputation is very important since it will impact cloud users. Therefore, CSPs are trying to achieve higher reputation. Reputation is shown by a comprehensive score that is based on the overall opinion and score for the major aspects of performance.

At the first time that a user want to choose a cloud service reputation of the CSP who offer the service is very important but it is not important afterwards since performance and reliability of the service can establish trust between user and CSP.

This category of trust models contains those trust models that collect the feedback and opinions from other CCs to evaluate trust on cloud services. The trust model selects the most reliable and trusted CSP by evaluating the CCs' feedback.

- **Authentication Based**

Encryption and Key Management are important technologies that can help secure applications and data in cloud. PKI is a technology that introduces a trust mechanism to support digital signature, key certification and validation, attribute certification and validation.

It is supposed that Alice has a digital document supposedly signed by Bob using his private key K_b' and to validate, she needs Bob's public key K_b. K_1 is Alice trust public key and CA_1 is her certification authority. Alice uses CA_1's public key K_1 to validate CA_2's public key K_2; because Alice trusts CA_1 on public key certification, and CA_2's public key is certified by CA_1, Alice can believe that CA_2's public key is K_2; then Alice uses K_2 to validate CA_3' public key K_3; and finally uses K_3 to validate Bob's public key K_b. In order to become sure that Bob's key is K_b, Alice should trust CA_3This trust can come from recommendations along the chain of certificates by those certificate issuers or comes from compliance with certain certificate policies [63].

A public key certificate also contains a certificate policy (CP) extension. The certificate means that the issuing CA who conforms to the specified CP asserts that the subject CA has the certified public key, and the subject CA also adheres to the specified CP. As a result, to infer Alice's belief in CA3's key and Bob's key, she must trust that CP in the sense that any CA conforming to that CP will generate valid public key certificates [63].

Since PKI is currently practiced, trust in a certification authority (CA) with respect to issuing and maintaining valid public key certificates is based on the CA's conformance with certain certificate policies. Certificate policies play a central role in PKI trust.

This category includes trust models that use certificates (issued by standardized bodies), trust tickets, private and public keys, TPM endorsement keys issued by trusted third party or certificate authority (CA) to ensure the integrity, availability and confidentiality of data on Cloud and evaluates the confidence of CCs regarding the expected behavior of cloud services.

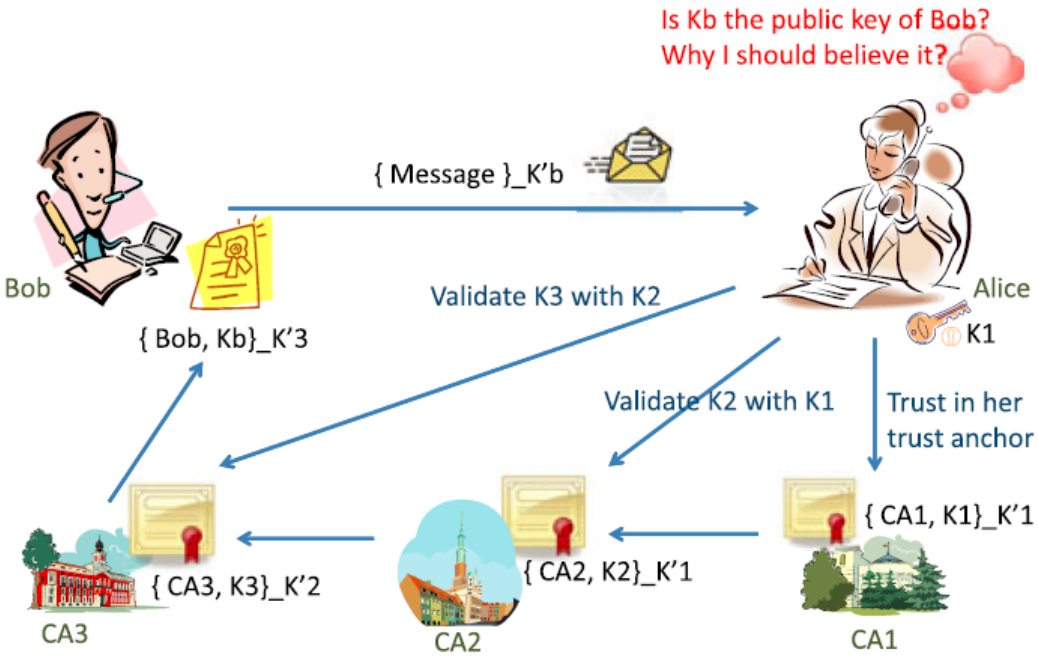

Figure 2.1. Trust relations in public key validation and certificate [63]

- **SLA Based**

A service level agreement (SLA) is a legal contract between a cloud user and a CSP. SLA is a service lever agreement. It is one of the approaches to establish trust on CSPs. The entities that are providing services are required to follow standardized SLA, e.g., proposed by Cloud Computing Use Cases community [64]. SLA validation [65] and monitoring [66] schemes are used to verify the quality of CSPs and CCs are responsible for monitoring SLA violations. Since SLA compensation clauses are developed by the CSPs, CCs do not have enough chance to apply for compensation if SLA violation happens and this is a problem due to lack of standardized SLAs for

the stakeholders in cloud computing market. However industry driven initiative [67] have addressed this problem but still it is not fully implemented.

There are a number of other issues with SLA based trust. First, SLA focuses on the "visible" elements of cloud service performance, and does not address "invisible" elements such as security and privacy. Second, many cloud users do not have enough capability to perform SLA verification on their own and they need a professional third party help to provide these services. In a private cloud the trusted broker or trust authority who is trusted in the trust domain of the private cloud can provide the users the services of SLA verification. In a hybrid cloud, a user within a private cloud might still rely on the private cloud trust authority to conduct and SLA verification; however, in a public cloud, individual users and some small organizations without technical capability may use a commercial professional cloud entity as trust broker.

Trust establishment under this category is based on contracts and agreements signed by CSPs for the delivery of different services to CCs. SLA provides the basis for trust establishment. Various security concerns and quality of service attributes are included in contracts and agreements to establish trust on CSP [15].

- **Domain Based**

Basic idea in domain based trust model is to divide the Cloud into number of autonomous domains and distinguish two types of within-domain and inter-domain trust relationships respectively. Within-domain trust values depend upon the transactions between the entities that are in the same domain. If an entity needs to compute the trust value for some other entity, it checks the direct trust table but if the direct trust value is not found then it looks for the recommended trust values from other entities [68].

The inter-domain trust relationship is using the trust relationship between domains. There is an authentication mechanism for each domain which trusts the authentication mechanisms of other domains. If an entity is authenticated by one domain, then its authentication is acceptable by all other domains.

- **Platform Based**

Platform based trust models consists of policies that ensure applications are executing on platforms that meet a specified trust assurance level and evaluate the confidence of CCs on using cloud services lunch on a specific platform. Therefore, by using this trust model, CCs can trust a CSP to use the offered platform [68].

CHAPTER 3. TRUST MODELS STUDY METHODOLOGY

CC's trust in cloud computing systems can vary based on the scope and context of applications in cloud computing. For example, CCs who are using data storage applications for storing their sensitive information on the cloud, have different requirements than those who use cloud for online gaming service. CSPs should offer a secure and controllable environment for those CCs who use data storage applications to get CCs' trust, while, for those who use gaming services, CSPs should just offer a high performance environment. Therefore, there are different trust models available for evaluating the trustworthiness of cloud services and CCs can choose one based on the service they want to use. Therefore, it becomes difficult to select a trust model that best satisfies the user's requirements. There is a need for assessment criteria that can evaluate the trust models and helps the users in selection of most suitable model in line with their preferences. In this chapter, in order to evaluate the goodness of different solutions, fourteen assessment criteria [14-19, 81] have been selected to address different aspects of identified challenges for trust models. Figure 3.1 shows how the selected assessment criteria being categorized by the risks and threats listed in section 2.1.3. These assessment criteria help to evaluate the goodness of the trust models in evaluating the trustworthiness of cloud services and also help to define which aspect of trust is evaluated and which cloud computing risks and threats is addressed by considering each of these assessment criteria.

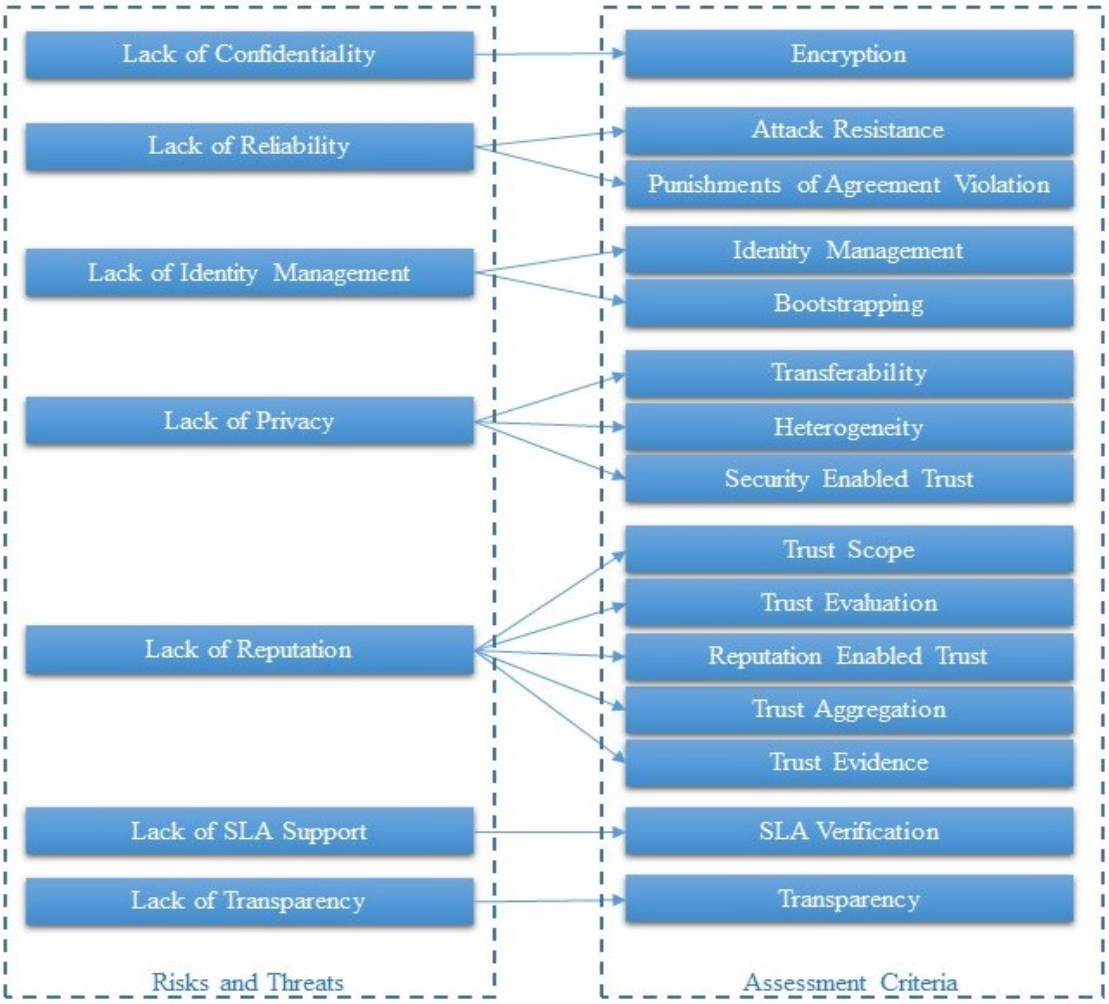

Figure 3.1. Assessment criteria and cloud computing risks and threats

3.1. Assessment Criteria for Evaluating Lack of Confidentiality in Cloud Computing

As explained in section 2.1.3, confidentiality is one of the challenges for trust models in cloud computing. The CCs who need a confidential cloud service, should choose the trust models that evaluates the following trust feature in trust management systems.

- **Encryption**

Based on [69], encryption is one strategy that CSPs use to protect enterprise cloud data from cybercriminals and any unauthorized access. Cloud Data Encryption mathematically transforms data so that it is undecipherable without the "key" that can be used to change the data back to its original form.

3.2. Assessment Criteria for Evaluating Lack of Reliability in Cloud Computing

As explained in section 2.1.3, reliability is one of the challenges for trust models in cloud computing. These assessment criteria help those CCs for whom reliability is very important and need reliable cloud services. They should choose the trust models that evaluate the following assessment criteria in the cloud system.

- **Attack Resistance**

As soon as the influence of trust and reputation models on the decision of CCs will grow, the interests in manipulating those values in Cloud environment will grow accordingly, as already seen in other service environments earlier [69]. A number of different attacks (e.g., playbooks, proliferation attacks, reputation lag attacks, false praise or accusation (collusion), whitewashing (reentry), Sybil attacks, etc.) against trust and reputation systems have been discussed [69, 70]. These types of attacks will also be of concern when designing trust and reputation system for Cloud computing environments. Thus, attack resiliency is a central design goal for developers of these kind of systems. This is one of the Quality of Service (QoS) parameters that is defined in [69] to evaluate the trust models.

- **Punishment of Agreement Violation**

Punishment is one of the approaches to deal with the violation of security policies. If violation cannot be prevented in a system, then they need to be detected and punishment protocols can be meting out to the violator. Punishment can be done automatically such as destruction of anonymity or it can be done by a mediator which can be an entity that is already part of the system, or can be someone external to the system. Punishment is been used in [68] to evaluate the trust models.

3.3. Assessment Criteria for Evaluating Lack of Identity Management in Cloud Computing

As explained in section 2.1.3 identity management is one of the challenges for trust models in cloud computing. The following assessment criteria help evaluating the trustworthiness of cloud services base on identity management criteria.

- **Identity Management**

CSA [71], explains one of the important concepts in cloud computing is identity management and authentication. Authentication is the process of validating or confirming that access credentials provided by a user are valid. A user in this case could be a person, another application, or a service; all should be required to authenticate. Many enterprise applications require that users authenticate before allowing access. Authorization, the process of granting access to requested resources, is pointless without suitable authentication [71]. When organizations begin to utilize applications in the cloud, authenticating users in a trustworthy and manageable manner becomes an additional challenge. Organizations must address authentication-related challenges such as credential management, strong authentication, delegated authentication, and trust across all types of cloud delivery models (SPI). In order to grant safe access to sensitive information and resources to all those who need it, organizations must carefully monitor which users are accessing what resources to ensure that they are accessing the resources that they need in an appropriate manner. Because of this, it is predicting that identity and access management in the cloud will be one of the top three most sought after services moving forward for cloud-based models [71]. This is one of the parameters that is defined in [2] to evaluate the trust models.

- **Bootstrapping**

A bootstrap [72- 75] is the process of starting up a system. In cloud computing trust management bootstrapping is the initial value that is assigned to trustee by trustor. During the

bootstrapping process the trustor will send out an authentication key to the trustee and the trustee can authorize the trustor. As an example, Google allows its cloud users to use SSH to connect to a Google compute engine virtual machine instance from within the Google Cloud Platform Console. However, this method is an alternative to other methods of connecting to an instance in Google cloud platform console [76]. This is one of the parameters that is defined in [2] to evaluate the trust models.

3.4. Assessment Criteria for Evaluating Lack of Privacy in Cloud Computing

As explained in section 2.1.3, privacy is one of the challenges for trust models in cloud computing. The following assessment criteria help evaluating the trustworthiness of cloud services base on privacy criteria.

- **Transferability**

CC trust in a CSP depends on the specific application context or the scope of interaction. Transfer of trust across those contexts is a significant challenge for trust and reputation systems. Consider, for example, a service provider offering an email service and a video rendering service – both belonging to the SaaS category. Both application contexts require different competencies, for example spam protection and storage for the email context, whereas for video rendering context, latency, bandwidth and assessment criteria dealing with performance matters (e.g., response time, Content Delivery Node (CDN) facilities, etc.) are important. Here, transferring trust established in one context (email) to the other one (video rendering) is not a trivial task, and could, for instance, be supported by combining the outside-in and the inside-out evaluation [81]. This is one of the QoS parameters that is defined in [69] to evaluate the trust models.

- **Heterogeneity Support**

Heterogeneity in cloud computing refers to structural differences of the devices and it comes in different forms. In cloud computing high level of homogeneity is required for resource virtualization which means the same infrastructure can be used to support different tenants with different protection and system requirements. This can led to trust problems since heterogeneous cloud infrastructures make it difficult to have effective controls to check privacy compliance in an automated way and the end-user has no means to verify that his/her privacy requirements are being fulfilled. This is one of the parameters that is defined in [2] to evaluate the trust models.

- **Security Enabled Trust**

Security Enabled Trust (Hard trust [77]) which is defined as trust that is derived from concrete security mechanisms such as validation of properties through certificates [78] and involves aspects like authenticity, encryption, and security in transactions[79]. This is one of the QoS parameters that is defined in [69] to evaluate the trust models.

3.5. Assessment Criteria for Evaluating Lack of Reputation

As explained in section 2.1.3, trust is one of the challenges for trust models in cloud computing. The following assessment criteria help evaluating the trustworthiness of cloud services base on reputation criteria.

- **Trust Evaluation**

For complex distributed environments like cloud computing, trust can be evaluated based on the following approaches: Black box approach in which the trustworthiness of an entity or a service is evaluated taking into account only the observed output such as feedback, Inside-out approach in which the trustworthiness of an entity or a service is derived based on the knowledge about the architecture of the service and the trustworthiness of its components or subsystems,

Outside-in approach which requires knowledge about the internal architecture of a service and its components as input as well as information stating the observed behavior of the overall service. The goal of this kind of model is to derive the trustworthiness of internal components of a service composition based on its external behavior [80]. This is one of the QoS parameters that is defined in [69] to evaluate the trust models.

- **Trust Scope**

In global reputation systems, the reputation of a service provider is based on the opinions from the general population, which is public and visible to all the system members, while in local reputation systems, the reputation of a CSP is built on the opinions from a group of particular people. It is much harder and more complicated to design a global reputation mechanism in a decentralized system than in a centralized system [81]. This is one of the QoS parameters that is defined in [69] to evaluate the trust models.

- **Reputation Enabled Trust**

Reputation Enabled Trust (Soft trust [77]) is defined as trust that is derived from past experiences and behavior associated with an entity. An example of soft trust is reputation, which is a component of online trust that is perhaps a company's most valuable asset [82]. This is one of the QoS parameters that is defined in [69] to evaluate the trust models.

- **Trust Aggregation**

There are two different fundamental approaches to store and aggregate trust-related information [81]. In a centralized trust management system a single globally trusted server will take all the responsibilities of managing reputations for all the members. In decentralized system, there is no central server and the members in the system have to cooperate and share the responsibilities to manage reputation in the whole system. Trust models that use centralized

architectures are prone to scalability and security issues. A centralized system relies on the assumption that the system participants completely trust the centralized authority which in turn must be correct and always available. If the centralized authority is not carefully designed, it can become a single point of failure for the entire system. This is one of the QoS parameters that is defined in [69] to evaluate the trust models.

- **Trust Evidence**

Trust evidence is a factor of trust which includes direct interactions and indirect interactions between trustor and trustee. Direct trust means trust that is obtained by entities' direct interaction. Indirect trust or recommended trust means trust that is obtained from credible third party who has direct contact with the designated one [83]. Some system are using both direct and indirect interactions while others use either only indirect or direct interactions. This is one of the QoS parameters that is defined in [69] to evaluate the trust models.

3.6. Assessment Criteria for Evaluating Lack of SLA Support in Cloud Computing

As explained in section 2.1.3, SLA support is one of the challenges for trust models in cloud computing. The following assessment criteria help evaluating the trustworthiness of cloud services base on SLA support criteria.

- **SLA verification**

As mentioned in earlier, SLA is a legal contract between a cloud user and a CSP and is one of the approaches to establish trust on CSPs. This assessment criteria evaluates the trust models to see if they support SLA. This is one of the QoS parameters that is defined in [69] to evaluate the trust models.

3.7. Assessment Criteria for Evaluating Lack of Transparency

As explained in section 2.1.3, transparency is one of the challenges for trust models in cloud computing. The following assessment criteria help evaluating the trustworthiness of cloud services base on transparency criteria.

- **Transparency**

The derived trust values or reputation scores must be transparent to and comprehensible enough for the consumers, so that they can easily and confidently make trust-based decision. To make the trust values transparent and comprehensible, users need to be supplied with an intuitive representation of trust together with enough information regarding the relevant assessment criteria. This is one of the QoS parameters that is defined in [69] to evaluate the trust models.

CHAPTER 4. TRUST MODELS IN CLOUD COMPUTING

As explained in chapter 2, this work has categorized trust models into five categories named trust mechanisms which are reputation based trust models, authentication based trust models, SLA based trust models, domain based trust models and platform based trust models. This chapter studies selected list of recent and popular trust models in each category and analyzes them based on the research methodology explained in chapter 3.

4.1. Reputation Based Trusts

As discussed in section 2.2.2, in reputation based trust models, an entity's reputation is usually evaluated based on feedback from those who have direct interactions with the entity. Therefore, this category includes the trust models that collect CC's feedback to evaluate trust on cloud services. In this section in order to evaluate reputation based trust models some of the recent trust models have been studied and analyzed based on the assessment criteria discussed in chapter 3.

- **Wanga et al. (2014) [84]**

The authors proposed a trust model named Accurate and Multi-faceted Reputation (AMRep) [84] trust model which introduces a couple of malicious rating detection approaches to improve the accuracy of reputation calculation. AMRep employs multiple rating indexes. By processing all index ratings in an inter-dependent way, AMRep builds an accurate index reputation calculation model, which can effectively identify malicious users and improve the accuracy of the reputation calculation. AMRep also designs a multi-faceted reputation evaluation method, which combines relevant index reputation values to deduce the attribute reputation values of a cloud service at various granularity [84].

The reputation value in this model is calculated as follow [84]:

$$CAR_l^{t+1} = \frac{\sum_{i=1}^{z} CIR_{li}^{t+1} \times w_{li}}{\sum_{k=1}^{z} w_{lk}} \quad (4.1)$$

Where CAR_l^{t+1} is the multi-faced reputation value of attribute A_l after t + 1 evaluation cycles. CIR_{li}^{t+1} is the the cumulative reputation value after t+1 evaluation cycle. w_{ij} is the corresponding weight vector elements.

- **Tan et al. (2014) [85]**

The authors proposed a trust model named Trust Work Flow Scheduling (TWFS) which is a trust model for workflow applications in cloud computing. This model proposes a trust service-oriented workflow scheduling algorithm. The scheduling algorithm adopts a trust metric that combines direct trust and recommendation trust.

The trust value in TWFS is calculated as follow [85]:

$$Tr(S_i) = \left(1 - \frac{1}{e^k}\right) * \left(\frac{n_i + 1}{N_i + 2}\right) + \left(1 - \left(1 - \frac{1}{e^k}\right)\right) * ((\frac{1}{|S_i|} * \sum_{j \in S_i} v_{ij}) + \frac{\sum_{i=1}^{n} w_{ai}(v_{ij} - \bar{v}_i)}{\sum_{i=1}^{n} |w_{ai}|} \quad (4.2)$$

where k is the number of times that the ith service is used by the service client. n_i, denoting the number of successful interactions, and N_i, denoting the total number of interactions for the ith service. S_i be the set of services rated by user i, and v_{ij} be the rating given by user i to the jth service. Next, let average $\bar{v}_i$ be the average rating by user i. The weight w_{ai} reflects the similarity between user a and user i. The weights for the users are calculated by the Pearson correlation coefficient (PCC) which is widely used to compute the degree of similar relationship between two variables. The algorithm for implementing TWFS proposed by WenAn Tan et al. is as follow: The Input of the algorithm is request processing time, cost, and trust value and the output is a workflow schedule strategy for Enterprise Information Systems [85].

- **Muchahari et al (2012) [86]**

The authors proposed a trust model named Cloud Service Registry and Discovery Trust (CSRD) [86] trust model which is a trust model that serves as CSPs' registry and lists their respective trust values calculated from Trust Calculator (TC). CSRD is a registry that enables CSP's to list themselves along with their respective trust values. CSPs need to make sure that they register themselves in their specific category based on three service delivery models namely SaaS, PaaS and IaaS. The Dynamic Trust Monitor (DTM) keeps watch on the deviating trust values with time and transactions dynamically as shown in Figure. 4.1 [86].

CSRD can help CSCs to opt for the right CSPs according to their need based on brand, usefulness of service, user experience and background. CSPs in order to register need to enter all their details relating to type of services they provide, since when providing services, specialty of their services etc. Other details like user experience, background etc., of any registered CSP can be depicted from TC, a transparent module of calculating trust. DTM looks after the changing trust values of CSPs that keeps on changing for the dynamic nature of cloud environment [86].

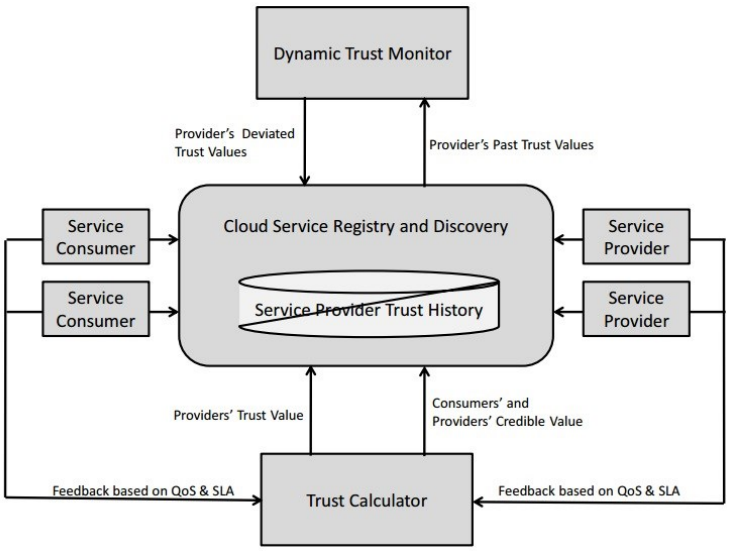

Figure 4.1. Cloud service registry and discovery with trust calculator [86]

Trust Calculator (TC) is where the trust values of CSRD registered CSPs are calculated. The calculated trust values are then displayed in CSRD against the respective registered CSPs. Trust factor can be calculated from the following formula [119]:

$$T(p_i) = \frac{T_{prev}(p_i) + T_{prov}(p_i) + T_{cons}(p_i)}{3} \tag{4.3}$$

where $T_{prev}(p_i)$ is the previous trust value of provider p_i from CSRD, $T_{prov}(p_i)$ is the average feedback value of p_i assigned by other credible providers. $T_{cons}(p_i)$ is the average feedback value of p_i assigned by other credible CSC:

$$T_{prov}(p_i) = \frac{\sum_{j=1, j\neq i}^{n} fb(P_i, P_j)}{n} \tag{4.4}$$

where $fb(P_i, P_j)$ is the feedback value of provider P_i assigned by provider P_j. fb is the average of total of ranges which is between 0 to 5, given to each questions in the feedback.

$$T_{cons}(p_i) = \frac{\sum_{j=1, j\neq i}^{n} fb(P_i, C_j)}{n} \tag{4.5}$$

where $fb(P_i, C_j)$ is the feedback value of provider P_i assigned by CSC C_j and n is the total number of CSCs.

- **Raghebi et al. (2013) [87]**

The authors proposed an adaptive method that helps distinguish between malicious and reliable CC feedbacks. Figure 4.2 shows the method architecture. As we can see from the figure, the trust broker is responsible for evaluating the dependability of any cloud service. The CC feedback reliability is based on the combination of the similarity between CC feedbacks for shared services and closeness to majority consensus of feedbacks for other services. The trust broker acts as a third party to use the proposed trust evaluation method which analyzes the feedbacks and

calculates the trust value. This method use both direct and indirect trust. In indirect trust a CC 'A' wants to evaluate the trust of a CSP 'C' whom it has never interacted with before. Besides, 'A' knows another customer 'B' who has interacted directly with the target CSP. In this method, we focus on the reliability of the feedback of the middle customer 'B'. In the first step, it evaluates the reliability of a middle customer 'B' by comparing its feedbacks for commonly rated services. Each customer feedback is weighted based on a newly introduced similarity measure. In the second step, it considers all the feedbacks that customer 'B' has rated and compare them with the majority of feedbacks. This trust evaluation method is defined in section A and B [87].

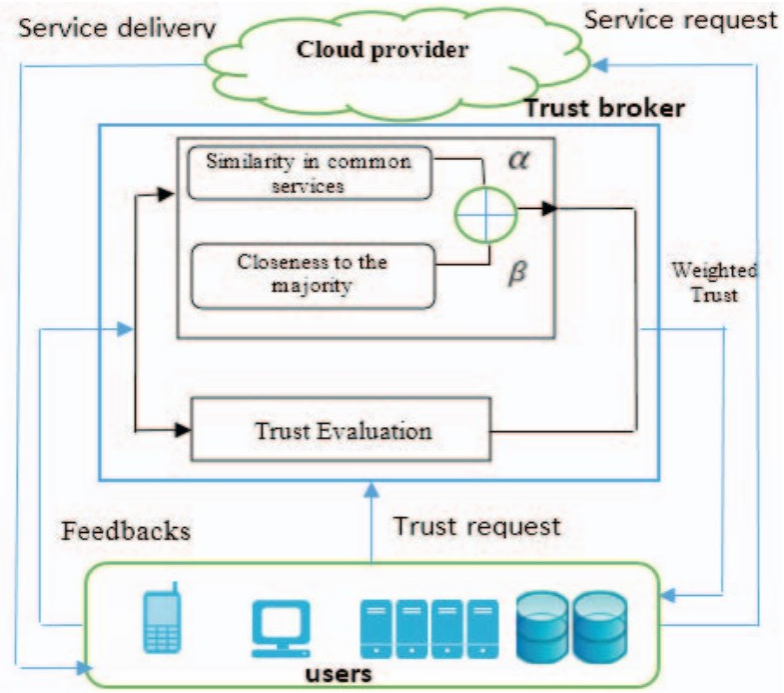

Figure 4.2. The overview of a trust method in a cloud architecture [87]

- **Fan et al. (2013) [88]**

The authors proposed a trust model named Recommendation Based Trust Management Mechanism (RBTM) [88] which is a trust management framework for cloud computing environments, and then introduced an effective reliability-based filtering mechanism to ensure the reliability of trust feedback for cloud computing services. The filtering mechanism uses two

important factors, namely, familiarity and consistency, to filter out unreliable trust feedback. As shown in Figure 4.3, this system is divided into two domains: the service provider domain and the service user domain. In the service provider domain, the system mainly performs the function of connecting a user to the portals of different CSPs. In the service user domain, it mainly performs the function of collecting cloud service trust feedback from distributed service users, searching reliable users, deriving the real time trust results on the cloud services. This system provides a publishing platform of service and trust management information. It also selects, filters, judges and aggregates the information so as to connect service users and providers with a view to enhancing trust relation. Service providers and users should subscribe to the system through their respective entries for service provider domain and user domain in order to obtain a provider ID and user ID respectively. Service providers should provide their provider ID, property (individual or organization etc.), scale (small, mediate, or large), and other information. Also, for each service, the service provider should provide the SLA, which should include performance guarantees, QoS and cost, the service functions, and basic guarantees on nonfunctional properties of the services [88].

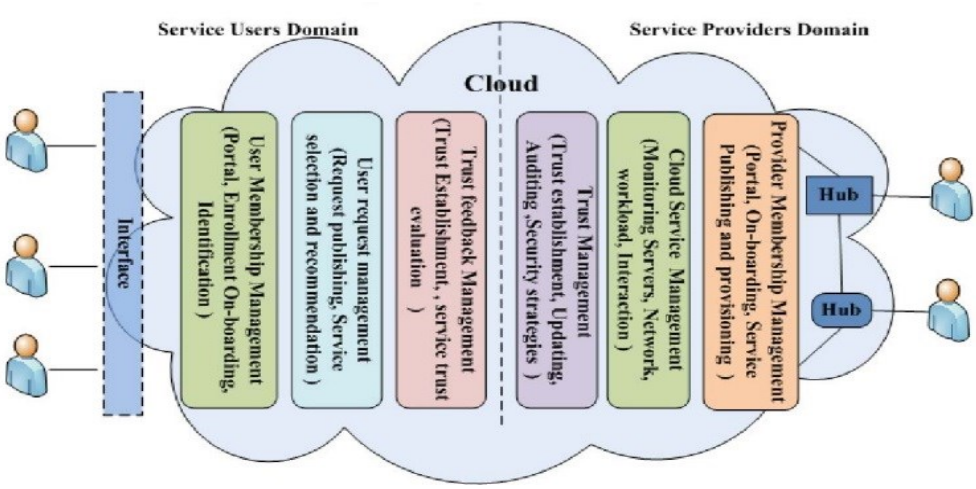

Figure 4.3. The trust management system for cloud services [88]

Users will deem a service provider as trustworthy if its services satisfy the advertised SLA and other specifications. For users, the system will also create a table of their basic information after they subscribe to the system, where they can publish their required information about services. After using a service, they are asked to input their trust feedback for the service. This method uses the "general trustworthiness" trust feedback given by five grades, i.e., 1 to 5, with 1 being the most untrustworthy, and 5 the most trustworthy [88].

- **Noor et al. (2014) [89]**

The authors proposed a trust model named CloudArmor [89], which is a reputation-based trust management framework that provides a set of functionalities to deliver Trust as a Service (TaaS), which includes a novel protocol to prove the credibility of trust feedbacks and preserve users' privacy, an adaptive and robust credibility model for measuring the credibility of trust feedbacks to protect cloud services from malicious users and to compare the trustworthiness of cloud services, and an availability model to manage the availability of the decentralized implementation of the trust management service [89].

The CloudArmor framework is based on the service oriented architecture (SOA), which delivers trust as a service. Figure 4.4 depicts the framework, which consists of three different layers. The CSP Layer consists of different CSPs who offer one or several cloud services, i.e., IaaS (Infrastructure as a Service), PaaS (Platform as a Service), and SaaS (Software as a Service), publicly on the Web. Interactions for this layer are considered as cloud service interaction with users and TMS, and cloud services advertisements where providers are able to advertise their services on the Web. The Trust Management Service Layer consists of several distributed TMS nodes which are hosted in multiple cloud environments in different geographical areas. These TMS nodes expose interfaces so that users can give their feedback or inquire the trust results in a

decentralized way. Interactions for this layer include: i) cloud service interaction with CSPs, ii) service advertisement to advertise the trust as a service to users through the Internet, iii) cloud service discovery through the Internet to allow users to assess the trust of new cloud services, and iv) Zero-Knowledge Credibility Proof Protocol (ZKC2P) interactions enabling TMS to prove the credibility of a particular consumer's feedback. The Cloud Service Consumer Layer consists of different users who use cloud services. Interactions for this layer include: service discovery where users are able to discover new cloud services and other services through the Internet, trust and service interactions where users are able to give their feedback or retrieve the trust results of a particular cloud service, and registration where users establish their identity through registering their credentials in IdM before using TMS [89].

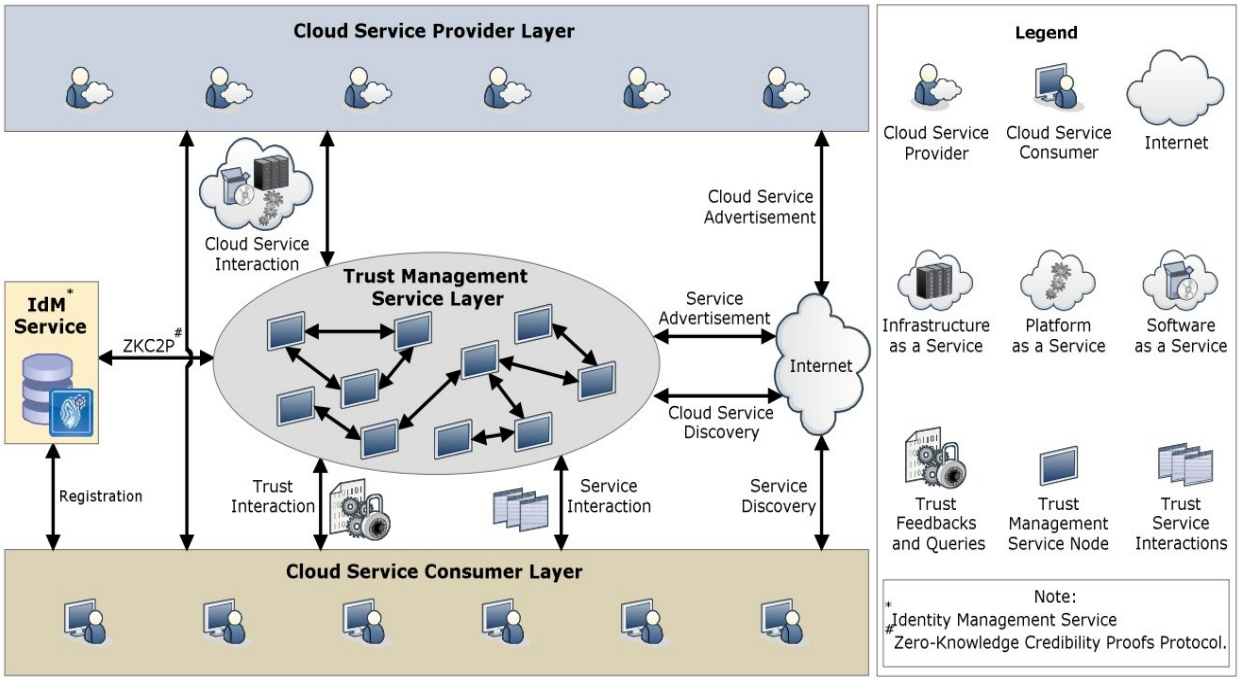

Figure 4.4. Architecture of the CloudArmor trust management framework [89]

- **Li et al. (2015) [90]**

The authors proposed a trust model named Data-driven and Feedback-Enhanced Trust (DFET) [90] trust model which is an enhanced and hierarchical feedback mechanism that can

effectively reduce networking risk while improving system dependability. Theoretical analysis shows that Data-driven and Feedback-Enhanced Trust (DFET) pattern is highly dependable against garnished and bad-mouthing attacks. They also build a prototype system to verify the feasibility of DFET pattern and the experiments yield meaningful observations that can facilitate the effective utilization of DFET in the large-scale multi-cloud collaborative environment. Figure 4.5 shows the architecture of this model. Unified Access Portal (UAP) includes the unified cloud management and unified cloud service portals. Cloud users open a unified cloud service portal and select a trusted service catalog when they would like to use providers. An administrator manages virtual servers on the unified cloud management portal. The trust-aware service monitor is located between users and cloud sites, providing a unified cloud service portal and a unified cloud management portal for a server user and a server administrator, respectively [90].

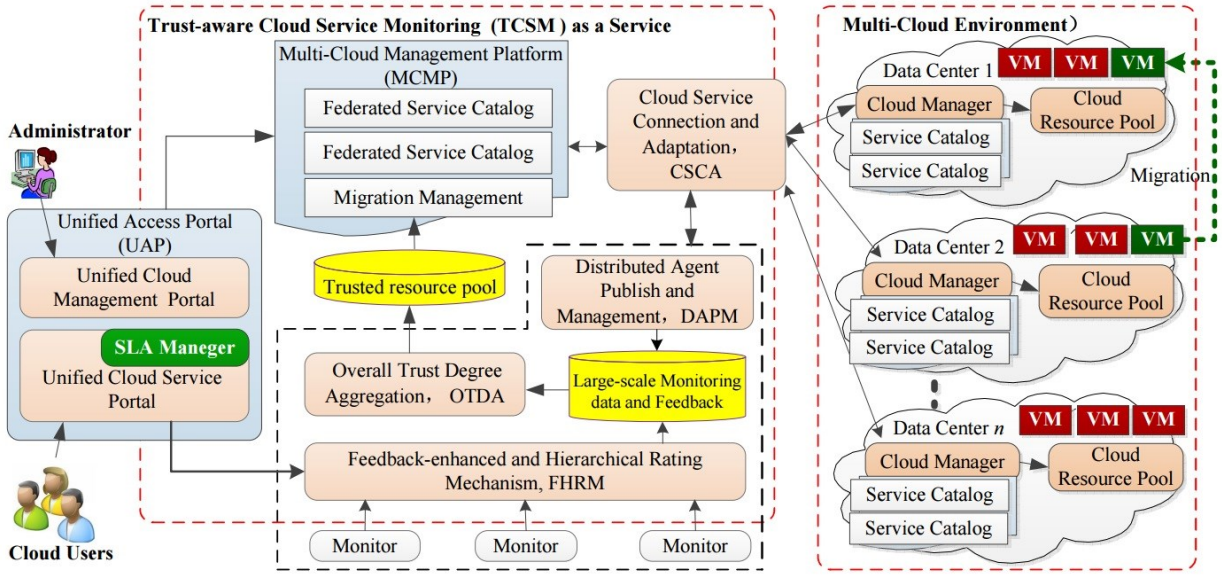

Figure 4.5. Trust-aware cloud service monitoring service based on large-scale monitoring data and feedback enhanced rating aggregation mechanism [90]

The unified cloud service portal creates virtual server templates, which are enrolled in advance as service catalogs into a Multiple Cloud Management Platform (MCMP). The MCMP module stores all available and trusted services from which it can automatically select highly

trusted services to meet user's requirements. The MCMP creates a service catalog that links with a highly trusted resource and then provides this catalog as a trusted resource for the user through the unified cloud service portal. Moreover, the MCMP provides a specific management service called migration service. In applications, a user often has to migrate one application from one cloud site to another more trusted one [90].

The Cloud Service Connection and Adaptation (CSCA) encapsulates different application programming interfaces (APIs) provided by different CSPs or cloud sites, such that other modules of the trust-aware service monitoring architecture only need to be aware of a single set of APIs. Distributed Agent Publish and Management (DAPM) monitors the real-time service data of allocated resources to guarantee SLA with users. The Overall Trust Degree Aggregation (OTDA) module is not only the core of the trust-aware cloud computing system, but is also a major focus of this paper. In Feedback-enhanced and Hierarchical Rating Mechanism (FHRM), the high-level feedback organization can be defined as other monitors, that is, the whole multiple-cloud market has more than one monitor (i.e., TCSM). Thus, multiple cloud monitors form a hierarchical feedback system [90].

- **Rizvi et al. (2014) [91]**

The authors developed a trust model named centralize trust model involving CSPs and CSUs as well as one or more third-party auditing body to determine a fair score for each service provider, using the third party assessment results and the feedback received from each CSU [91]. In this model, each entity needs differ when it comes to what it wants from the CSPs. Some entities may require more privacy than scalability. Individual entities can then have baseline measurements that they can use to decide which provider best suits their needs. When trust is built properly into the cloud architecture, it allows for three core qualities. The first is the incentive for responsibility

in the cloud setting. This is done by remembering the past actions of a CSP or a CSU. In other words, the trust model which evaluates entities trustworthiness based on their past behavior history provides incentives for both CSPs and CSUs to behave more responsibly for their actions in the cloud [91].

- **Bradai et al. (2013) [92]**

The authors proposed a trust model named Byzantine, Altruistic and Rational (BAR) [92] which is a trust model to detect rational and byzantine (malicious) peers in the context of hybrid cloud. In this model it is assumed that there are N peers (a finite set of local resources of the enterprise and peers allocated from the IaaS cloud computing) executing tasks for the same application. Reputation data is needed for the trust evaluation of the peers are stored in the portal. It stores the database of collected reputation. Its task is to compute the trust value for each peer and make the decision. For that, the portal has three modules. Reputation collector which is responsible for retrieving local reputation vectors. Trust manager which is responsible for calculating the global trust vector. We can notice in this module that the trust is performed in a dynamic and centralized way (in a trusted entity). Decision maker which is responsible for deciding for current and future execution. The decision is based on the final trust vector. It consists on guiding the scheduler in the application organization. It means that if the trust score is good, the scheduler will keep the peer in the execution plan. Each peer in the system has three modules and local storage on its executer manager. Evaluation engine which is responsible for a cyclic update of evaluation using the model. Execution engine which is responsible for the task of execution and result delivery to other peers. Reputation sender which is responsible for sending evaluations to the portal [92].

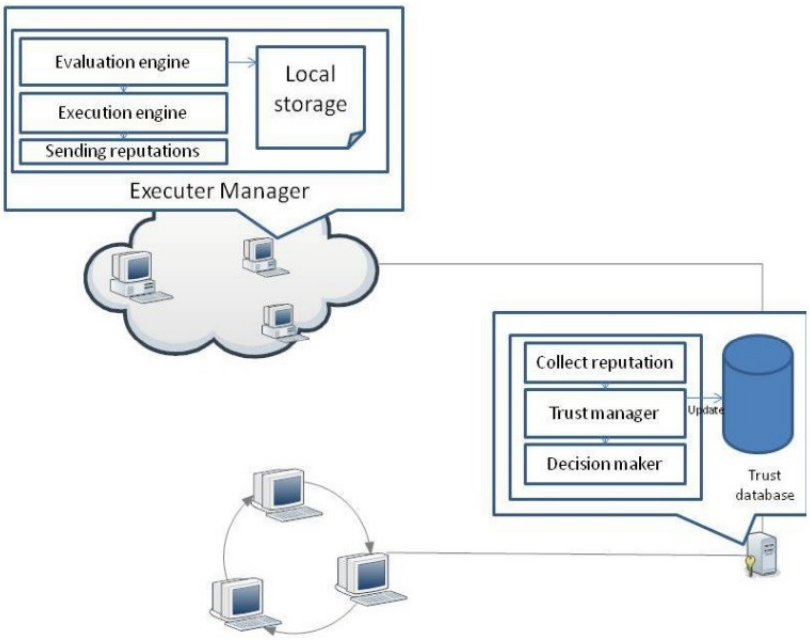

Figure 4.6. BAR architecture details [92]

- **Agheli et al. (2014) [93]**

The authors proposed a trust model named Select Service Provider Trust Model [93] which is a trust model for service provider selection. This model can help users to choose proper service providers according to their needs by calculating final trust. Figure 4.7 shows the architecture of the model. In this model Service request layer includes different service receptions which request for a special kind of service. The service requester is able to specify the interaction feedback to cloud service by calling trust management service. Services' information layer is a common unit for storage of service details and providers should be present. Trust management system is responsible for collecting and maintaining of recommendations and keeping scores. The value of service providers' trust is evaluated by this unit and is updated in accordance with the behavior of service provider and service receiver [93].

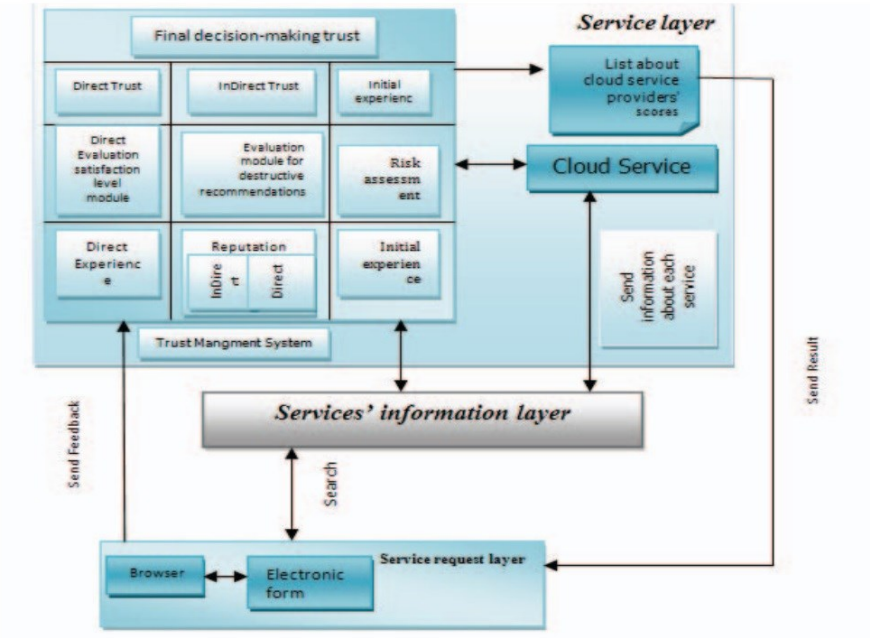

Figure 4.7. Select service provider trust model [93]

Direct experience allocates trust values according to the satisfaction level of interaction between two entities. Direct Evaluation satisfaction level module is used for evaluation of the direct interchange satisfaction level. Scores are allocated according to priority and importance of services. Indirect experience is the amount of trust collected by other trustful groups. Evaluation module for destructive recommendations recognizes feedback credit. With Initial experience: each service is categorized on the basis of initial trust security [93].

- **Zhang et al. (2010) [94]**

The authors proposed a trust model named Application-Oriented Remote Verification Trust Model (ARVTM) [94] which is a model that dynamically adjusts the users' trust value with the trust feedback mechanism to determine whether or not the requested resource or service should be provided. ARVTM uses the framework named Trust Evaluation Management based on Remote Verification (TEMRV). Figure 4.8 shows its components and the information flow among them. It is comprised of three parts: trust collection, trust assessment and trust maintenance. Trust collection consists of Credential Database (CDB) and Application Information Database (AIDB).

Trust assessment is completed by Trust Feedback Compliance Checker (TFCC). Trust maintenance contains Operation Monitor (OM), Credit Adjustment Rule Database (CARDB) and Evidence Database (EDB) [94].

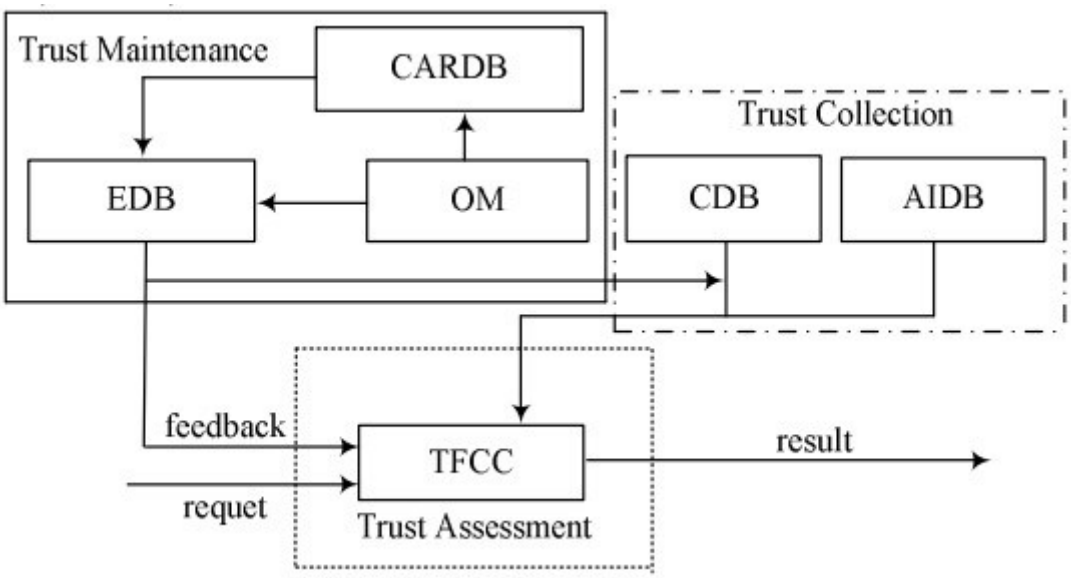

Figure 4.8. TEMRV framework [94]

The main function of trust collection is to collect and store all kind of basic information. CDB is to store various certificates that might be used in access process, including user authentication information, etc. AIDB can store the access security requirements of resources or services from the entities, as well as with their corresponding application verification contents. Trust assessment is responsible for user information verification and trust authorization verification. TFC0C uses the application-oriented remote verification mechanism. The result is determined by checking in EDB whether the user has any behaviors that violate the trust rules, and adjusting the trust value of the user based on rules of CARDB. Trust maintenance uses action monitoring to adjust the user's trust values and reflect the user's trust status. OM will monitor the user's operation actions after the authorization. Once an error, compromise or attack occurs, the information of the operation will be feedback to OM, and the abnormal operation will be recorded into EDB. Such information is essential for trustworthiness adjustment, and it also provides a

referential log to track and handle anomaly operations. CARDB stores the trustworthiness adjustment rules that are subject to different operations. OM controls the creation, modification, and deletion of the rules in CARDB [94].

- **Liu et al. (2011) [95]**

The authors proposed a trust model named Data Coloring by Cloud Watermarking [95] trust model which is using a data coloring method based on cloud watermarking to recognize and ensure mutual reputations. The flow chart of data coloring based on cloud watermarking is illustrated in Figure. 4.9 Ex is provided by data owner; En and He are produced by negotiation of data owner and service provider. Then, a lot of cloud drops will form by forward cloud generator and are used to color the user data. When the data are used, the cloud drops are extracted from colored data, and Ex, En, and He will be produced by reverse cloud generator. Final color matching will complete the confirmation. Data owner and storage service provider negotiate together to select En and He, just like the "key". Taking user print as an example, it is shown as a pixel matrix (m × n) where the grey-level value of each pixel ranges between 0 and 255. After traversing the matrix line by line from top to bottom, an image, which is Ex of a cloud watermark is obtained. In the process of data coloring, the location of the watermark to be embedded and the algorithm for embedding are decided by a user s requested security strength and allowable expense. Security strength will determine extra storage space, and algorithm complexity will decide time expense of data accessing. When users coloring data are used illegally, paint drops could be extracted from the data according to the selected embedding algorithm. Then, Ex, En, and He of these paint drops are computed by reverse cloud generator. When Ex is compared with the original user paint, the confirmation is completed. Because of the universality of data coloring and the basic certainty

hiding in the uncertainty represented by cloud model, the process can be applied even to part of user s data [95].

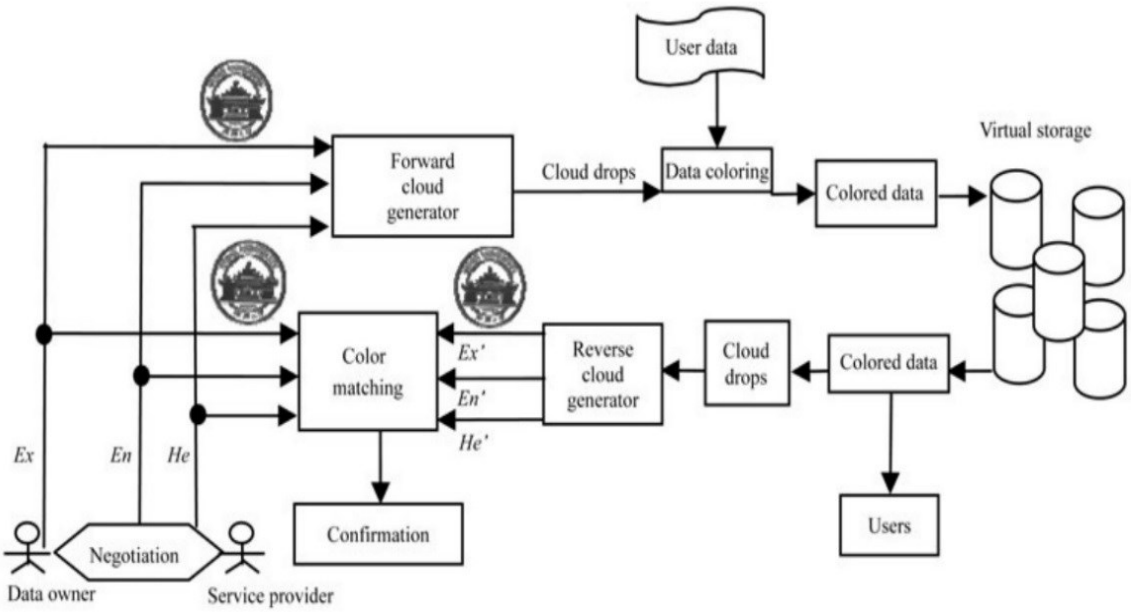

Figure 4.9. Flow chart for data coloring based on cloud watermarking [95]

- **Kong et al. (2012) [96]**

The authors proposed a trust model named Trust Based Recommendation System in Service Oriented Cloud Computing (TRSC) [96] trust model which is a particular mechanism, which evaluates CSP services based on the trust of them. In TRSC, the resulting trust value is obtained combining direct trust and recommendation trust. Direct trust of a user on a cloud service is computed as usual, that is according on the direct interaction. While the recommended trust is evaluated taking into account opinions coming from users, or other authority of the field, who are trusted by the user, considering that this kind of trust is more reliable.

- **Singh et al. (2014) [97]**

The authors proposed a trust evaluation mechanism which calculates final trust on a service provider based on a user's past experiences with service provider and based on friends and third party's recommendations.

In the proposed trust evaluation framework, trust is visualized from three different angles. One is consumer's self-trust with service provider which represents the consumer's trust on service provider based on his/her previous interactions with service provider and experiences felt as a result of those interactions. Second is friends' trust which represents the consumer's friends' trust on service provider. Friend's trust is based on that friend's previous interactions with the service provider. Third is independent feedback trust which is acquired from independent third party based the evaluation done by that third party. Figure 4.10 shows the pictorial representation of the environment envisaged for trust evaluation framework [97].

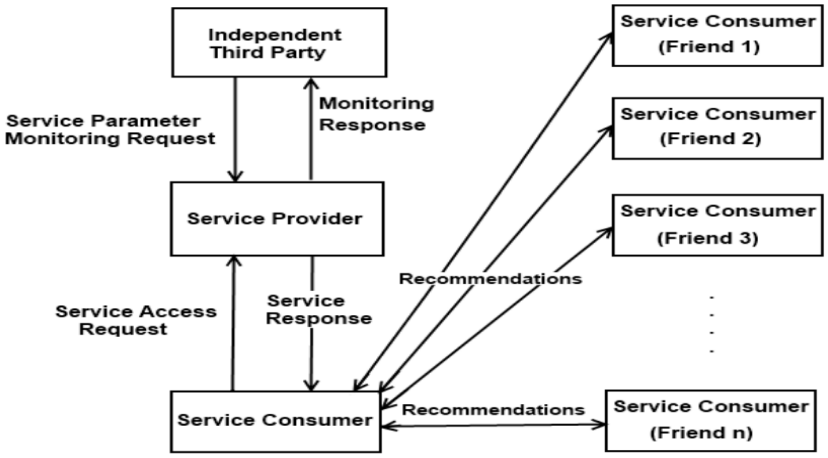

Figure 4.10. The pictorial representation of the environment envisaged for trust evaluation framework [97]

As per Figure 4.10 a service consumer accesses the services provided by a service provider. The services of service provider are monitored by an independent third party (which can be a cloud auditor) and monitoring reports are available to consumers to help them make decisions regarding the trustworthiness of a service provider. In the environment, there are many consumers and any of the consumers can be friends of other consumers as per their preferences. Friends send recommendations about service providers. Moreover, recommendations about the trustworthiness of any service provider can be asked from a friend if that friend has previously interacted with the service provider. Final trust is calculated considering the recommendations from friends,

monitoring results from independent third party and taking into account the consumer's past interactions with the service provider [97].

- **Han et al. (2009) [98]**

The authors proposed a trust model named Cloud Service Recommendation System (CSRS) [98] trust model which presents a Cloud service selection framework in the Cloud market that uses a recommender system (RS) which helps a user to select the best services from different CSP that matches user requirements. The RS recommends a service based on the network QoS and Virtual Machine (VM) platform factors of difference CPs.

The Cloud service RS (CSRS) proposed in this paper recommends the effective resources from the Cloud market using QoS and service rank analysis of resources provided by CSPs. QoS includes execution time, average execution time, response time, average response time etc. of Cloud services. Service-rank (SRank) considers the quality of virtualization hypervisors used by different Cloud service platforms, user feedback and cost of services to provide better services. The proposed RS for Cloud services will contribute to the research model for Green IT and a use to manage the resources efficiently [98].

The proposed architecture of Cloud service RS (CSRS) is shown in Figure 4.11. The main components of the CSRS include web portal, request manager, resource register, resource manager, application specific service, resource monitoring and provisional manager [98].

Web portal is used by the CPs to register their resources/services to the CSRS system. Also user can submit their requirements of Cloud services and get the recommendation results through the web portal. When a CP wants to register its resources/services to CSRS system, the request is passed to the Request.

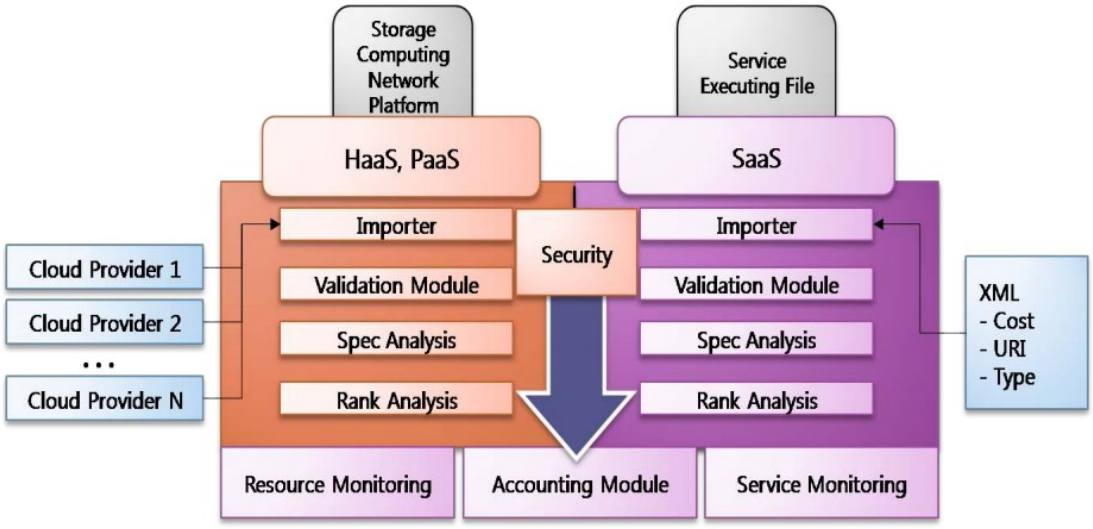

Figure 4.11. Architecture of CSRS trust model [98]

Manager through the web portal. The Request Manager evaluates each of the resource information and sends the register confirmation request to the Resource Register which calculates the S-Rank and QoS values of the CP's and stores all the information in the Resource Repository. The Request Manager also handles user requests for Cloud services and provides recommendation results through web portal. Resource Manager as a core module of the system controls the market system and manages various resources of CPs. It stores meta-data that is the logical organization of the resources distributed by the VO (Virtual Organization) in VO repository and the cost of the market and measured performance in Policy repository. It has a role to set up the virtual environment in the CPs via Provisioning Manager. It also monitors the status of the resources through the Resource Monitoring and sends the S-Rank update request to the Resource Register. Application Specific Service is used to provide the software services to user directly [98].

- **Noor et al. (2011) [99]**

The authors overviewed the design and implementation of a Trust as a Service framework. The proposed system is based on a credibility model, responsible for distinguishing between the believable and the malicious trust feedbacks, taking into account the majority consensus of

feedbacks too. In addition to the credibility model, the other salient feature of the discussed framework is that it allows trust feedback assessment and storage to be managed distributive, avoiding common drawbacks of centralized architectures.

The proposed framework uses the Service Oriented Architecture (SOA) to deliver trust as a service. SOA and Web services are one of the most important enabling technologies for cloud computing in the sense that resources are exposed in clouds as services. In particular, this framework uses Web services to span several distributed TMS nodes that expose interfaces so that trust participants (i.e., the cloud service consumers) can give their trust feedbacks or inquire about the trust results based on SOAP or REST messages. Figure 1 depicts the framework, which consists of three different layers, namely the CSP Layer, the Trust Management Service Layer, and the Cloud Service Consumer Layer. The CSP Layer consists of different CSPs who provide cloud services. The minimum indicative feature that every CSP should have is to provide the infrastructure as a service [99].

The Trust Management Service Layer. This layer consists of several distributed TMS nodes that expose interfaces so that cloud service consumers can give their trust feedbacks or inquire about the trust results represents. The Cloud Service Consumer Layer. Finally, this layer consists of different cloud service consumers who consume cloud services. For example, a new startup that has limited funding can consume cloud services. A cloud service consumer can give trust feedbacks of a particular cloud service by invoking the TMS. This framework also contains a Registry Service that has several responsibilities including. Service Advertisement: both CSPs and the TMS are able to advertise their services through the Service Registry; Service Discovery: the TMS and cloud service consumers are able to access the Service Registry to discover services [99].

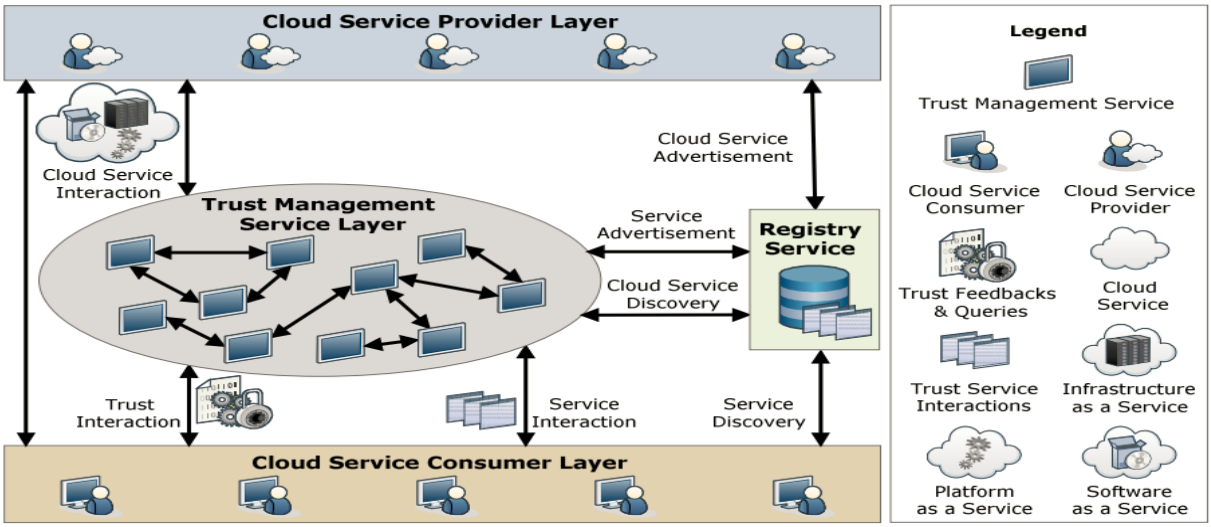

Figure 4.12. Trust management service architecture [99]

- **Fridhous et al. (2011) [100, 101]**

The authors proposed a model for trust formation and evolution based the Quality of Service of cloud nodes. Figure 4.13 shows the proposed system that is used to form, evolve and manage the trust of computing nodes in a cloud system. The trust formulation unit computes the initial trust values based on the type of service and level of service. Service monitor monitors the performance of the service provider and informs the trust evolution unit if the service was carried out satisfactorily or not. Trust evolution unit keeps track of the current trust values for different service types and evolves them based on the feedback received from the service monitor.

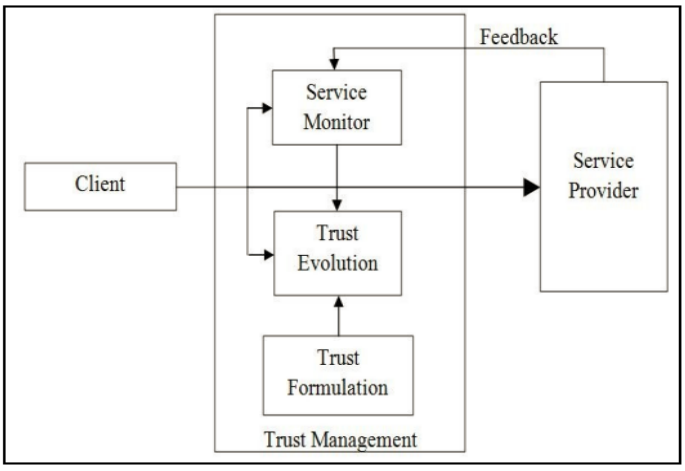

Figure 4.13. Trust evaluation system architecture [100]

- **Wang et al. (2013) [102]**

The authors proposed a trust model named Usability feedback Rating Trust Model. The proposed reputation measurement approach of Cloud services contains two phases. The first phase is Trust Vector, in which we adopt cloud model to analysis the unstability level of feedback rating. The second phase is Calculating Reputation, in which we adopt fuzzy logic to calculate the reputation score of each Cloud service. Eventually, the reputation scores are stored that is an important criterion for Cloud service systems [102].

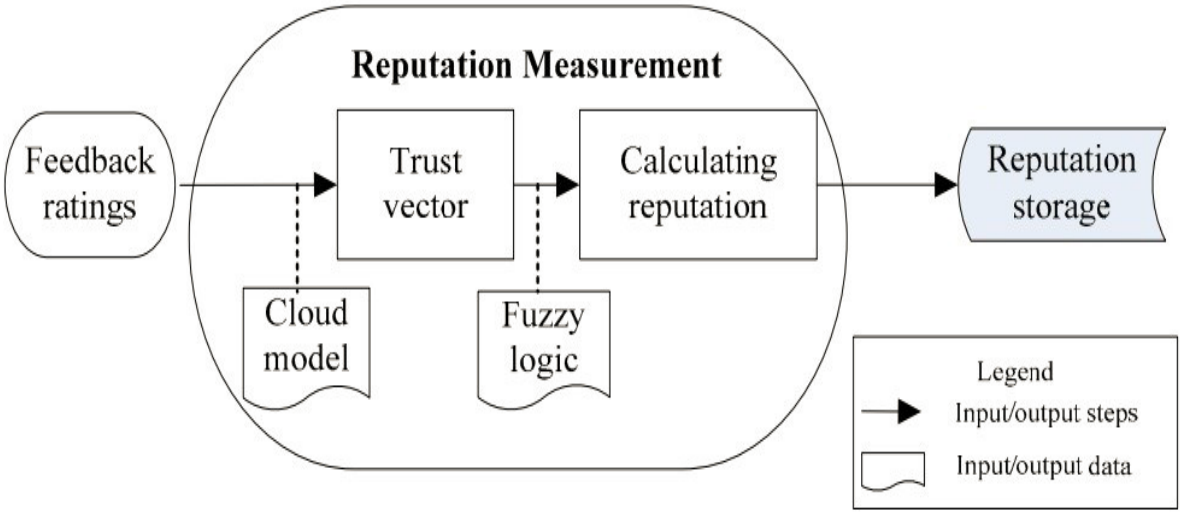

Figure 4.14. Procedures of the reputation measurement approach [102]

4.1.1. Analyzing Reputation Based Trust

Table 4.1, 4.2, and 4.3 represents the analysis of Reputation based trust management models based on the selected assessment criteria defined in chapter 3. The assessment criteria and the cloud computing challenges that each trust feature address are indicated in the tables. In the tables × means the trust model does not evaluate the specified assessment criteria and ✓ means that it evaluates the specified assessment criteria. Also, there are some assessment criteria that if the trust model evaluates them instead of having ✓ as an indicator, it is mentioned how it support that criteria such as attack resistance, trust evaluation, trust scope and trust aggregation.

Table 4.1. Evaluating Reputation Based Trust Models – Part A

Challenges	Lack of Confidentiality	Lack of Reliability	
	Encryption	Attack resistance	Punishment of Agreement Violation
AMRep	×	Individual malicious / Malicious pre-trusted peers / Collusion	×
TWFS	×	×	×
CSRD	×	Individual malicious	×
Raghebi (2013)	×	×	×
RBTM	×	Malicious feedback	×
CloudArmor	×	Man-in the-Middle (MITM) attack, Collusion, Sybil	×
DFET	×	garnished, bad-mouthing	✓
Rizvi (2014)	×	×	×
BAR	×	malicious peers	×
Agheli (2014)	×	×	×
ARVTM	×	×	×
Liu (2015)	×	malicious peers	×
TRSC	×	×	×
Singh (2014)	×	×	×
CSRS	×	×	×
TMS	×	×	×
Firdhous (2011)	×	×	×
Wang (2013)	×	malicious users	×

Based on table 4.1, none of the selected reputation based trust models evaluates encryption that address lack of confidentiality challenge. In order to find the trust models that address lack of reliability, DFET is the only model that evaluates all of the assessment criteria under this category. It evaluates punishment and attack resistant assessment criteria. On the other hand, AMRep, CSRD, RBTM, CloudArmor, BAR, Liu (2015), and Wang (2013) trust models just evaluate attack resistance trust feature. Therefore, the trust consumers who like to use trust models that both

evaluate attack resistance and punishment can use DFET otherwise, if they are not concerned about punishment, they can use other reputation based trust models that evaluates attack resistance trust feature.

Table 4.2. Evaluating Reputation Based Trust Models – Part B

Challenges	Lack of Identity Management		Lack of Privacy		
	Identity Management	Trust Bootstrapping	Transferability	Heterogeneity	Security Enabled Trust
AMRep	×	×	×	×	×
TWFS	×	✓	✓	×	×
CSRD	×	×	×	×	×
Raghebi (2013)	×	×	×	×	×
RBTM	×	✓	×	×	×
CloudArmor	✓	×	×	×	✓
DFET	×	×	×	×	×
Rizvi (2014)	×	✓	×	×	×
BAR	✓	✓	×	×	×
Agheli (2014)	×	×	×	×	×
ARVTM	✓	×	×	×	✓
Liu (2015)	×	×	×	×	✓
TRSC	×	×	×	×	×
Singh (2014)	×	×	×	×	×
CSRS	×	✓	✓	×	×
TMS	✓	✓	×	×	×
Firdhous (2011)	×	✓	×	×	×
Wang (2013)	×	×	×	×	✓

Table 4.2 analyzes trust models that evaluate trust based on lack of identity management and lack of privacy assessment criteria. Considering lack of identity management, two trust models

named BAR and TMS both evaluate identity management and trust bootstrapping. In addition, CloudArmor and ARVTM are the Raghebi (2013) that just evaluate identity management trust feature and TWFS, RBTM, Rizvi (2014), CSRS and Firdhous (2011) are the trust models that just evaluate trust bootstrapping. Therefore, based on the CC concern in case of lack of identity management they can choose the trust model that does their required evaluation. Table 4.2 also analyzes selected reputation based trust models based on three assessment criteria that address lack of privacy. Among all of the studied trust models, just TWFS and CSRD evaluate transferability. In addition, none of the trust models evaluate the heterogeneity support and CloudArmor, ARVTM, Liu (2015) and Wang (2013) trust models are evaluating security enabled trust. If a CC is trying to find a reputation based trust model that address the lack of confidentiality support, these trust models can help them. None of the studied trust models evaluate the three selected assessment criteria together. Therefore, if lack of privacy is important for a CC, he can choose a reputation based trust model based on the trust feature that is more important for him.

Table 4.3 analyzes trust models that evaluate trust based on lack of reputation, lack of SLA Support and lack of transparency. Considering lack of reputation, CloudArmor, DFET, Rizvi (2014), ARVTM, Liu (2015), CSRD evaluate trust based on outside in method which means not only they evaluate trust based on CSP reputation but also based on architectural concerns. While, other trust models evaluate trust based on black box approach which means they just evaluate trust based on CSP reputation. In addition, TRSC, Raghebi (2013), Singh (2014) and Wang (2013) evaluates local trust while others evaluate based on global trust and BAR is the only reputation based trust model that evaluates trust based on both methods. Furthermore, all trust models evaluate reputation enable trust. CloudArmor, DFET, BAR, Liu (2015), TRSC, Singh (2014),

CSRS and Wang (2013) can evaluate decentralize trust models while others evaluate centralize trust model.

Table 4.3. Evaluating Reputation Based Trust Models – Part C

Challenges	Lack of Reputation					Lack of SLA Support	Lack Of Transparency
	Trust evaluation	Trust Scope	Reputation Enabled Trust	Trust Aggregation	Trust Evidence	SLA Verification	Transparency
AMRep	Black Box	Global	✓	Centralize	-	×	×
TWFS	Black box	Global	✓	Centralize	Direct / Indirect	×	×
CSRD	Black box	Local	✓	Centralize	Direct	✓	✓
Raghebi (2013)	Black box	Local	✓	Centralize	Indirect	×	×
RBTM	Black box	Global	✓	Centralize	Direct	✓	
CloudArmor	Outside in	Global	✓	Decentralize	Direct	×	×
DFET	Outside in	Global	✓	Decentralize	Direct	✓	✓
Rizvi (2014)	Outside in	Global	✓	Centralize	Indirect	×	×
BAR	Black box	Local/Global	✓	Decentralize	Direct	×	×
Agheli (2014)	Black box	Global	✓	Centralize	Direct/Indirect	×	×
ARVTM	Outside in	Global	✓	Centralize	Direct	×	×
Liu (2015)	Outside in	Global	✓	Decentralize	Indirect	×	×
TRSC	Black box	Global	✓	Decentralize	Direct/Indirect	×	✓
Singh (2014)	Black box	Local	✓	Decentralize	Indirect	×	×
CSRS	Outside-in	Global	✓	Centralize	Indirect	×	×
TMS	Black-box	Global	✓	Decentralize	Direct/Indirect	×	✓
Firdhous (2011)	Black box	Global	✓	Centralize	Direct	×	×
Wang (2013)	Black box	Local	✓	Decentralize	Direct	×	×

Finally, Raghebi (2013), Rizvi (2014), Liu (2015), Singh (2014), and CSRS are evaluating indirect trust evidence, however, TWFS, Agheli (2014), TRSC, TMS and Wang (2013) are evaluating both direct and indirect trust models while others evaluate direct trust evidence. By

considering lack of SLA support, CSRD, Raghebi (2013), and DFET are having SLA verification mechanism. In addition, regarding lack of transparency, CSRD, DFET, TRSC, and TMS are evaluating transparency in trust management systems.

4.2. Authentication Based Trust Models

As discussed in section 2.2.2 authentication based trust models use encryption and key management technologies to establish trust between CCs and CSPs. This category includes trust models that ensures the availability, integrity and confidentiality of data on cloud by using certificates from standardized body, trust tickets, private and public keys, TPM endorsement keys and etc. and evaluates the confidence of CCs regarding the expected behavior of cloud services. In this section in order to evaluate authentication based trust models some of the recent trust models have been studied and analyzed based on the assessment criteria discussed in chapter 3.

- **Krautheim et al. (2010) [103]**

The authors proposed a trust model called Trusted Virtual Environment Module (TVEM) [103] is presented as a software appliance. For cloud environments already provided with Trusted Platform Module (TPM) virtualization techniques, TVEM introduces better features like improved application program interface (API), cryptographic algorithm flexibility, and a configurable modular architecture.

Also a unique Trusted Environment Key is introduced, combining trust from the information owner, and the CSP to create a dual root of trust for the TVEM that is distinct for every virtual environment and separate from the platforms trust [103].

Figure 4.15 shows the configuration of a Host Platform (HP) with multiple virtual environments that require a TVEM. The virtual environment may be an entire virtualized OS that supports many applications or a special purpose virtual environment that performs a single

application. The TVEM lies between the hypervisor and its associated VM. The hypervisor must be aware of TVEMs and provide support via a TVEM manager. The TVEM manager provides mediation for TPM services from each TVEM and other processes that require TPM services. The host platform must provide the TVEM manager and allow access for TVEMs. The host platform's TPM is used as the RTS to secure the TVEM's private information on the HP. A transitive trust chain is built from the TPM through the hypervisor and TVEM manager to the TVEM ensuring trust in the TVEM is rooted in the hardware trust of the platform [103].

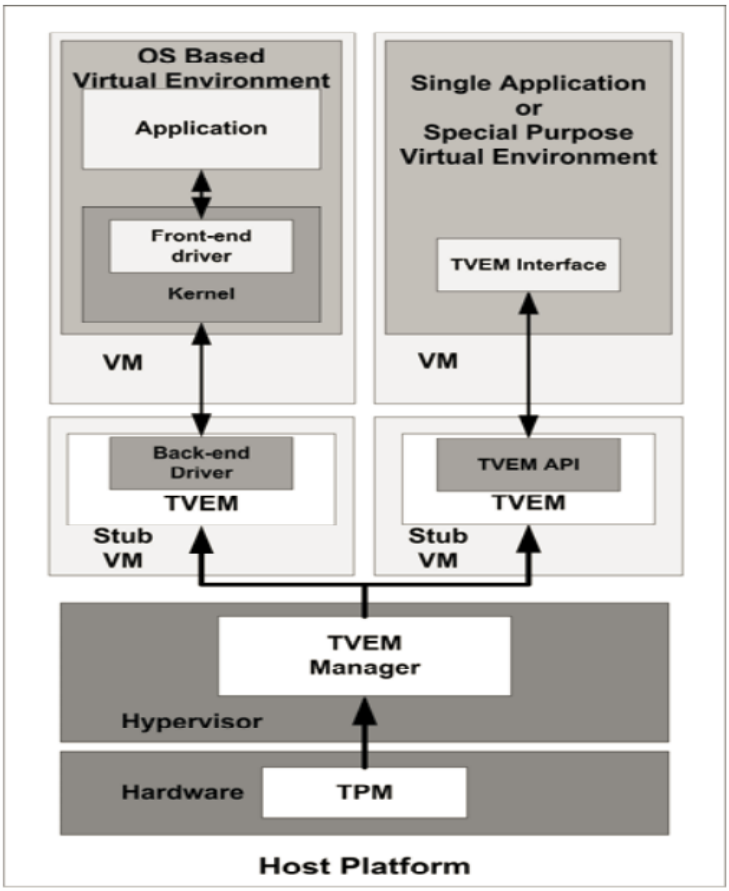

Figure 4.15. Configuration of a host platform with multiple virtual environments [103]

- **Guoyuan (2014) [104]**

The authors proposed a trust model named Mutual Trust Based Access Control (MTBAC) which not only considers user's behavior trust and ensure that user's access request poses no

malicious threat to cloud server, and also takes cloud service node's credibility into account. Trust relationships between users and cloud service nodes are established by mutual trust mechanism and only trusted users have access to the Cloud, and simultaneously users can select the most credible cloud service nodes [104].

The physical structure of MTBAC consists of users, Authentication and Authorization Center (AAC), cloud service nodes, user's behavior trust database and cloud service node's trust database. User represents individuals or organization who request access to cloud services or resources. Cloud service nodes are entities that provide services or resources to users in cloud computing platform. User's behavior trust database and cloud service node's trust database store interact history, behavior information, trust values and trust models of the User and Cloud service nodes respectively. According to user's behavior in user's trust database, AAC will detect user's behavior in the first place in order to prove user's identical legitimacy and behavior trustworthy. And then sort nodes according to trust levels and recommend the best service node for the user. AAC verifies user's legitimacy firstly, including identity legitimacy and behavior trust. AAC ensures that only when user's trust degree is higher than the trust threshold, user's access request can be accepted by cloud server. Afterwards, the most suitable cloud node will be selected to provide services according to user's request and node's credibility [104].

The access control policy of MTBAC can not only guarantee that access request of users could get response, but also ensure that all cloud service nodes can't be attacked or illegally occupied by malicious users. The algorithm of MTBAC is as follow [104]:

1) In the process of resource requesting and service providing, after the user submits access request to AAC, AAC first check that the user has a valid authentication token.

2) According to the user's identity, AAC start to access user's behavior trust database and obtain the user's behavior trust level. Compare the user's trust level with the trust threshold, if it is higher than the threshold, turn to step (3); else, refuse to provide services to the user.
3) Read the user's access request, and put all the cloud nodes which could provide the corresponding service into the candidate node queue.
4) Select the best service node in the candidate node queue and give the user the service access right.
5) The best service node provides services to the user and then updates user's trust degree.

- **Manuel et al. (2009) [105]**

The authors proposed a trust model named Grid and Cloud Trust Model which is a trust model that is integrated with CARE resource broker [104]. The proposed trust model can support both grid and cloud systems. The resource broker has been implemented with Kerberos Based Authentication Module and PERMIS Role Based Authorization Module to enhance the security measure of the broker compared to the conventional security mechanism incorporated in it. Kerberos is a network authentication protocol. It allows nodes communicating over a non-secure network to prove their identity to one another in a secure manner. Kerberos is aimed primarily at a client-server model, and it provides mutual authentication between the user and the server to verify each other's identity. PERMIS is a policy controlled role based authorization system that uses digitally signed X.509 attribute certificates or Kerberos tickets to hold user's roles/attributes. The PERMIS based authorization makes the decision for the user's access to be granted or denied based on the policy for the target domain [105].

The architecture for Trust Management System for Grid and Cloud resources has been shown in Figure 4.17. The model computes trust using three main components namely, Security Level Evaluator, Feedback Evaluator and Reputation Trust Evaluator. Security Level Evaluation has been carried out based on authentication type, authorization type and self-security competence mechanism. Multiple authentication, authorization mechanism and self-security competence mechanisms are supported. Depending on the strength of individual mechanism, different grades are provided for trust value. Feedback Evaluation also goes through three different stages namely feedback collection, feedback verification and feedback updating. The Reputation Trust Evaluator computes the trust values of the grid/cloud resources based on their capabilities based on computational parameters and network parameters. Finally the overall trust value has been computed taking the arithmetic sum of all the individual trust values computed.

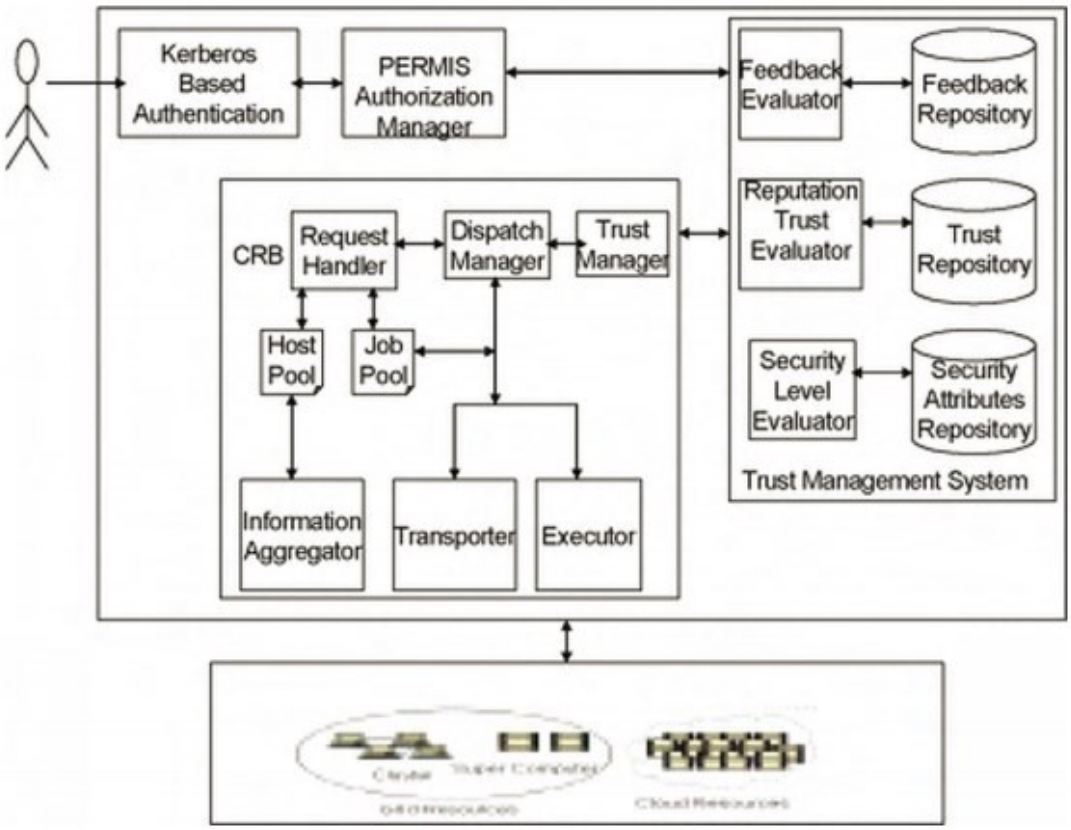

Figure 4.16. Grid and cloud resource trust model architecture [105]

- **Wang et al. (2010) [106]**

The authors proposed a trust model named Family-gene Based model for Cloud Trust (FBCT) which is a cloud trust model based on family gene and confirms family blood relation according to family gene. The family gene model is divided into three stages. The initialization of family gene system in cloud, the identification of family gene in cloud, and the assignment of family gene. In the initialization stage the ancestor cloud node of the whole cloud family trust is established and Family gene and aberrance gene of the ancestor cloud node are produced. The process of the ancestor of cloud family generates a new family member is as follows: 1) A new node is created. 2) The node family gene is generated according to the ancestor aberrance gene, the registration information of the member. The ancestor assigns the aberrance gene of node. The node member gene is produced. 3) The new member is take on as "child", "visitor", or "the ancestor" according to the type of a child [106]. Figure 4.18 shows the FBCT model [105].

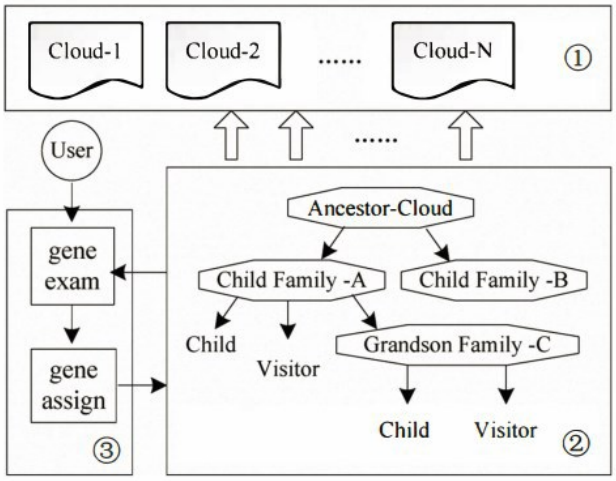

Figure 4.17. Cloud trust model based on family gene [106]

In the identification phase firstly the ancestral aberrance gene of cloud family members are obtained and signature information from its aberrance gene are extracted. Secondly it will be decrypted with the private key. Then that user's family gene is extracted and the user's family gene

code are compared with the deciphered aberrance gene code. If both codes are equal, visitor is considered as the family member.

In the assignment of Family gene system gene stage examining agent confirms that the gene belongs to the family. If the visitor is a new user, the ancestor produce a corresponding family gene and an aberrance gene while the ancestor is creating a new family member node for it. If there has been the visitor, a privilege level and a gene role is assigned to the user. The gene role is endowed a certain privilege and behavior (read, write).

Figure 4.19 represents the trust Relationship between cloud family members. Users of the brothers in the same class trust each other therefore they can visit each other, but the relation of the brothers' kid (cousinly trust) is delivery-trust at a certain time. The relation of the brothers' kid is shown in the right view in the Figure 4.19. At a moment, If F trust E, namely E→F, E can visit F at this time. And then S is a child of F, so according to genetic trust relation, then as long as the father trust is satisfied, the sub trust will be satisfied. The S trusts F, F→S. At this time E→F, the S trusts E, namely E→S, the E can visit S, a child of F. But the E does not believe F at this time, so the deliver can't be contrary. If E believes F, trust can be delivered according to this principle [106].

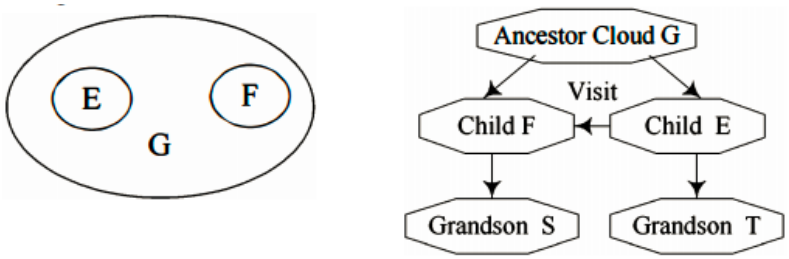

Figure 4.18. Trust relationship between family members [106]

- **Wan et al. (2011) [107]**

The authors proposed a trust model named Hierarchical Attribute Set Based Encryption (HASBE) [107] which is a full-fledged access control scheme for cloud computing. The HASBE

scheme seamlessly incorporates a hierarchical structure of system users by applying a delegation algorithm to ASBE. HASBE not only supports compound attributes due to flexible attribute set combinations, but also achieves efficient user revocation because of multiple value assignments of attributes [107].

Figure 4.20 shows the hierarchical structure of system users. HASBE model consists of a trusted authority, multiple domain authorities, and numerous users corresponding to data owners and data consumers. The trusted authority is responsible for generating and distributing system parameters and root master keys as well as authorizing the top-level domain authorities. A domain authority is responsible for delegating keys to subordinate domain authorities at the next level or users in its domain. Each user in the system is assigned a key structure which specifies the attributes associated with the user's decryption key [107].

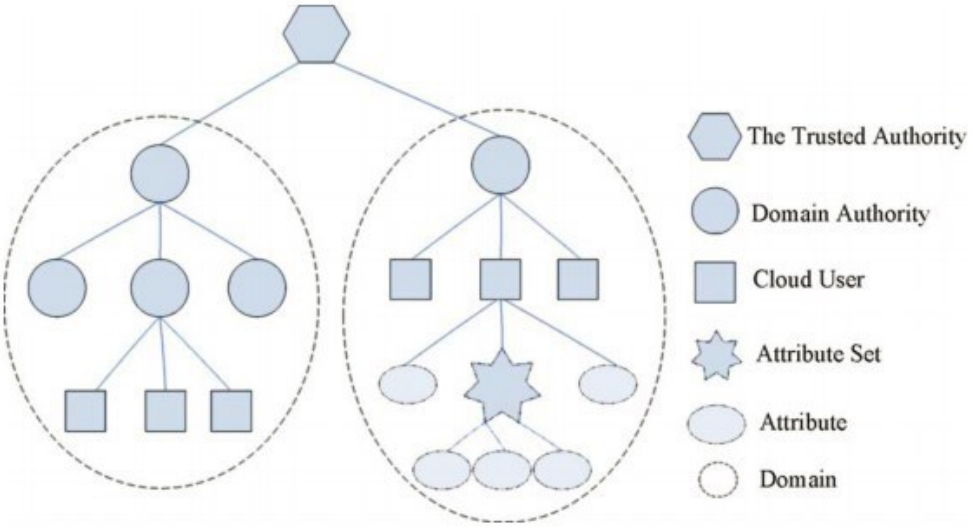

Figure 4.19. Hierarchical structure of HASBE system users [107]

Zhiguo Wan et al. believe that HASBE achieves great scalability, can support compound attributes and multiple numerical assignments for a given attribute conveniently, can easily achieve fine-grained access control, has efficient user revocation, a more natural access control system [107].

- **Shen et al. (2010) [108, 109]**

The authors have analyzed the security of cloud computing environment and described the function of trusted computing platform in cloud computing [108, 109]. They have also proposed a method named Trusted Platform Software Stack (TSS) to evaluate the security and dependability of cloud computing integrating the Trusted Computing Platform (TCP) into the cloud computing system. The TCP has been used in authentication, confidentiality and integrity in cloud computing environment.

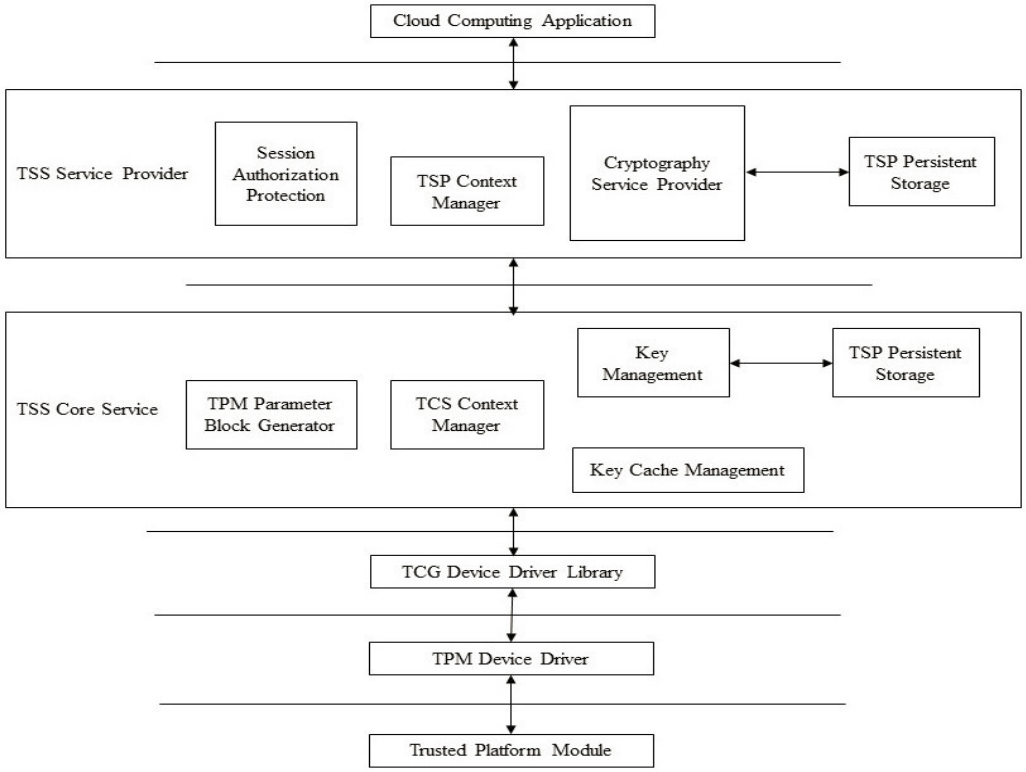

Figure 4.20. TSS architecture for cloud computing based on TCP [108]

TSS components are the major parts of the TCP enabled cloud computing. It provides fundamental resources to support the TPM. In this design, TSS should be a bridge between the up-application and the low-hardware. As depicted in Figure 4.6, TSS includes two layers, the TSS service provider (TSP) and TSS core services (TCS). The applications call the function of TSP. TSP provides some basic security function modules. These basic modules send calls to TCS. Then

TSS converts these calls to according TPM instructions. Since TPM is hardware, the TCG Device Driver Library (TDDL) is necessary. TDDL convert the calls from TCS to the TPM orders. After the TPM process the order, it will return the results up forward. Each layer gets results from low layer and coverts them to responding results that the up layer needs [108].

- **Yong et al. (2010) [110]**

The authors proposed a trust model named Multi-Tenancy Trusted Computing Environment Model (MTCEM) [110] which is a model for cloud computing [110]. Since cloud facilities belong to multiple stakeholders such as CSPs and CCs, they belong to multiple security domain and server different security subjects simultaneously. The different stakeholders may be driven by different motives such as best service, maximization of the return on investment and hence may work detrimental to the other party involved. Hence cloud computing should have the capability to compartmentalize each CC and CSP and support security duty separation defining clear and seamless security responsibility boundaries for CSP and CCs.

MTCEM has been designed as two-level hierarchy transitive trust chain model which supports the security duty separation and supports three types of distinct stakeholders namely, CSP, CCs and auditors. In this model, CSP assume the responsibilities to keep infrastructures trusted while the CC assumes responsibility starting from the guest OS which installed by the CC on the Virtual Machines provided by the CSP. The auditor monitors the services provided by the CSP on behalf of the CCs. The authors have implemented a prototype system to prove that MTCEM is capable of being implemented on commercial hardware and software. But no evaluation of the prototype on performance has been presented. Figure 4.7 shows the MTCEM architecture.

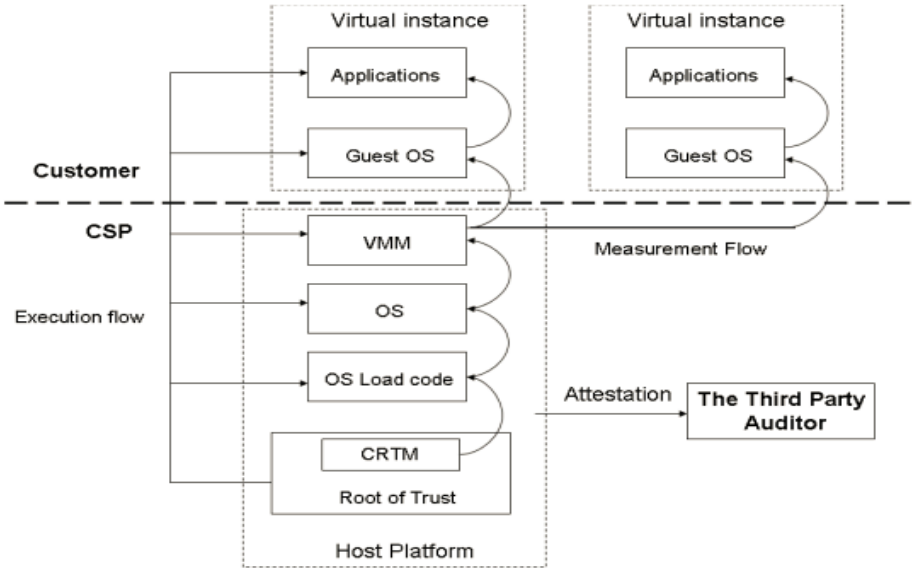

Figure 4.21. MTCEM [110]

- **Ma et al. (2012) [111]**

The authors proposed a trust model named Trusted VM Clone Model (TVMCM) [111] which is consisted of two individual models TCVMM and TCVDM where TCVMM ensures the security when cloning the memory of VM and TCVDM ensures the security when cloning the disk of VM. TVMCM resolves three problems: the identities verification of involved servers; the attestation of source VM and destination VM; the protection of integrity of transmitted data. TVCM relies on the concept developed in Trusted Computing Group (TCG) and take use of hardware elements such as Trusted Platform Module (TPM), and the emulated TPM for virtual machines vTPM is involved.

There are three functions provided by TCVMM: first, before the cloning procedure begins, S_L and S_R should be able to authenticate the identity of each other; second, VM_S should be able to authenticate the identity of newly generated VM VM_D; finally, during the procedure, S_R should be able to authenticate the memory data transmitted from S_L.

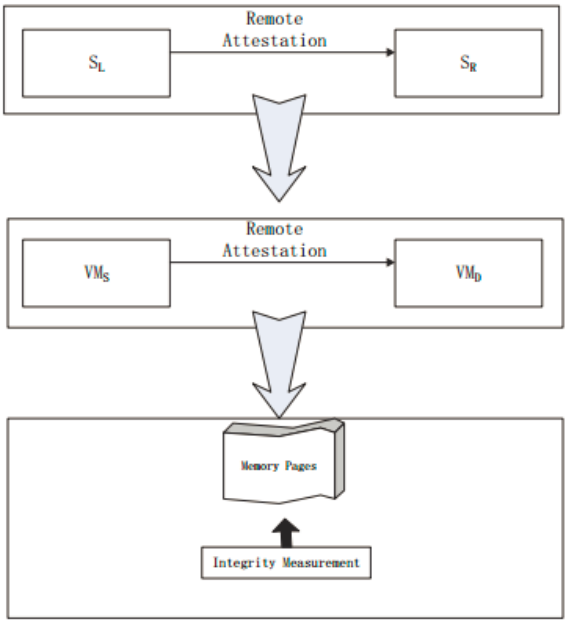

Figure 4.22. TCVMM architecture [111]

There are two phases in TCVMM: phase 1 is an attestation process between servers and VM_S, and phase 2 is integrity measurement of memory pages during the cloning process. Figure 4.24 shows the architecture of TCVMM [111].

In TCVDM before the cloning process start the identities of involved entities have been attested. Unlike virtual memory, virtual disks are always with big size and hard to be transferred over network. To reduce the overhead, the authors of this thesis have designed a mechanism to clone virtual disk in TCVDM. When VM_D is generated, an empty virtual disk is assigned to it and a bitmap is associated to it. This bitmap indicates the status of every data block and with this bitmap the virtual disks of VM_S and VM_D are logically connected. VM_D determines how to read/write data from/to its virtual disk by inquiring this bitmap. When VM_D reads data from local virtual disk, the data will be transferred from the virtual disk owned by VM_S over the network. During the cloning process, VM_S execute the write operation to its virtual disk should be reflected to VM_D's virtual disk. So when VM_S writes data to its own virtual disk, VM_D will write the same content to the same block of VM_D's virtual disk. So the security issue goes to verify the integrity

of data transmitted from VM_S before VM_D use the data. Simply, this process is like memory integrity verification: before the data is transmitted, it will be hashed and vTPMS will use its AIK to sign the hash value which will be sent to VM_D with hashed value and the data itself. By checking the hash value of received data, VM_D determines to trust the data or not. This mechanism ensures the efficiency of TCVDM. The above figure 4.24 illustrates the process of TCVDM [111].

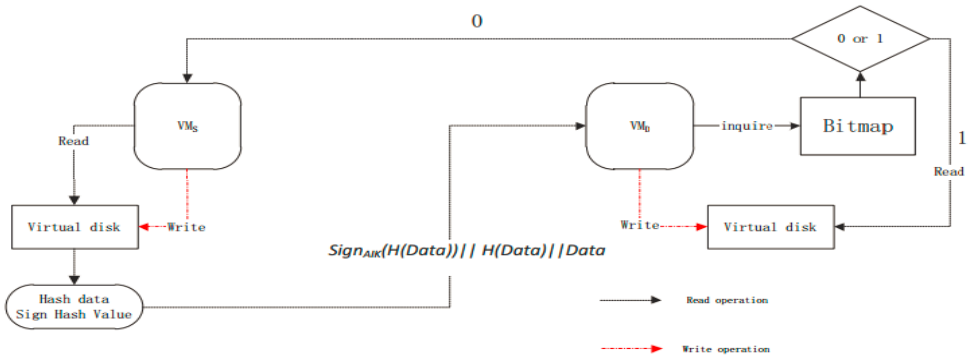

Figure 4.23. TCVDM process [111]

- **Kashif et al. (2015) [112]**

The authors proposed a trust model named Distributed Trust Protocol (DTP) [112] which offers the opportunity to the consumer to take part in securing the VM. The proposed protocol is distributed between the provider and the consumer. In the protocol client stores the hash values of VM components (BIOS, Boot Loader and OS) in its Platform Configuration Registers (PCRs) of Trusted Platform Module (TPM) at client side. Likewise, in Trusted Boot Process, in which TPM measures (hashes) all the software and firmware components, including the BIOS, boot loader, and operating system kernel etc. before they are loaded and stores hash values in PCRs of TPM. In this DTP, TPM at consumer side is linked with VM hosted at provider side. By doing so, consumer VM will be booted according Trusted Computing Group (TCG)'s Trusted Boot Process. Hence integrity of VM can be checked by utilizing consumer's infrastructure [112].

This model is consist of three steps. In the step 1 of protocol, consumer initiates communication by sending nonce N and his public part of his Attestation Identity Key (AIK_{pub})

and this complete message is encrypted with the public key of provider (P_{pub}). In the next step session key is encrypted with the AIK_{pub} so that the session should be dedicated to the consumer, and the whole message is encrypted with the private key of provider (P_{pr}). So that it can be said that this message has come from the provider. In the third step, user VM at provider side is prepared when consumer starts its VM, it gets booted according to the TCG's Trusted Boot Process. In the boot process of VM digest value of the VM components (BIOS, Boot Loader, Kernel, OS etc.) are calculated with support of consumer's TPM and these values are sent to the consumer by encrypting the message with the AIK_{pub} After some period of time when user gets the VM restarted all the measurements of VM components are again computed and matched with previous values. If the measurement values are found to be same the VM starts normally otherwise consumer is notified regarding the tempering of VM. Figure 4.24, graphically presents the process of the VM integrity checking protocol [112].

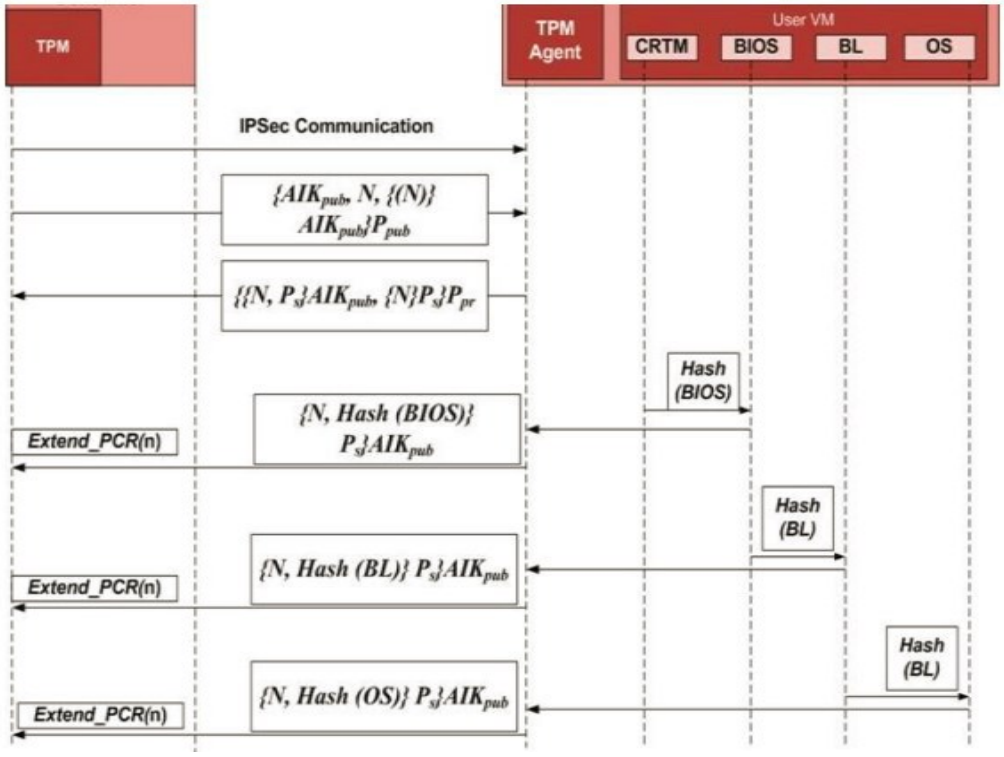

Figure 4.24. Distributed trust protocol [112]

- **Han et al. (2010) [113]**

The authors proposed a trust model named Improved Trusted Cloud Computing Platform Model (Improved TCCP) which is used Direct Anonymous Attestation (DAA) and Privacy CA scheme to evaluate the anonymity and availability of the TCCP model. This model ensures the confidentially and the integrity of a CC's VM, and is able to solve the dependence issue on the Trusted Coordinator (TC) [113].

The TCCP includes two components: a TVMM and a TC. A TVMM adopts the technologies from trusted system, hence is able to protect its own confidentiality and integrity. According to different demands of CCs, OS can choose to run in a standard virtual machine associates with a unique vTPM instance ("open-box trusted VM") which has the trusted computing functionality, or a closed-box virtual machine associates with a unique vTPM instance("closed-box trusted VM"). Figure 4.19 shows the TCCP architecture. In this picture the TC Manager (TCM) makes the connection between the TC and the CC. The role of managing the nodes will be transferred to the Trusted Node (TN) itself based on the neutral feature of the TPM. Every node can link with any other nodes in the same zone through the Cluster Controller (CC). TNs in light blue are ordinary trusted nodes, while green TNs are selected as privacy Certification Authority (CAs) by the TC, each of them manages a portion of TNs. Green TNs contact with each other through the secretive dark blue TN. The reason why we use multiple green TNs is that a zone in the cloud contains a very large number of nodes [113].

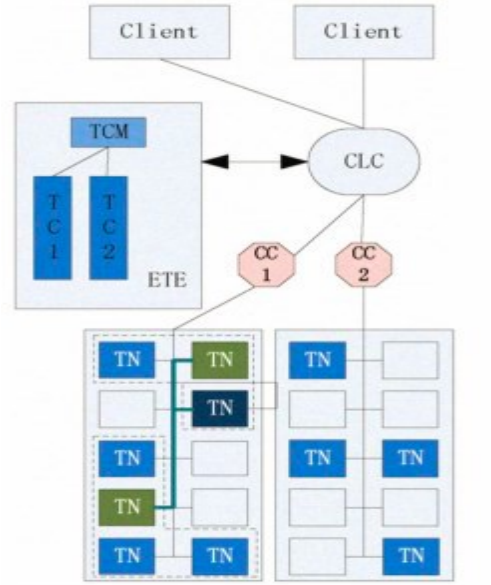

Figure 4.25. The TCCP architecture [113]

- **Navneet et al. (2013) [114]**

The authors proposed a trust model named Effiecient Watermarking Technique [114], which is an efficient approach for security of cloud using watermarking technique. Their method ensures that the attacks involved in cloud computing such as distributed denial of services, cloud malware injection attack, side channel attack can be avoided up to some extent. Also, it ensures that even if the data is steeled by an intruder or third party then also the copyright protection is achieved as the watermarking is done by shifting the coefficient [114].

The proposed algorithm transforms original image X into 256×256 standard image as the original image, pick up the decomposed component get a 128×128 image where the decomposed low frequency coefficient matrix is A (128×128). For an added watermark equally distributed in 128×128 pixel image, matrix A is blocked by 8×8 size and then make DCT for each block, choose the first value in the matrix composed of DCT transformed coefficient of each block as $F_n(1,1)$ $(1 \leq n \leq 1024)$. Read the binary watermark $(1 \leq n \leq 1024)$ firstly according to positive direction Z scanning (from left to right and from upper to lower), transform watermark

image to the sequence $\{W_k\}$ as the length of N, create $0 \sim N-10$ random sequence $\{r_j\}$ as random seed of key K and increase it to $\{W_k\}$ sequence to create a new watermark sequence $\{W_k\}$. The new watermark is embedded in $F_n(1,1)$. This embedding algorithm ensures that coefficient with larger amplitude in original chart field corresponds to watermark image DCT with larger amplitude. Another issue is that DCT coefficient is comprised watermark image with low frequency information. Hence the image will be secured in the data centers where there is a lack of trust among users and data owners the DCT will work as the key factor in providing robustness against distortion attacks as well as other security threatening attacks [114].

- **I.Sudha et al. (2014) [115]**

The authors proposed a trust model named Hash algorithm [115] which is a method of evaluating security by using HASH algorithm for the periodic authentication to ensure whether the legitimate users are accessing the data. The proposed system has the most secure authentication mechanism in accessing the data because, a periodic authentication is made to ensure whether the legitimate users are accessing the data in the cloud. In the existing system using RSA algorithm, the key is generated to ensure whether the legitimate users are accessing the data in the cloud and continuous monitoring will be taken by providing periodic authentication. In this method it first produces the hash values for accessing data or for security. A hash value is a number generated from a string of text. The hash is substantially smaller than the text itself, and is generated by a formula in such a way that it is extremely unlikely that some other text will produce the same hash value. Then the sender generates a hash of the message, encrypts it, and sends it with the message itself. The recipient then decrypts both the message and the hash, produces another hash from the received message, and compares the two hashes. If they're the same, there is a very high probability that the message was transmitted intact [115].

- **Hwang et al. (2012) [116]**

The authors proposed a trust model named Secure Resources and Data Coloring is a trust-management scheme augmented with data coloring and software watermarking. It is using a trust-overlay network over multiple data centers to implement a reputation system for establishing trust between service providers and data owners. Data coloring and software watermarking techniques protect shared data objects and massively distributed software modules. These techniques safeguard multi-way authentications, enable single sign-on in the cloud, and tighten access control for sensitive data in both public and private clouds [116].

This model has two layers of trust overlays (Figure 4.26). At the bottom layer is the trust overlay for distributed trust negotiation and reputation aggregation over multiple resource sites. This layer handles user or server authentication, access authorization, trust delegation, and data integrity control. The upper trust overlay deals with worm signature generation, intrusion detection, anomaly detection, DDoS defense, piracy prevention, and so on. These two layers facilitate worm containment and IDSs to protect against virus, worm, and DDoS attacks [116].

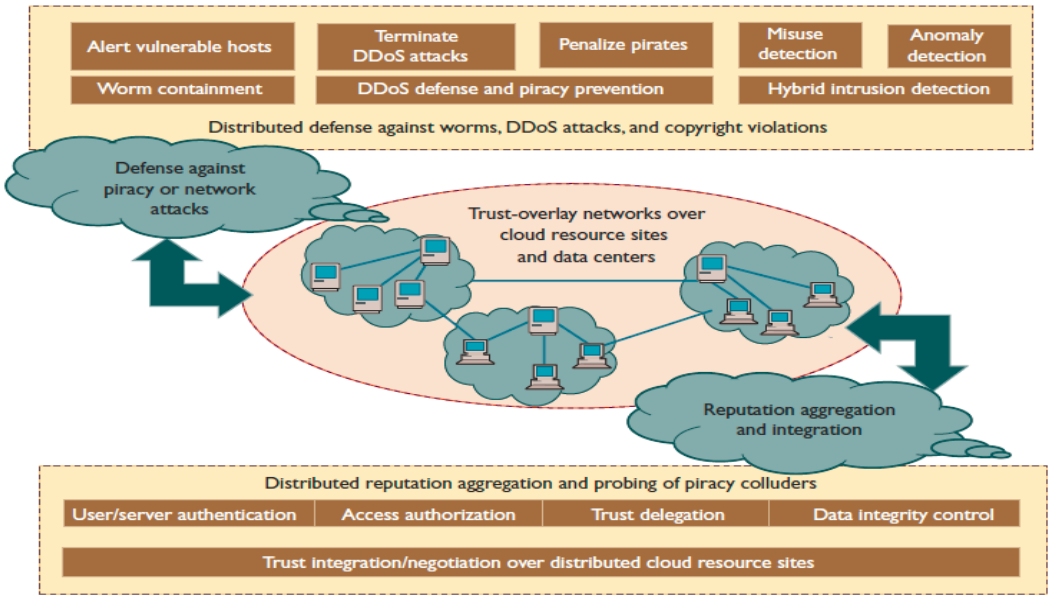

Figure 4.26. Distributed-Hash-Table (DHT)-based trust overlay networks [116]

This model uses data coloring or watermarking at the software file or data object level to segregate user access and insulate sensitive information from provider access. The coloring process uses three data characteristics to generate the color: the Expected value (Ex) depends on the data content, whereas Entropy (En) and Hyperentropy (He) add randomness or uncertainty, which are independent of the data content and known only to the data owner. Figure 4.27(a) shows the forward and backward color-generation processes. This model has added the cloud drops (data colors) into the input photo (left) and removed color to restore the original photo (right). Figure 4.27(b) shows the details involved in the color-matching process. Figure 4.27(b) shows the details involved in the color-matching process, which aims to associate a colored data object with its owner, whose user identification is also colored with the same Ex, En, and He identification characteristics. The color-matching process assures that colors applied to user identification match the data colors [103].

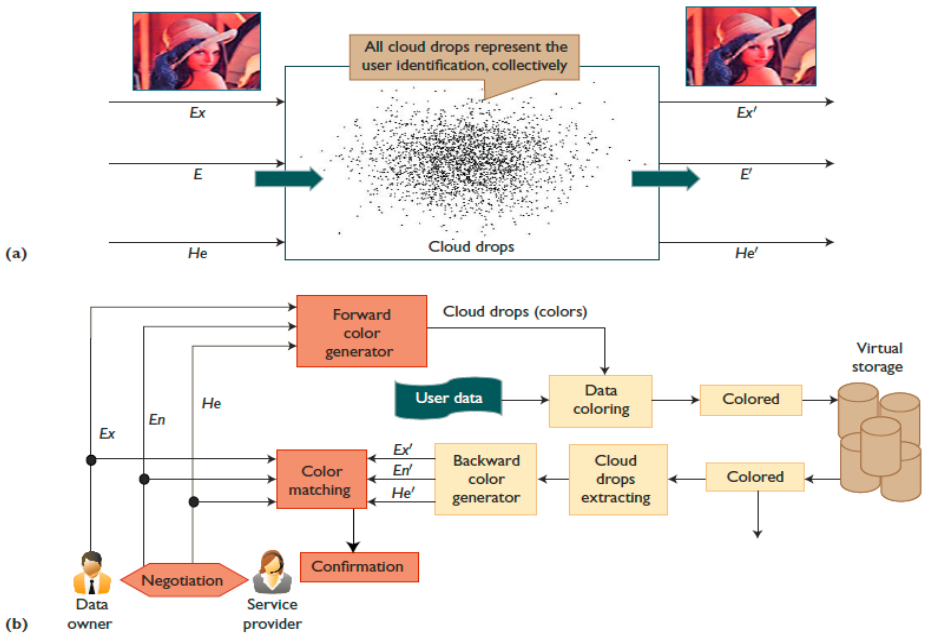

Figure 4.27. Data coloring and water marking technique. (a) forward and backward data coloring processes by adding or removing unique cloud drops (colors) in data objects. (b) Data coloring and user identification color matching through trust negotiation [116]

4.2.1. Analyzing Authentication Based Trust

Tables 4.4, 4.5, and 4.6 represent the analysis of authentication based trust models based on the selected assessment criteria defined in chapter 3. The assessment criteria and the cloud computing challenges that each trust feature address are indicated in the tables. In the tables × means the trust model does not evaluate the specified assessment criteria and ✓ means that it evaluates the specified assessment criteria. Also, there are some assessment criteria that if the trust model evaluates them instead of having ✓ as an indicator, it is mentioned how it support that criteria such as attack resistance, trust evaluation, trust scope and trust aggregation.

Table 4.4. Evaluating Authentication Based Trust Models – Part A

Challenges	Lack of Confidentiality	Lack of Reliability	
	Encryption	Attack resistance	Punishment of Agreement Violation
TVEM	AIK	SHA1 collision attacks	×
MTBAC	PKI	Individual malicious / Malicious pre-trusted peers	×
Manuel (2009)	Kerberos and PERMIS	×	×
FBCT	PKI	×	×
HASBE	PKI	×	×
TSS	PKI	×	×
MTCEM	AIK	×	✓
TVMCM	AIK	Individual malicious	×
DTP	PKI	Malicious Provider / VM Introspection / Hypervisor vulnerabilities / Root kit	×
Improved TCCP	PKI	Rudolph attack	×
Navneet (2013)	PKI	×	×
I.Sudha (2014)	RSA	cryptanalytic attack	×
Hwang (2012)	PKI	Worm containment/ DDoS	×

Based on table 4.4, all of the selected reputation based trust models evaluates encryption that address lack of confidentiality challenge. In order to find the trust models that address lack of reliability, TVEM, MTBAC, Unstable Feedback Rating Trust Model, TVMCM, DTP, Improved TCCP, I.Sudha (2014) and Hwang (2012) are the trust models that evaluate attack resistance. MTCEM is the only model that evaluates punishment of agreement violation. Therefore, the trust consumers who like to use trust models that both evaluate lack of confidentiality in cloud services can use any of the authentication based trust models. However, if they want to evaluate lack of reliability along with confidentiality, they need to decide whether attack residence or punishment of agreement violation is more important for them. Table 4.5 analyzes trust models that evaluate trust based on lack of identity management and lack of privacy assessment criteria. Considering lack of identity management, MTBAC, HASBE, MTCEM, DTP, Improved TCCP, and Hwang (2012) and TMS are the trust models that both evaluate identity management and trust bootstrapping. In addition, Secure Resource and Data Coloring, Manuel (2009), FBCT, TSS, MTCEM, and I.Sudha (2014) trust models are the reputation based trust models that just evaluate identity management trust feature. Therefore, based on the CC concern in case of lack of identity management they can choose the trust model that does their required evaluation. Table 4.5 also analyzes selected reputation based trust models based on three assessment criteria that address lack of privacy. Among all of the studied authentication based trust models, just Manuel (2009) and Improved TCCP evaluate transferability.

In addition, Manuel (2009) and HASBE trust models evaluate the heterogeneity support and all of the authentication based trust models are evaluating security enabled trust. Just Manuel (2009) evaluate the three selected assessment criteria together. Therefore, if lack of privacy is

important for a CC, he can choose an authentication based trust model based on the trust feature that is important for him.

Table 4.5. Evaluating Authentication Based Trust Models – Part B

Challenges	Lack of Identity Management		Lack of Privacy		
	Identity Management	Trust Bootstrapping	Transferability	Heterogeneity	Security Enabled Trust
TVEM	✓	✓	×	×	✓
MTBAC	✓	✓	×	×	✓
Manuel (2009)	✓	×	✓	✓	✓
FBCT	✓	×	×	×	✓
HASBE	✓	✓	×	✓	✓
TSS	✓	×	×	×	✓
MTCEM	✓	×	×	×	✓
TVMCM	✓	✓	×	×	✓
DTP	✓	✓	×	×	✓
Improved TCCP	✓	✓	✓	×	✓
Navneet (2013)	×	×	×	×	✓
I.Sudha (2014)	✓	×	×	×	✓
Hwang (2012)	✓	×	×	×	✓

Table 4.6 analyzes trust models that evaluate trust based on lack of trust on CSPs, lack of SLA Support and lack of transparency. Considering lack of reputation, Secure Resource and Data Coloring and Hwang (2012) evaluate trust based on outside in approach. While, MTBAC, Manuel (2009), and Unstable feedback rating trust model are evaluating trust based on black box approach and other trust models are evaluating trust based on inside out approach. In addition, Unstable feedback rating trust model evaluates local trust while Secure Resource and Data Coloring,

MTBAC, Manuel (2009) are evaluating trust based on global trust and other trust models do not evaluate any trust scope. Furthermore, Secure Resource and Data Coloring, MTBAC, Manuel (2009), Unstable Feedback Rating Trust Models evaluate reputation enable trust. Secure Resource and Data Coloring, Manuel (2009), Unstable Feedback Rating Trust Model and Hwang (2012) can evaluate decentralize trust models while MTBAC evaluates centralize trust model.

Table 4.6. Evaluating Authentication Based Trust Models – Part C

Challenges	Lack of Trust on CSPs					Lack of SLA Support	Lack Of Transparency
	Trust evaluation	Trust Scope	Reputation Enabled Trust	Trust Aggregation	Trust Evidence	SLA Verification	Transparency
TVEM	Outside in	Global	✓	Decentralize	Direct	×	✓
MTBAC	Black box	Global	✓	Centralize	Direct / Indirect Recommendation	×	×
Manuel (2009)	Black box	Global	✓	Decentralize	-	×	×
FBCT	Inside-out	-	×	-	Indirect	×	✓
HASBE	Inside-out	-	×	-	Direct	×	×
TSS	Inside-out	-	×	-	-	×	×
MTCEM	Inside-out	-	×	-	Direct	✓	✓
TVMCM	Inside-out	-	×	-	-	×	×
DTP	Inside-out	-	×	-	Direct	×	×
Improved TCCP	Inside-out	-	×	-	Direct / Indirect	×	×
Navneet (2013)	Inside-out	-	×	-	Indirect	×	×
I.Sudha (2014)	Inside-out	-	×	-	Indirect	✓	×
Hwang (2012)	Outside-in	Global	✓	Decentralize	Direct	×	×

Finally, Secure Resource and Data Coloring, Unstable Feedback Rating Trust Models, HASBE, MTCEM, DTP, and Hwang (2012) are evaluating direct trust evidence, however, FBCT, Navuneet (2013) and I.Sudha (2014) are evaluating indirect trust evidence and MTBAC and

Improved TCCP evaluate both direct and indirect trust models. By considering lack of SLA support, just I.Sudha (2014) trust model has SLA verification mechanism. In addition, regarding lack of transparency, FBCT, MTCEM, Hwang (2012) are evaluating transparency in trust management systems.

4.3. SLA-Based Trust

As discussed in section 2.2.2, SLA based trust models are based on contracts and agreements signed by CSPs for the delivery of different services to CCs. The trust models in this category are evaluating CSPs based on the security and quality of service attributes the CSPs included in the SLA. In this section in order to evaluate SLA based trust models some of the recent trust models have been studied and analyzed based on the assessment criteria discussed in chapter 3.

- **Alhamad et al. (2010) [117]**

The authors evaluates cloud services in order to help cloud users select the most reliable resources. It recommends the most related and trusted resources from various CSPs. The most related services mean the services which match all the main functional requirements of the desired service. This model uses the SLA management and trust techniques to provide a reliable model to select the best available provider among various CSPs to fulfil both types of requirements [117].

Figure 4.28 shows the SLA-based trust model architecture which includes SLA agent, cloud services directory, CSPs, and CC entities. The responsibilities of the SLA agent is are: Grouping CCs according to different classes based on business needs, Designing SLA metrics based on the consumers' needs, Negotiating with CSPs, Selecting CSPs based on non-functional requirements, Monitoring business activities for consumers, Monitoring SLA parameters. The CC is the entity who requests the external execution of one or more services. A CC is required to pay

the bill upon completed execution of services based on a well-defined model of prices. Trust management manages the trust relationships between CSPs and also the other users of cloud services. Cloud services directory is a directory that helps CCs to find the services they require. CSPs are the entities who own the cloud infrastructure and provide cloud services for consumers [117].

The first step of the protocol in this model is that cloud services must present their services in the cloud services directory. So, any consumer can easily find a suitable provider using the functional requirements discovery process. Then CCs use the discovery operation to find the related providers who are able to fulfil the consumers' requirements. The list of providers must be submitted to the trust management system to filter out non-trusted providers using credibility values and the reports of the SLA agent. Then a trusted list of CSPs should be sent to the SLA agent together with more details about business objectives. When CCs submit the request for cloud services, they will wait to get the Id of CSP with all details of SLA agreements. If the consumers agree to continue the contract, they will be asked to sign the SLA with the SLA agent and start to communicate with the selected provider [117].

The authors have proposed only the model and no implementation or evaluation has been developed or described. Hence the each and every module will have to be evaluated for their functionality and the effectiveness and finally the overall model will have to be evaluated for its effectiveness.

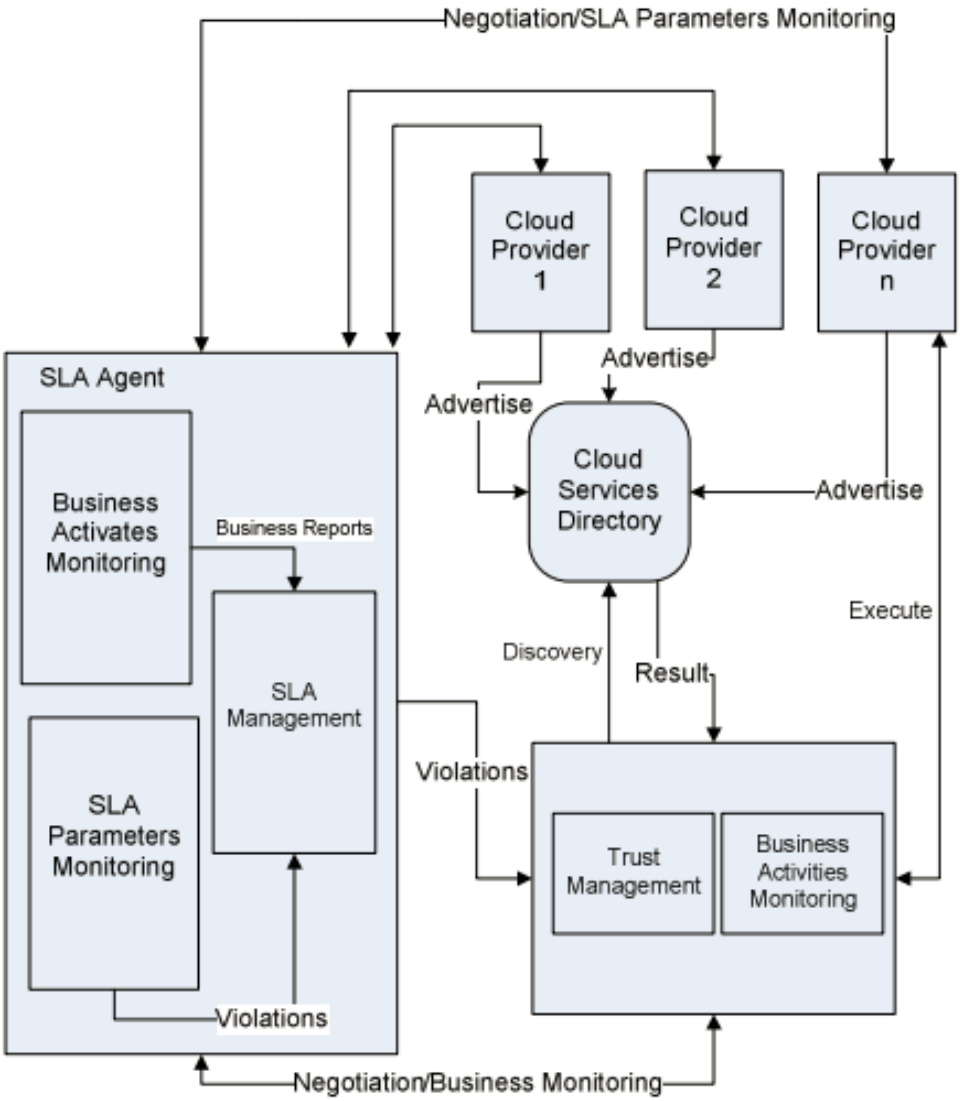

Figure 4.28. SLA-based trust model for cloud computing [117]

- **Hammadi et al. (2012) [118]**

The authors proposed a trust model named SLA Assurance which is a framework which consists of a Third-Party Service Level Agreement monitor (TP SLA Monitor) which is a service provider that monitors near real-time performance of service provider based on the SLA. The TP SLA Monitor consists of reputation assessment module and risk assessment module to meet QoS assessment. These two modules are independent but they complement each other. Figure 4.29 shows how this model works. First the TP SLA Monitor should define the total time space for

which QoS has to be assessed. Then it divides this time space into pre-interaction and post-interaction time phases. QoS evaluation in pre-interaction time phase is achieved by using reputation assessment methodology by soliciting the recommendations given by Recommending Users (RU). RU in this framework is a group of users who received a reputation query from a consumer and they reply on the basis of their past experiences with the service provider. The TP SLA Monitor aggregates the opinions of RU to reach the final reputation value of the service provider. This reputation value then can be used to determine near real-time assessment of SLA parameters in the post-interaction time phase using risk assessment methodology [118].

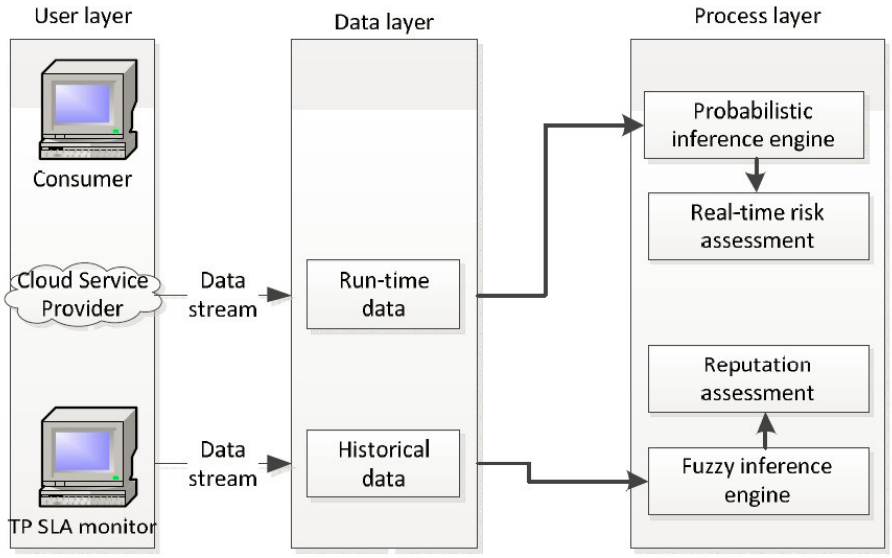

Figure 4.29. SLA monitoring framework [118]

Figure 4.30 shows the process of a reputation assessment request from a consumer. In this process the TP SLA Monitor sends a reputation query about a service provider's reputation to a network of RU. Then the TP SLA monitor has previously solicited these RU in the past and stores their credibility values in its information repository. Credibility here is defined as the trustworthiness of the opinion of a RU. It is represented on a scale of [0, 5]. Only those RU reply to the reputation query who have interacted with the service provider in the same context in which consumer wants to interact with the service provider. The reputation query reply from each RU

contains the trust value assigned to the service provider on the basis of its past interaction with that service provider and the time in which this interaction has occurred [118]. When a RU shares this trust value with TP SLA Monitor, this value is called Recommendation Opinions. Time factor is important since a RU can have multiple interactions with the same service provider in the same context but in different time slots. The time factor can be represented by a time delay which indicates the time elapsed since the last interaction of a RU with the service provider and we represent time delay on a scale of [0,5].

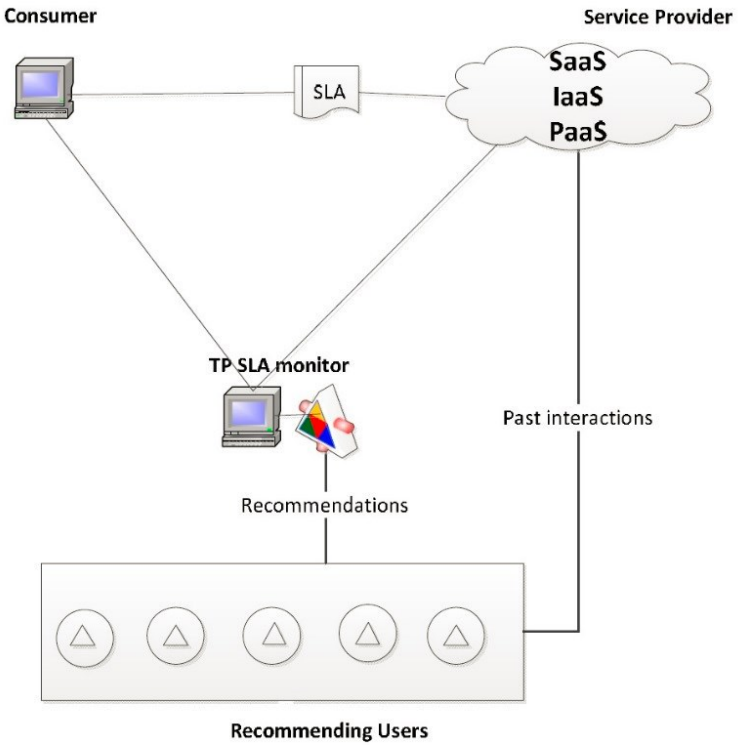

Figure 4.30. Pre-interaction time phase assessment model [118]

For each reputation query response received from a RU, the TP SLA Monitor needs to consolidate credibility value, time delay value and the Recommendation Opinion. The outcome of this consolidation is the reputation of the service provider provided by a RU. After consolidating these factors for each RU, the TP SLA Monitor then needs to aggregate all reputations by all RU involved in the reputation query. It will give the final reputation value for a service provider [133].

- **Chakraborty et al. (2012) [119]**

The authors proposed a trust model named SLA trustworthiness framework which helps a CC to estimate trustworthiness of a CSP. The main component of the framework is a 'trust evaluation engine' that evaluates consumer's trust on the CSP. Figure 4.31 illustrates the architecture of the framework, followed by descriptions of the components [119].

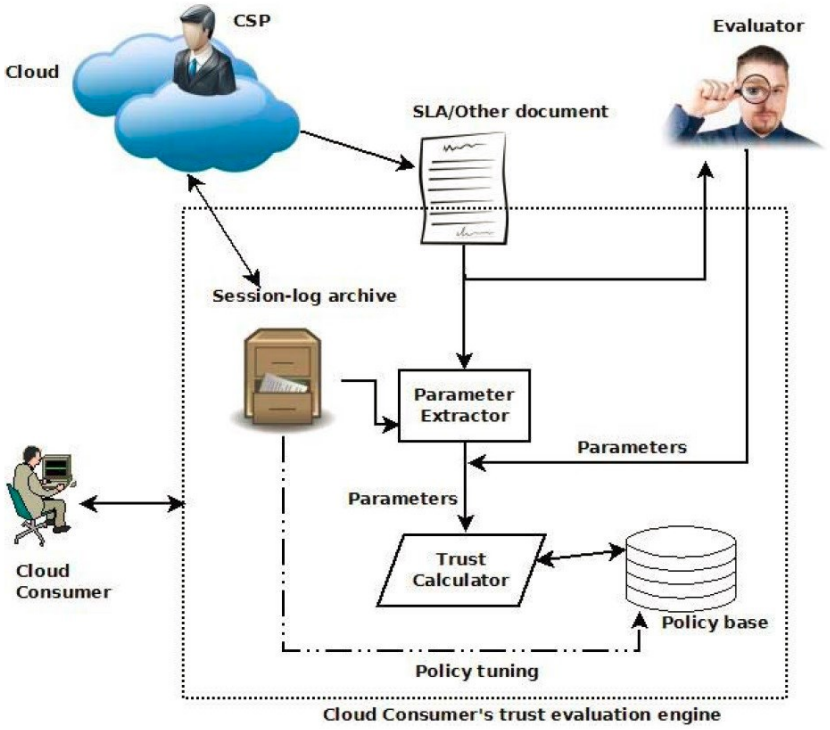

Figure 4.31. Architecture of the framework [119]

CSP is an entity who is accountable for the services provided through that cloud. CC is an entity which has job(s) to execute and seeks service from a CSP. Evaluator obtains and evaluates parameters mentioned to estimate trustworthiness of the CSP. Parameter extractor examines SLA and other documents and extracts the pre-SLA parameters. It also analyzes archived session logs to extract post-SLA parameters. Trust calculator is the module which actually evaluates the parameters and calculates trust. Session-log files stores log files containing interaction histories of all sessions between a consumer and the cloud [119].

Figure 4.32 shows a top view of the total process. Consumer's trust engine receives SLA from CSP. The trust engine sends, if needed, a request for more documents regarding the services. This step is optional as not all consumers will require this step. The CSP responds back with new or updated documents. This step is executed only if the previous step has been executed. The trust engine extracts the parameters. If needed, the trust engine sends the SLA and other documents to the evaluator. The evaluator, who is a third party expert, examines these documents and extracts parameters for trust estimation. The evaluator supplies the extracted parameter to the trust engine [119].

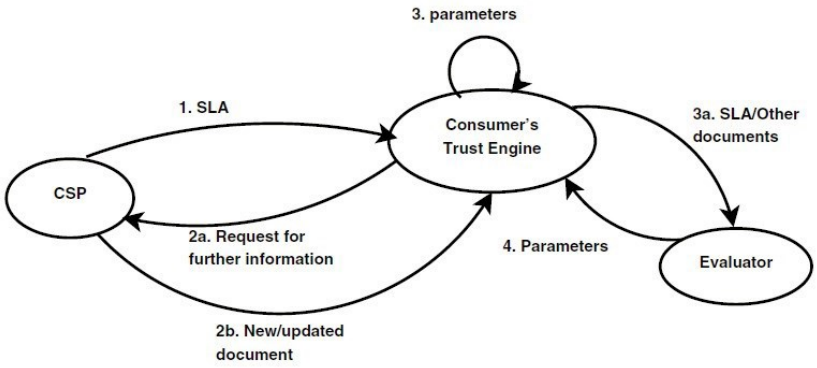

Figure 4.32. Top view of trust estimation steps [119]

The internal working of the trust engine, which worked as a black box in the first layer, is depicted in Figure 4.33.The parameter extractor receives information (SLA and other documents) about the parameters from CSP. It extracts parameters with the help of consumer's trust evaluation authentication base. Once a relationship has been established between the CSP and the consumer, the parameter extractor module receives session-logs and extract trust parameters. At this step it consults the authentication base as well. These parameters are passed to the trust calculator. The trust calculator may also receive trust parameters from the evaluator. It then evaluates the parameters and calculates trustworthiness with the aid of the authentication base. The authentication base is updated or tuned using information received from session log archive [119].

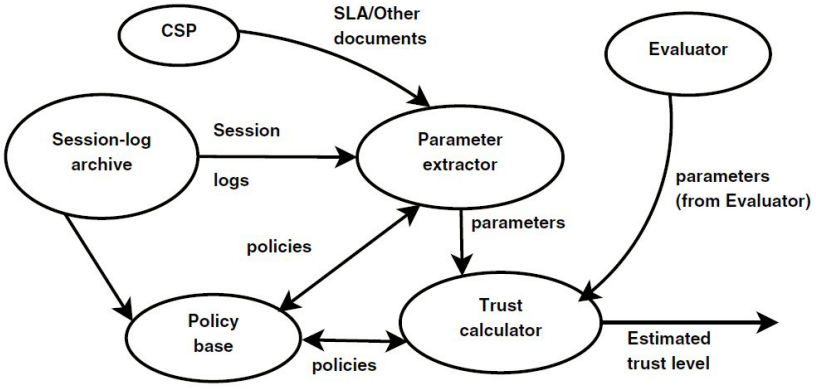

Figure 4.33. Trust estimation steps inside the trust engine [119]

- **Jules et al. (2014) [120]**

The authors proposed a trust model named Probabilistic Ontology Driven Trust Model which is allows the CC to choose a trusted provider based on its reputation calculated using a Bayesian network and continuously updated via a Cloud directory. Their model use a smart and dynamic SLA scheme that uses a probabilistic ontology capable of detecting potential violations of contract parameters; in addition, it alerts the service provider about these violations [120].

This model contains a reputation based trust module which classifies CSPs based on the probability (i.e., trust level) of each provider in violating the SLA; this probability is computed using a Bayesian inference engine that relies on historical data and measurements of low-level metrics and can be calculated from the following formula [120]:

$$\mathrm{p(H|D)} = \frac{\mathrm{p(D|H).p(H)}}{\mathrm{p(D)}} \tag{4.6}$$

Where $p(H|D)$ is the posterior probability of H given knowledge data D; $p(H)$ is the prior probability for H; $p(D|H)$ is the likelihood probability of H given D; and $p(D)$ is the probability that would have happened whether or not H is true.

In this model an autonomous agent is used which assesses the reputation of a provider using quantitative parameters like available resources and saved session log files. To measure the

trust level of a provider, the SLA parameters need to be extracted, these parameters are defining the QoS offered by the provider to meet the expectations of its CCs. Then, map the low-level metrics to high-level SLA parameters, hence they are available and used for calculations. In this model two types of mapping is used. One of them is simple mapping which is a one-to-one correspondence. For example, disk space is mapped to storage in SLA; and the second one is complex mapping which is many to-one mapping [120].

- **Lianyong et al. (2014) [121]**

The authors proposed a trust model named Fluctuant QoS and Flexible SLA (FL-FL) which is based on cloud service historical records with fluctuant QoS and flexible SLA. The main idea of FL-FL [121] is: First, calculate user's satisfaction degrees towards the criteria in historical record by comparing fluctuant QoS and flexible SLA over. Second, calculate user's satisfaction degree towards historical record by integrating the obtained values. Third, the trust of is calculated by synthesizing users' satisfaction degrees towards historical records. The trust of a cloud service in this model is calculated by the following formula [121]:

$$\mathrm{TRUST_{WS}} = \sum_{i=1}^{L}\sum_{J=1}^{m} w_j * \mathrm{Sat\_degree}(ws, \mathrm{HR_i}, q_j) \tag{4.7}$$

where ws is the cloud service, HR_i is the historic record and q_j is the criteria.

$$\mathrm{Sat\_degree}(ws, \mathrm{HR_i}, q_j) = \begin{cases} -1, & \text{if } \mathrm{MAX}(f(t)) > \mathrm{Peak_{SLA}} \\ -1, & \text{if } \int_0^T (\mathrm{Overall_{SLA}} - f(t))\,dt < 0 \\ \int_0^T (\mathrm{Overall_{SLA}} - f(t))dt \,/\, \int_0^T \mathrm{Overall_{SLA}}\,dt, & \text{else} \end{cases} \tag{4.8}$$

The first two conditions mean that the flexible SLA is violated. In these situations, user's satisfaction degree is equal to -1. While the last condition means that the flexible SLA (both

$Overall_{SLA}$ and $Peak_{SLA}$) is satisfied; in this situation, user's satisfaction degree could be calculated and normalized into range [0, 1], through a ratio between two covered areas [121].

- **Yang et al. (2013) [122]**

The authors proposed a trust model named Hybrid Trust Mechanism [122] which is a trust model in which primary includes two trust modules named the initial trust module and trust-aided evaluation module. After an initial and a basic trust is established in initial trust module, the trust-aided evaluation module will be used to verify the service provider dependable further. The initial trust technologies are SLA, the third-party audits, self-assessment mechanisms etc. Cloud users are responsible for monitoring SLA violations and applying for compensation from the service providers. Also, a professional and impartial third party is needed to provide these services. CSPs employ different cloud audit standards to assure cloud users about their offered services and platforms. The audit standards, however, are not sufficient to alleviate the cloud users' security concerns and most of the CSPs are not willing to share the audit reports, which also lead to a lack of transparency. A cloud user can study information about the service which is revealed by a service provider by requiring the self-assessment questionnaires. Accreditation is somewhat similar to audit, which can conduct independent assessment of the cloud services performance. Figure shows the trust service architecture [122].

As we can see in the figure 4.34 it has two primary trust modules between cloud users and service providers, namely, initial trust module, and trust-aided evaluation module. The initial trust module includes SLAs, cloud Audits, self-assessment questionnaire, 677 accreditation, and so on, which just provides an initial and a basic trust in selecting service providers and devoted by some public research organizations like Cloud Security Alliance. After establishing the initial and basic trust, the trust-aided evaluation module is used to verify and reevaluate the trust. The trust-aided

evaluation module encompasses three phases: trust formulation, trust evolution, and service monitor. The trust formulation unit is employed to compute the overall trust values based on the direct trust values and the reputation values by using Bayesian statistic, evidence theory or other approaches. The final trust values can be updated while taking into account the confidence factor and the time decay factor in the trust evolution phase. The confidence factor is defined as the number of transactions between cloud users and CSPs [122].

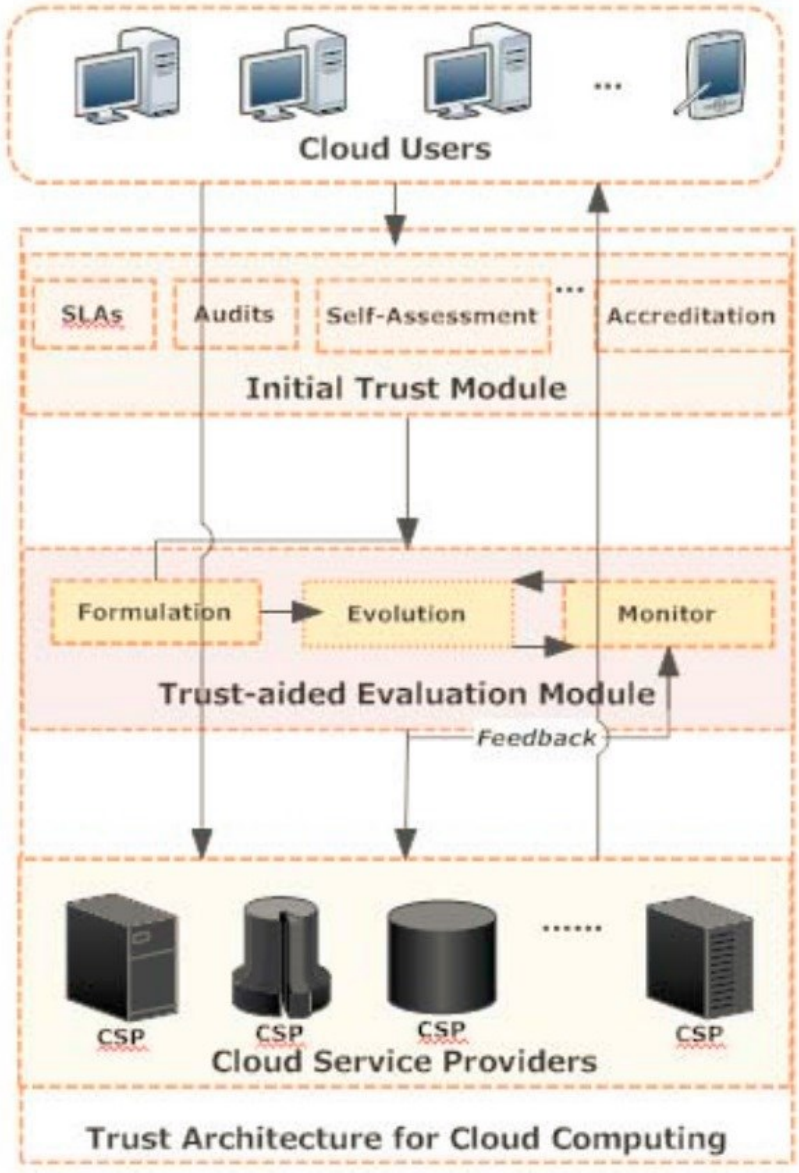

Figure 4.34. Trust service architecture for cloud computing [122]

- **Habib et al. (2014) [123]**

The authors proposed a trust model named Multi-Faceted trust model which is a SLA based Trust Management (TM) system architecture for a cloud computing marketplace. Figure 4.39 shows the system architecture of this model. As we can see in the figure 4.39 CSPs register through the Registration Manager (RM) to be able to act as sellers in a cloud marketplace. They have to provide system/service specifications related to the service delivery models (e.g., SaaS, PaaS, IaaS) they offer and fill in the CAI questionnaire as a part of cloud marketplace policy. The RM forwards the answers of the questionnaire and system/service description to the CAIQ engine and TI (Trust Information) respectively for further processing. The CAIQ engine CSPs to fill in the CAI questionnaire by providing an intuitive graphical interface through the RM [123].

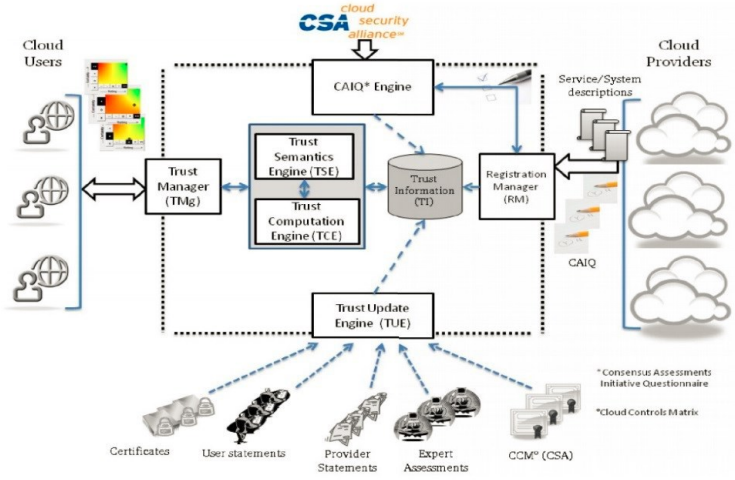

Figure 4.35. Multi-faceted architecture Overview [123]

The questionnaire helps CSPs to represent their competencies to the potential users with respect to different attributes. The questions are designed to be answered in ’yes’ or ’no’. All the answers are stored in the TI for further processing. The TMg allows cloud users to specify their requirements and opinions when accessing the trust score of CSPs. It provides a web-based front end to the users for specifying their requirements. Based on the requirements, the TMg provides the trust score of CSPs by using the Trust Semantic Engine (TSE) and Trust Computation Engine

(TCE). By default, users receive the trust value of a CSP based on the self-assessment using the CAIQ and assessment of cloud-based services/systems. Otherwise, users can specify their own preferences (e.g., security and performance are preferred over CC support), according to their business policy and requirements, to get a customized trust value of the CSPs. The TSE models which configuration of PLTs are considered to be the expected (trustworthy) behavior of a CSP in terms of a specific attribute. A default configuration of PLTs should be based on the CAIQ answers stored in the repository (TI). The TSE should be able to convert every trust relevant information into PLTs. The TCE consist of operations related to the operators (AND, OR, NOT, FUSION, CONSENSUS, DISCOUNT ING), used in PLTs to compute the corresponding trust values. The TCE is tightly coupled with the TSE to evaluate the PLTs and compute corresponding trust values. The trust values are archived in the TI repository after computation. The TUE allows to collect opinions from various sources and roots about the trustworthiness of CSPs [123].

- **Kanwal et al. (2014) [124]**

The authors proposed a trust evaluation model for Cloud federation to evaluate and establish bi-directional trust between home and foreign CSPs. The evaluation of trust is based on feedback (collected from registered CCs) and SLAs of CSPs. The trust evaluation model acts as a trusted third party that evaluates the trust of CSPs and provides the required trust credentials on receiving the trust requests from CSPs participating in federation. Figure shows the architecture of the model. The Registration Manager (RM) is responsible for registration of CSPs and the CCs that are consuming different services of these CSPs. During registration, the RM also collects standard SLAs of CSP which are later used by the SLA Manager for evaluation of trust score. Furthermore, each registered CSP submits its metadata to the RM. SLA Management (SM) is responsible to extract the QoP attributes from the provided SLA of CSP and evaluate the trust

score from these parameters. It includes three main sub modules namely; SLA Repository (SR) which manages the storage of SLAs collected from the registered CSPs at the time of registration, Parameter Extraction (PE) which retrieves the SLA from the SR and extracts the essential QoP attributes offered by the particular CSP and SLA based Trust Evaluation (STE) which receives The extracted parameters from PE module [124].

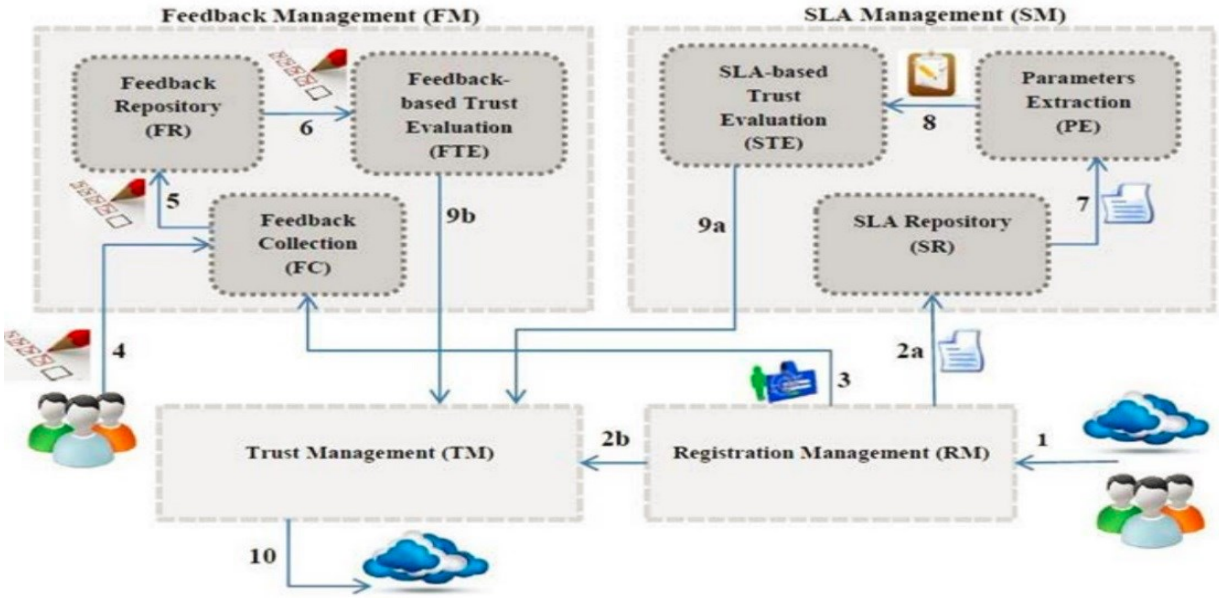

Figure 4.36. Overall workflow of Trust Evaluation Model [124]

The Feedback Management (FM) module receives the information from registration module and evaluates the trust of CSP based on the received feedback. It has three major sub-modules namely the Feedback Repository (FR) which collects the submitted feedback from Feedback Collection Manager and manages the storage of the feedback at backend database, Feedback based Trust Evaluation (FTE) which retrieves the feedback from FR and evaluates the trust score based on this feedback and Feedback Collection (FC) which is responsible for collecting feedback regarding security and privacy features supported by CSPs. The Trust Management module is responsible for receiving the trust requests from registered CSPs participating in federation with other CSPs; furthermore it verifies the trust requests and generates the trust

responses accordingly. In order to generate the trust response, the TM collects the evaluated trust scores and calculate total aggregate trust as follow [124]:

$$T_{final} = \frac{T_{SLA} + T_{feedback}}{2} \quad (4.9)$$

- **Marudhadevi et al. (2015) [125]**

The authors proposed a trust model named Trust mining model (TMM) [125] to identify trusted cloud services while negotiating an SLA. The challenge for the user is to monitor the services provided from the CSP and check if they meet the conditions mentioned in the agreement. To perform this, the user needs further information such as prior data or knowledge about what is happening on the CSP side, which can help him to better realize the effective QoS. The trust model evaluates the trust degree on the prior data obtained about the service at the time of the SLA. Then, this information is divided into multiple common attributes like the number of service denials, average response time, task success ratio and number of complaints registered by the users. Usually, attributes used to formulate any trust model can be either objective or subjective, while this work uses both types of values. In this way, advantages introduced with this approach are both for CSPs and end users. From one side, the CSP can monitor the performance and improve its services to establish better trust relations with the users. And from the other side, the CC can perceive as secure working with the CSP [125].

- **Pawar et al. (2012) [126]**

The authors proposed a trust model named SLA monitoring compliance which is an uncertainty model, which calculates trust values based on different parameters, namely SLA monitoring compliance, service provider ratings, and service provider behavior. More in detail, the SLA monitoring defines the opinion about a CSP from the established SLAs about its services. Each of them are provided with a single SLA that includes several common indicators, such as

CPU, memory, disk space usage, number of virtual machines. For each indicator of an SLA, a monitor evaluating the compliance/noncompliance of the indicator is provided. Then, CSP ratings are determined with the computation of all ratings, based on consensus and conjunction ratings. To calculate trust values, the model take into account features like belief, disbelief, uncertainty, and base rate [126].

4.3.1. Analyzing SLA Based Trust

Tables 4.7, 4.8, and 4.9 represent the analysis of SLA based trust models based on the selected assessment criteria defined in chapter 3. The assessment criteria and the cloud computing challenges that each trust feature address are indicated in the tables. In the tables ✖ means the trust model does not evaluate the specified assessment criteria and ✓ means that it does. Also, there are some assessment criteria that if the trust model evaluates them instead of having ✓ as an indicator, it is mentioned how it support that criteria such as attack resistance, trust evaluation, trust scope and trust aggregation.

Based on table 4.7, none of the selected SLA based trust models evaluates encryption that address lack of confidentiality challenge. In order to find the trust models that address lack of reliability, Habib (2014) is the only model that evaluates attack resistance assessment criteria under this category and Chakraborty (2012) is the only trust model that evaluates punishment of agreement violation.

Table 4.7. Evaluating SLA Based Trust Models – Part A

Challenges	Lack of Confidentiality	Lack of Reliability	
	Encryption	Attack resistance	Punishment of Agreement Violation
Alhamad (2010)	×	×	×
SLA Assurance	×	×	×
Chakraborty (2012)	×	×	✓
Jules (2014)	×	×	×
FL-FL	×	×	×
Yang (2013)	×	×	×
Habib (2014)	×	Sybil	×
Kanwal (2014)	×	×	×
TMM	×	×	×
SLA Monitoring Compliance	×	×	×

Table 4.8 analyzes trust models that evaluate trust based on lack of identity management and lack of privacy assessment criteria. Considering lack of identity management, Yang (2013) and Kanwal (2014) are the two trust models that both evaluate identity management and trust bootstrapping. In addition, FL-FL and SLA Monitoring Compliance trust models are the SLA based trust models that just evaluate trust bootstrapping trust feature. Therefore, based on the CC concern in case of lack of identity management they can choose the trust model that does their required evaluation. Table 4.8 also analyzes selected reputation based trust models based on three assessment criteria that address lack of privacy. None of the studied SLA based trust models evaluate transferability and heterogeneity support. In addition, among all of the studied SLA based trust models, just Yang (2013) and Kanwal (2014) evaluate security enabled trust.

Table 4.8. Evaluating SLA Based Trust Models – Part B

Challenges	Lack of Identity Management		Lack of Privacy		
	Identity Management	Trust Bootstrapping	Transferability	Heterogeneity	Security Enabled Trust
Alhamad (2010)	×	×	×	×	×
SLA Assurance	×	×	×	×	×
Chakraborty (2012)	×	×	×	×	×
Jules (2014)	×	×	×	×	×
FL-FL	×	✓	×	×	×
Yang (2013)	✓	✓	×	×	✓
Habib (2014)	×	×	×	×	×
Kanwal (2014)	✓	✓	×	×	✓
TMM	×	×	×	×	×
SLA Monitoring Compliance	×	✓	×	×	×

Table 4.9 analyzes trust models that evaluate trust based on lack of reputation, lack of SLA Support and lack of transparency. Considering lack of reputation, SLA Assurance, Jules (2014), Kanwal (2014), TMM, and SLA Monitoring Compliance evaluate trust based on outside in approach. While, Chakraborty (2012), SLA Based Trust and Yang (2013), Habib (2014) and SLA Monitoring Compliance are evaluating trust based on inside out approach and FL-FL is evaluating trust based on black box approach. In addition, Alhamad (2010), SLA Assurance, Jules (2014), and TMM are evaluating trust based on global trust and FL-FL and Kanwal (2014) trust models are evaluating local trust.

Table 4.9. Evaluating SLA Based Trust Models – Part C

Challenges	Lack of Reputation					Lack of SLA Support	Lack Of Transparency
	Trust evaluation	Trust scope	Reputation Enabled Trust	Trust Aggregation	Trust Evidence	SLA Verification	Transparency
Alhamad (2010)	Inside out	Global	✓	Decentralize	Indirect	×	×
SLA Assurance	Outside in	Global	✓	Centralize	Indirect	×	×
Chakraborty (2012)	Inside in	-	-	-	Direct	×	×
Jules (2014)	Outside in	Global	✓	Centralize	Direct	×	×
FL-FL	Black box	Local	✓	Decentralize	Direct	×	×
Yang (2013)	Inside out	-	-	-	Indirect	×	×
Habib (2014)	Inside out	-	-	-	Indirect	✓	×
Kanwal (2014)	Outside-in	Local	✓	Centralize	Indirect	×	×
TMM	Outside-in	Global	✓	Centralize	Direct	✓	×
SLA Monitoring Compliance	Inside-in	-	-	-	Direct	✓	×

Furthermore, Alhamad (2010), SLA Assurance, Jules (2014), TMM, FL-FL and Kanwal (2014) are evaluate reputation enabled trust. SLA Based Trust and FL-FL can evaluate decentralize trust models while SLA Assurance, Jules (2014), Kanwal (2014) and TMM evaluate centralize trust model. Finally, Chakraborty (2012), Jules (2014), FL-FL, TMM and SLA Monitoring Compliance are evaluating direct trust evidence, while others evaluating indirect trust. By considering lack of SLA support, just Habib (2014), TMM and SLA Monitoring Compliance have

SLA verification mechanism. In addition, regarding lack of transparency, none of the studied trust models evaluate transparency in trust management systems.

4.4. Domain Based Trust Models

As discussed in section 2.2.2, domain based trust model divides the cloud into a number of autonomous domains and provides trust evaluation both for within-domain and inter-domain interactions. In this section in order to evaluate domain based trust models some of the recent trust models have been studied and analyzed based on the assessment criteria discussed in chapter 3.

- **Wu et al. (2012) [127]**

The authors proposed a trust model named Fuzzy Reputation-Based Trust Model (FRTM) [127] which is based on fuzzy logic inferences. The main idea of this trust model is to use fuzzy logic inferences to handle uncertainty, fuzziness, and incomplete information in cloud trust reports. This model helps CCs determine the level of trust that is to be placed on any particular CSP [127].

The steps of the fuzzy reputation-based trust model are presented in figure 4.37. Such steps are as follows [127]:

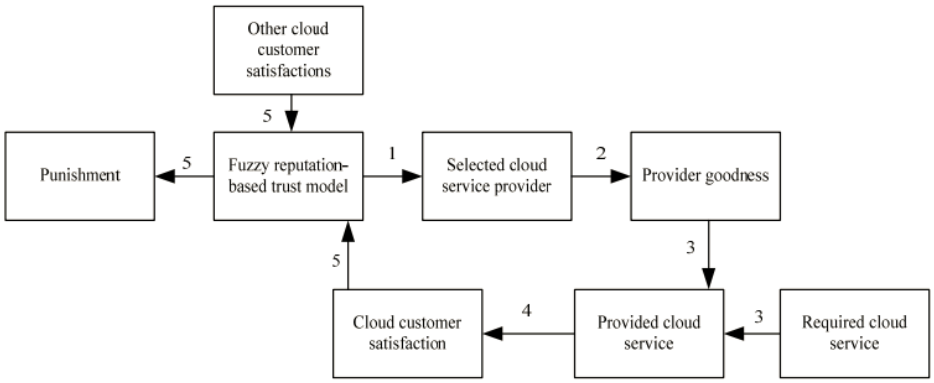

Figure 4.37. The steps of the fuzzy reputation-based trust model [127]

1) The trust model FRTM selects the CSP to have an interaction with.
2) Such providers has a perceived certain goodness (‘very low’, ‘low’, ‘medium’).

3) According to the required cloud service attributes and the provider goodness, the CSP provides a service to the CC.
4) The CC satisfaction is assessed based on the provided service in the previous step.
5) Finally, the punishment level is determined by the CC satisfaction with the received service, together with and other CC satisfactions.

Xu Wu has calculated the global reputation using the following formula [127]:

$$R_{p_i} = \alpha \sum_{C_j \in Q} \left[\frac{\vartheta_{C_j}}{\sum_{C_j \in Q} \vartheta_{C_j}} t_{p_i c_j} \right] + \beta \frac{f(p_i)}{l(Q)} = \alpha \frac{\sum_{C_j \in Q} \vartheta_{C_j} t_{p_i c_j}}{\sum_{C_j \in Q} \vartheta_{C_j}} + \beta \frac{f(p_i)}{l(Q)} \quad (4.10)$$

where Q is the set of the CCs with whom the CSP p_i has conducted interactions, $t_{p_i c_j}$ is the local trust score of the CSP p_i rated by c_i , and ϑ_{C_j} is the aggregation weight of $t_{p_i c_j}$. $f(p_i)$ denotes a punishment function of a ratio of the service satisfaction the CSP p_i is received during the recent time window. $l(Q)$ denotes the total numbers of interactions performed by the CSP p_i with all other CCs. α and β denote the normalized weight factors in order to give the CSP p_i a punishment level.

In this model five frequently used fuzzy inference rules are applied which are: 1) If the interaction data size is very high and the interaction time is new, then the aggregation weight is very large. 2) If the interaction data size is very low or the interaction time is very old, then the aggregation weight is small. 3) If a CSP's reputation is good and the interaction data size is high, then the aggregation weight is very large. 4) If a CSP's reputation is good and the interaction data size is low, then the aggregation weight is medium. 5) If a CSP's is bad, then the aggregation weight is very small [127].

- **Yang et al. (2010) [128]**

The authors proposed a collaborative trust model of firewall-through based on Cloud Computing combining the strength of domain-based trust model and the feature of the firewall. The model has many advantages: Firstly, there are different security policies for different domains. Secondly, the model considers the transaction context, the historical data of entity influences and the measurement of trust value dynamically. Thirdly, the trust model is compatible with the firewall and does not break the firewall's local control policies.

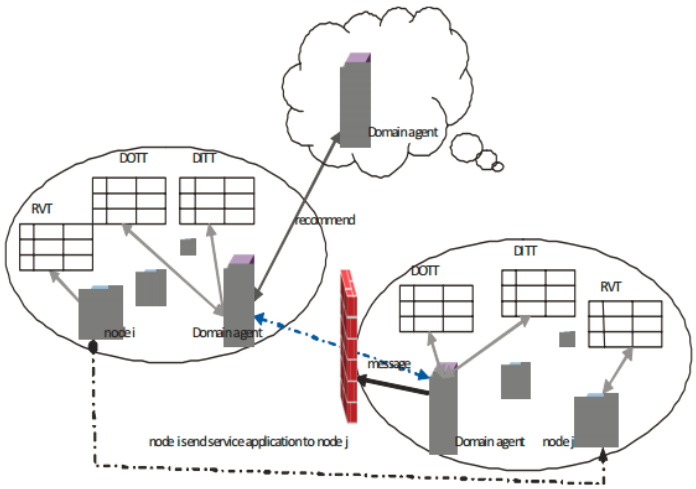

Figure 4.38. The structure of firewall-through based on cloud computing [128]

Figure 4.38 shows the structure of this model. This model divides the Cloud into different autonomous domains according to the physical address. The trust value of the nodes which have traded with other nodes within the domain are kept in Trust Table (TT). Each domain sets up a domain agent maintaining three tables; Domain inside trust table (DITT) stores trust values of all the nodes inside the domain, calculated by the average weight of recommended trust value of nodes which already had direct trading with this node; Domain outside trust table (DOTT) stores the value of the overall confidence for this domain to adjacent domains whose value is also average of nodes weight of trust evaluation. The risk-value table (RVT) is the definition of risk. When a user requests for resources, the domain agent broadcasts a message to the neighborhood agent in

inter-domain trust table, after successfully receive the message, neighborhood agents interact with the firewall and drive firewall by the introduction of agent sending the XML format message, then new members IP open up. If it is the first trading, service providers require the requester's digital signature. The approach of user releasing resources is similar with the process of requesting for resources. Set the time decay function to update the impact on trust value while taking into account the number of successful transactions [128].

- **Varadharajan et al. (2012) [129]**

The authors proposed a trust model named Trust Enhanced Security for Cloud Environments (TREASURE) [129] which is a property based attestation techniques for the cloud. Our model extends the node controller with the functionality of the CA to certify the behavior of the tenant virtual machines. Since the Node Controller is aware of the dynamic changes to the tenant virtual machine, it can ensure that the certified properties are satisfied by the tenant virtual machines. In this model authorization decisions in trusted platforms can be made not only by taking into account the identities and privileges of users and applications but also the state of the platform in terms of what types of applications and software components are running on that platform [129].

Figure 4.39 shows a generic cloud architecture. In TREASURE model the Certification Authority (CA) is used to certify some of the properties related to the communication between the tenant virtual machine (Attestation Provider (AP)) and the tenant customer (Attestation Requestor (AR)). In this case, the CA only certifies the basic security properties which are the assurance on the traffic originating from the AP and validation of the AP transactions. Spoofing is one of the fundamental challenges which makes it extremely difficult to deal with the attacks in the current environment. With the current TPM attestation techniques, there is no guarantee that the AP platform will not flood the AR with malicious traffic. The AR which needs to perform transaction

makes an attestation request to the tenant virtual machine (AP). The trusted platform provides an attestation report with the stored values that are measured during boot time. Now the AR can use the values in the report to validate the state of tenant virtual machine. In this model, the CA only certifies the basic security properties for communication between the AP and AR. The security properties are: assurance on the source address of the traffic originating from tenant virtual machine (AP) and validation on the traffic originating from tenant virtual machine (AP). If the traffic from the tenant server was found to be malicious then the AR can report the malicious traffic to the CA. Now CA can validate the AR report and take measures to deal with the possible compromise of the tenant virtual machine. After a successful transaction, the CA generates a signed report to the AP and AR with the transaction details [129].

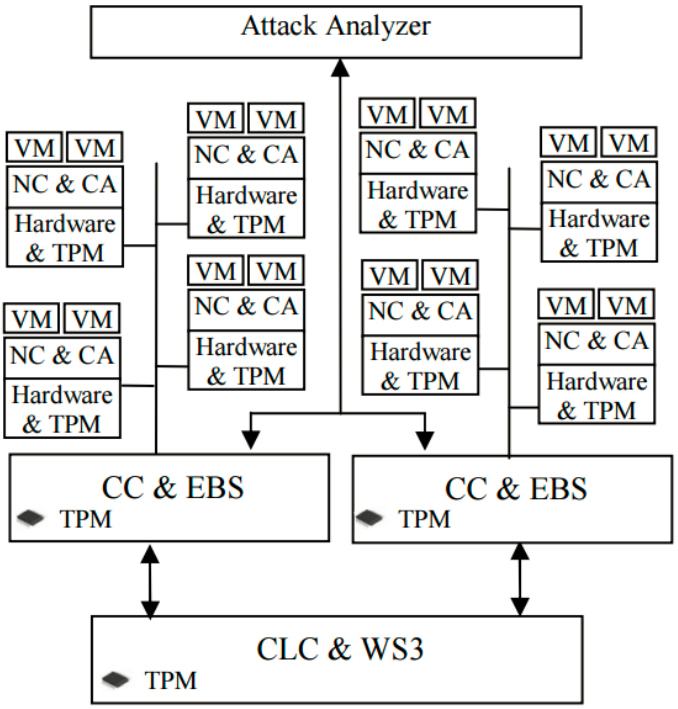

Figure 4.39. TREASURE cloud architecture [129]

- **Li et al. (2010) [130]**

The authors proposed a domain-based trust model to ensure the security and interoperability of cloud and cross-clouds environment and a security framework with an

independent trust management module on top of traditional security modules. They also put forward some trust based security strategies for the safety of both CCs and providers based on this security model. This model is domain-based and it differentiates two kinds of cloud roles: CC and CSP. Resources that belong to the same providers will attend the same trust domain. In each trust domain, it sets a trust agent to manage trust.

The propose security framework contains five layers. From bottom to up are physical security layer, system security layer, security abstract layer, trust management layer and security application layer. Physical security layer includes the electronics and physical measures to ensure the security of data storage or data transmission. System security layer provides security measures such as firewall, VPN (Virtual Private Network), SSL (Secure Sockets Layer), SSH (Secure Shell), Kerberos, Integrity Check, etc. Security abstract layer provides uniform interface for different kinds of security technologies. Trust management layer provides trust strategies to aid security judgements in cloud and heterogeneous cross-clouds environments and security application layer provides variety kinds of safe and trustable cloud applications [130].

Figure 4.40 shows the working mechanism in the new framework and the relationship between traditional security module and the trust management module. The basic process of making security decision in the proposed security model is [130]:

1) Security management layer passes the set of related cloud nodes to trust management layer,
2) Trust management layer evaluates the corresponding nodes' trust level and returns a credible subset according to its' local trust policies,
3) Security management layer made security judgement.

The basic procedure of trust-based authorization mechanism for CSPs contains the following six steps [130].

- Step 1: B's security management module transfers the identity of user A and list A to trust management module when job request of T is received from A.
- Step 2: B's trust management module judges whether user A is trust enough. If A is creditable, it goes to step 3 else rejects to trade with A directly.
- Step 3: B's trust management module searches for trust thresholds of other participants in list A and generates list T hold based on its local trust policy and the current security level.
- Step 4: B's trust management module computes each cooperator's trust value and evaluates if it is bigger than the corresponding threshold. It deletes creditable nodes from list A and adds them to list T.
- Step 5: B's trust management module returns list T to security management module.
- Step 6: B's security management module compares list T with list A. If the two are equal, B agrees to trade with A else it rejects.

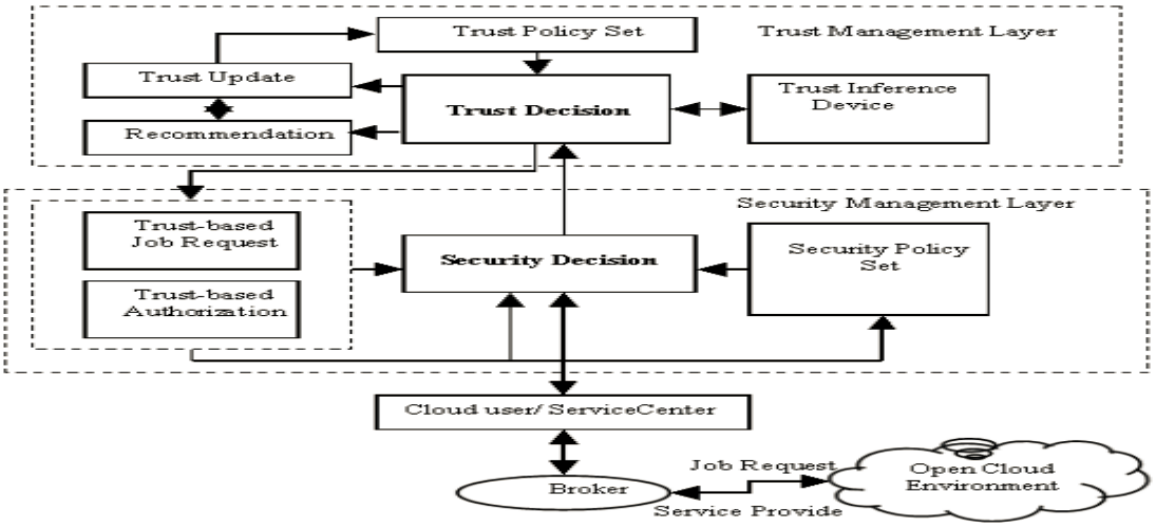

Figure 4.40. Working mechanism of the proposed security framework and relationship between security management module and trust management module [130]

- **Li et al. (2009) [131]**

The authors proposed a trust model which divides one CSP's resource nodes into the same domain and sets trust agent. It distinguishes two different roles CC and cloud server and designs different strategies for them. In our model, trust recommendation is treated as one type of cloud services just like computation or storage. The model achieves both identity authentication and behavior authentication. Figure 4.41 shows the basic realization framework of this model.

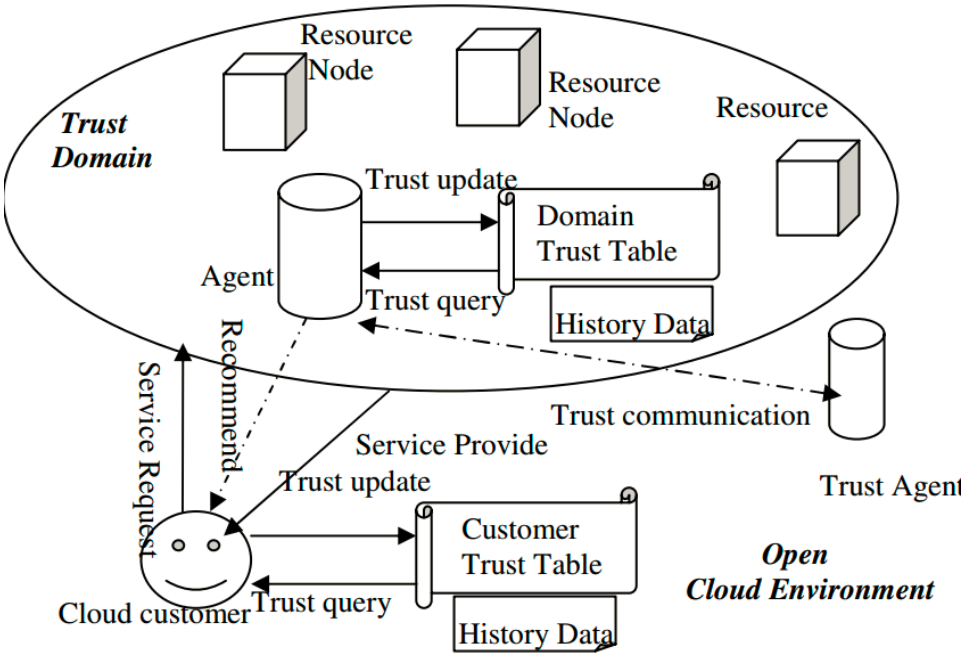

Figure 4.41. Realization framework of interoperability enhancing trust model [131]

In this model Safe transactions in cross-clouds environment are ensured by trust mechanism of which the key is trust decision. When CCs want to use cloud services, they have to make trust decision. Also when service providers want to cooperate, they have to make trust decision. Explicit trust decision needs a threshold. Unless trust value is bigger than or trust degree is higher than the threshold, entities will not continue their transaction. In our model, threshold is customizable and cloud entity or trust domain can independently set its threshold according to its current level of security. The general process of trust decision in our model is as follows: first of

all, to search corresponding value of special trading partner in local trust table (for CC is customer trust table and for provider is domain trust table). If there exists the value and it exceeds the threshold, entity will agree to continue the transaction else transaction will be suspended. If no corresponding record is found, entity will broadcast trust request within familiar domains. And original trust for counterparty will be calculated using the received recommendation trust and corresponding recommendation factor [131].

Two factors cause the update of trust: one is time and the other is re-evaluation of trust after each transaction. Time influence is continuous while transactions' are leaping. So the model adopts different strategies to evaluate them. It tends to use appropriate attenuation function to measure time influence. And in contrast it counts much on the evaluation of last time transaction rather than history cooperation data. Below is the different update policy for different cloud role [131].

4.4.1. Analyzing Domain Based Trust Models

Tables 4.10, 4.11, and 4.12 represent the analysis of domain based trust models based on the selected assessment criteria defined in chapter 3. The assessment criteria and the cloud computing challenges that each trust feature address are indicated in the tables. In the tables × means the trust model does not evaluate the specified assessment criteria and ✓ means that it does. Also, there are some assessment criteria that if the trust model evaluates them instead of having ✓ as an indicator, it is mentioned how it support that criteria such as attack resistance, trust evaluation, trust scope and trust aggregation.

Based on table 4.10, none of the selected domain based trust models evaluates encryption that address lack of confidentiality challenge. In order to find the trust models that address lack of

reliability, TREASURE is the only model that evaluates attack resistance assessment criteria under this category and FRTM is the only trust model that evaluates punishment of agreement violation.

Table 4.10. Evaluating Domain Based Trust Models – Part A

Challenges	Lack of Confidentiality	Lack of Reliability	
	Encryption	Attack resistance	Punishment of Agreement Violence
FRTM	×	×	✓
Yang (2010)	×	×	×
TREASURE	×	Spoofing / malicious traffic	×
Li (2010)	×	×	×
Li (2010)	×	×	×

Table 4.11 analyzes trust models that evaluate trust based on lack of identity management and lack of privacy assessment criteria. Considering lack of identity management, TREASURE and Li (2010) are the two trust models that evaluate identity management.

Table 4.11. Evaluating Domain Based Trust Models – Part B

Challenges	Lack of Identity Management		Lack of Privacy		
	Identity Management	Trust Bootstrapping	Transferability	Heterogeneity	Security Enabled Trust
FRTM	×	✓	✓	✓	×
Yang (2010)	×	×	✓	✓	✓
TREASURE	✓	×	×	✓	✓
Li (2010)	✓	×	×	✓	✓
Li (2010)	×	×	×	✓	×

In addition, FRTM evaluate trust bootstrapping trust feature. Therefore, based on the CC concern in case of lack of identity management they can choose the trust model that does their required evaluation. Table 4.11 also analyzes selected domain based trust models based on three assessment criteria that address lack of privacy. FRTM and [128] evaluate transferability. All of

the studied domain based trust models evaluate heterogeneity support. In addition, among all of the studied domain based trust models, just Yang (2010), TREASURE and Li (2010) evaluate security enabled trust.

Table 4.12 analyzes trust models that evaluate trust based on lack of reputation, lack of SLA Support and lack of transparency. Considering lack of reputation, Firewall Through based Trust Model and Li (2009) evaluate trust based on outside in approach. While, FRTM is evaluating trust based on black box approach and TREASURE and Li (2010) evaluate trust based on inside out approach. In addition, FRTM and Li (2009) are evaluating trust based on global trust. Furthermore, FRTM, and Li (2009) are evaluate reputation enabled trust. Firewall through based trust model and Li (2009) can evaluate decentralize trust models while FRTM evaluate centralize trust model. Finally, FRTM, TREASURE, Trust Management Model are evaluating direct trust evidence, however, firewall through trust model is evaluating indirect trust evidence while Li (2009) evaluate both direct and indirect trust. By considering lack of SLA support, just Li (2009), has SLA verification mechanism. In addition, regarding lack of transparency, none of the studied trust models evaluate transparency in trust management systems.

Table 4.12. Evaluating Domain Based Trust Models – Part C

Challenges	Lack of Reputation					Lack of SLA Support	Lack Of Transparency
	Trust evaluation	Trust Scope	Reputation Enabled Trust	Trust Aggregation	Trust Evidence	SLA Verification	Transparency
FRTM	Black Box	Global	✓	Centralize	Direct	×	×
Yang (2010)	Inside out	-	×	-	Indirect	×	×
TREASURE	Inside-out	-	×	-	Direct	×	×
Li (2010)	Inside-out	-	×	-	Direct	×	×
Li (2010)	Outside in	Global	✓	Decentralize	Direct/Indirect	✓	×

4.5. Platform Based Trust

As discussed in section 2.2.2, platform based trust models consists of policies that ensure applications are executing on platforms that meet a specified trust assurance level and evaluate the confidence of CCs on using cloud services lunch on a specific platform. In this section in order to evaluate platform based trust models some of the recent trust models have been studied and analyzed based on the assessment criteria discussed in chapter 3.

- **Li et al. (2013) [132]**

The authors proposed a trust model named Cloud-Trust [132] which is an adaptive trust model for efficiently evaluating the competence of a cloud service based on its multiple trust attributes. In Cloud-Trust, two kinds of adaptive modelling tools (rough set and Induced Ordered Weighted Averaging (IOWA) operator) are organically integrated and successfully applied to trust data mining and knowledge discovery. This model advocates an objective, attribute-based scheme in which users record their experiences with service providers rather than their subjective ratings. A user's experience with a service provider captures the difference between the requested service and the delivered service in terms of service specific attribute values. This can eliminate the issues of collaborative cheating or fraudulent practices prevalent in traditional reputation-based trust models. Also, this model uses rough set theory to adaptively conduct knowledge discovery of multiple trust attribute values (evidences). This feature of the model surpasses the limitations of traditional weighting methods for multiple attributes, in which weights are assigned subjectively [132]. The trust value in this model is calculated as follow [132]:

$$T_t(N_i) = r_t \times \{\varpi_1, \varpi_2, \dots, \varpi_k, \dots, \varpi_m\}^T \tag{4.11}$$

where N_i is ith service in the cloud. ϖ_k is the weight which assigned to this normalized evidence and $\varpi_k \in [0,1], \sum_{k=1}^{m} \varpi_k = 1$. $r_t = (r_{t1}, r_{t2}, \ldots, r_{tk}, \ldots, r_{tm})$ is a sample of trust degree measurement, and it is an m-dimensional vector and r_{tk} is [132]:

$$r_{tk} = 1 - \frac{x_{tk} - \min(x_{tk})}{\max(x_{tk}) - \min(x_{tk})} \tag{4.12}$$

where x_{tk} is the number of illegal connections, the number of scanning of important ports, the number of denial services and the average response time.

- **Raju et al. (2014) [133]**

The authors proposed a trust model named Build Trust on Cloud (BTC) [133] which is a model to build trust on the CSP from the consumer's perspective. This model is built by considering two issues i.e. security and lack of transparency. In this model whenever a cloud Virtual Machine (VM) is configured, a sensor is assigned and dedicated to it. Each sensor identifies and collects the events that are commonly suspicious. To by using anomaly based or any hybrid based detection technique. Each sensor identifies sets of patterns that are deviating from the normal behavior and reports to the Master Module (MM) at the VMM level as shown in Figure 4.42.

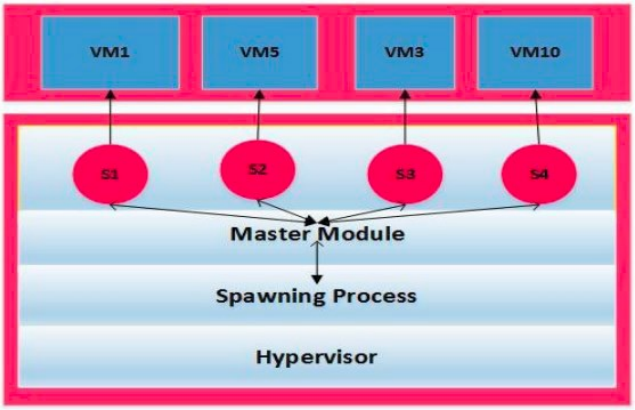

Figure 4.42. Sensor based malware detection [133]

Then based on the type of the abnormal events reported, MM can invoke and spawn the appropriate processes and get them applied on the compromised VM. The incident can be

confirmed after correlating all the spawned process results. After its conformance, it is more suggestible to do live migration or a snapshot of the VM to increase the reliability of the evidence collection [133].

In order to solve lack of transparency when an instance requested for some resource the scheduling algorithm checks for the resource that has not allotted to any other VM recently. If that type of resource is not available then it checks for the resource previously used by the instance holding less sensitive data by starting checking from the same zone to nearby zones. For calculating sensitivity, it is used the Natural Language Processing to perform filtering at various levels. The outputs have to be properly linked by applying item response theory. If the above two options have not satisfied then the less trusted VM will be migrated and then allotted to the current requested VM (If requested VM priority is high). While performing the migration the underlying instructions of the function will also look at the performance issues. Once the hypervisor receives the request from virtual machine VM_{α} then its trust levels will be identified. VM_{α} has been assigned with a resource which was previously used by another instance (VM_{β}). Then, its sensitivity of the operations and data used by it (instance β) are identified. Once the two parameters are identified (Trust and sensitivity) then whether to overwrite or not can be decided [133].

- **Habib et al. (2011) [134]**

The authors introduced CertainTrust which provides an architecture of a multi-faceted TM system for cloud marketplaces providing means to efficiently differentiate between a good and a poor quality (beyond the performance issues) providers. The system is designed to provide a customized trust score of a provider based on the attributes selected by the CCs. Moreover, the system aims to provide trust scores of the CSPs based on trustworthy behavior of the underlying systems and the service providers' answers to the CSA CAI questionnaire. Figure 4.43 shows this

model architecture. In this model CSPs register through the Registration Manager (RM) to be able to act as sellers in a cloud marketplace. The RM forwards the answers of the questionnaire and system/service description to the CAIQ engine and Trust Information (TI) respectively for further processing [134].

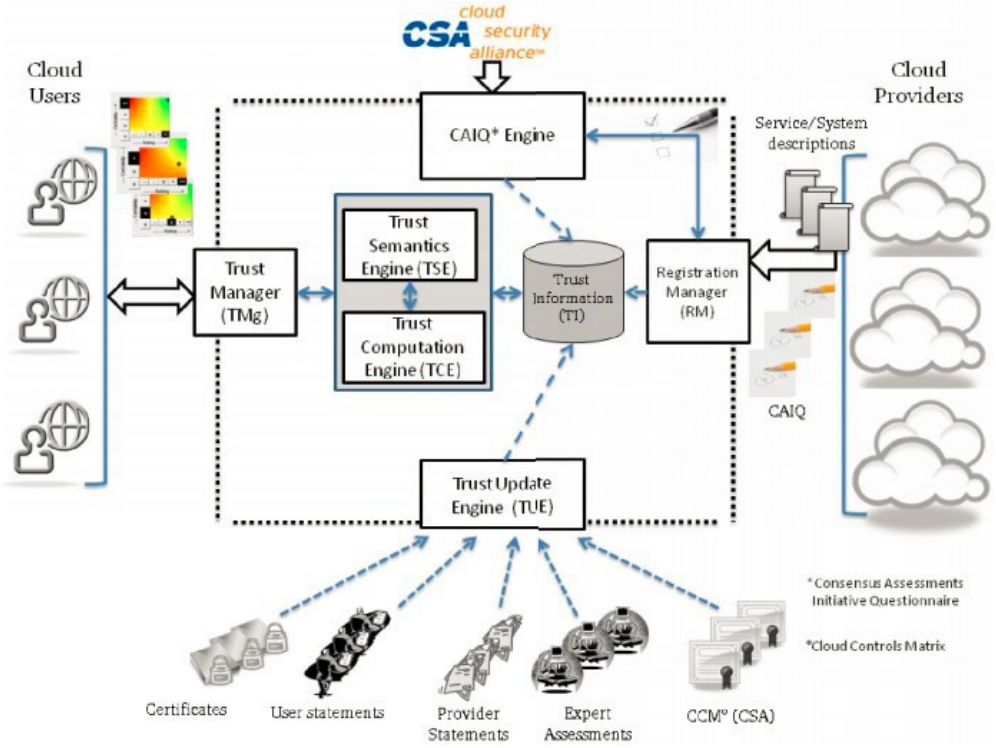

Figure 4.43. Trust model architecture [134]

The CAIQ engine allows CSPs to fill in the CAI questionnaire by providing an intuitive graphical interface through the RM. The TMg allows cloud users to specify their requirements and opinions when accessing the trust score of CSPs. It provides a web-based front end to the users for specifying their requirements. The TSE models are considered to be the expected (trustworthy) behavior of a CSP in terms of a specific attribute. The TCE consist of operations related to the operators (AND, OR, NOT, FUSION, CONSENSUS, DISCOUNT ING), used in PLTs to compute the corresponding trust values. The Trust Update Engine (TUE) allows to collect opinions from various sources and roots about the trustworthiness of CSPs [134].

- **Zhexuan et al. (2010) [135]**

The authors have taken a look at the security issues SaaS might create due to the unrestricted access on user data given to the remotely installed software. The authors have presented a mechanism to separate software from data so that it is possible to create a trusted binding between them. The mechanism introduced involves four parties namely the resource provider, software provider, data provider and the coordinator. The resource provider hosts both data and software and provides the platform to execute the software on data. The software provider and data provider are the owners of the software and data respectively. The coordinator brings the other parties together while providing the ancillary services such as searching for resources and providing an interface to execute the application on the data [135].

The operation of the model is as follows: Software provider and data provider upload their resources to the resource provider. These resources will be encrypted before stored and the key will be stored in the accountability vault module of the system. A data provider searches for and finds the required software through a coordinator and then runs the software on the data uploaded to the resource provider‘s site. Once the execution has started an execution reference ID is generated and given to the data provider. When the execution of the software is over, the results are produced only on the data provider‘s interface which can be viewed, printed or downloaded. Data provider will then pay for the service that will be split between the software provider and resource provider. An operation log has been created and posted to the software provider without disclosing the data provider‘s identity or the content on which software was run. This helps the software provider know that his software has been used and the duration of use [135].

Even though the authors claim that this model separates the software and data, there is no assurance that the software cannot make a copy while the data is being processed as only the

algorithm or description of the software is provided to the data owner. Without the source code, there is no assurance that the code will not contain any malicious code hidden inside. Also, since the software runs on data owner‘s rights and privileges, the software would have complete control over data. This is a security threat and the audit trail even if it is available, will not detect any security breaches [135].

The authors do not address the question of trust on the proposed platform as this would be another application or service hosted on the cloud. Both application providers and data providers need some kind of better assurance as now they are entrusting their data and software to a third party software.

- **Sato et al. (2010) [136]**

The authors proposed a trust model named Security Aware Cloud which is a trust model of cloud security in terms of social security [136]. The authors have identified and named the specific security issue as social insecurity problem and tried to handle it using a three pronged approach. They have subdivided the social insecurity problem in to three sub areas, namely; multiple stakeholder problem, open space security problem and mission critical data handling problem.

The multiple stakeholder problem addresses the security issues created due to the multiple parties interacting in the cloud system. As per the authors, three parties can be clearly identified. They are namely, the client, the CSPs and third parties that include rivals and stakeholders in business. The client delegates some of the administration/operations to CSPs under a SLA. Even if the client would like to have the same type of policies that it would apply if the resources were hosted on site on the delegated resources, the provider‘s policy may differ from that of the client. The providers are bound only by the SLA signed between the parties. The SLA plays the role of

glue between the policies. Also the authors opine that once the data is put in the cloud it is open for access by third parties once authenticated by the CSP [136].

The open space security problem addresses the issue of loss of control on where the data is stored and how they are physically managed once control of data is delegated to the CSP. They advise to encrypt the data before transferring, converting the data security problem to a key management problem as now the keys used for encryption/decryption must be handled properly.

The mission critical data handling problem looks at the issue of delegating the control of mission critical data to a service provider. They advise not to delegate control of this data but to keep them in a private cloud in a hybrid setup, where the organization have unhindered control. However setting up of a private cloud may not be an option to small and medium sized organizations due to the high costs involved. Hence enhancement of security of the public cloud is the only option to serve everybody.

Authors have developed a trust model named “cloud trust model” to address the problems raised above. Two more trust layers have been added to the conventional trust architecture. These layers have been named as the internal trust layer and the contracted trust layer. The Internal trust layer acts as the platform to build the entire trust architecture. It is installed in the in house facilities and hence under the control of the local administration. ID and key management are handled under the internal trust. Also any data that is considered critical or needs extra security must be stored under this layer.

Contracted trust has been defined as the trust enforced by an agreement. A CSP places his trust upon the client, based on the contract that is made up of three documents known as, Service Policy/Service Practice Statement (SP/SPS), Id Policy/Id Practice Statement (IdP/IdPS) and the contract. Level of trust required can be negotiated by parties depending on the level of security

needed for the data. A cloud system thus installed is called a secure cloud by the authors. Figure 4.44 shows the security our cloud proposed by this thesis. An organization must be aware of levels of services. On the other hand, a CSP can believe the assertions provided by an organization in a given level. By evaluating the quality of services by the contract and related documents, an organization can make decisions whether to delegate services to CSPs depending on the qualities stated in the contract. Data can be placed out of the organization depending on its criticality and cost [136].

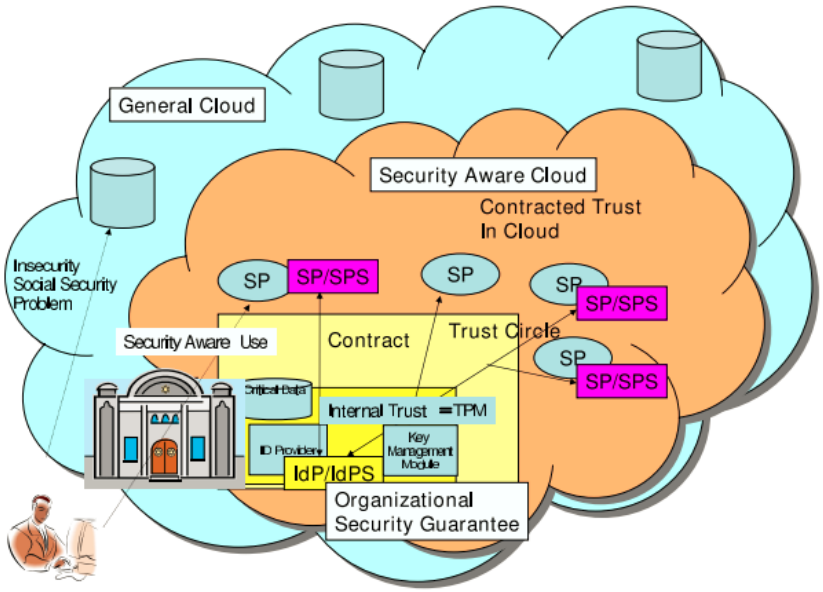

Figure 4.44. Security aware cloud [136]

- **Ko et al. (2011) [137]**

The authors proposed a trust model named TrustCloud which is a detective rather than preventive approaches to increasing accountability. The TrustCloud framework is independent of virtual or physical environments. This model makes accountability more achievable. It focuses on integrity and accountability of data stored in the cloud, so it requires a file-centric perspective, on top of the usual system-centric perspective for logging. Logs range from system-level logs to workflow-level audit trail transactional logs. Figure 4.45 shows the abstraction layers for the type

of logs needed for an accountable cloud. The lowest TrustCloud layer is the system layer. The system layer tracks data containers by performing file centric logging within operating systems (OS), file systems and cloud's internal network. The data layer supports the data abstraction and facilitates data-centric logging through provenance logger and consistency logger. The workflow layer focuses on audit trails and audit related data found in the software services in the cloud [137].

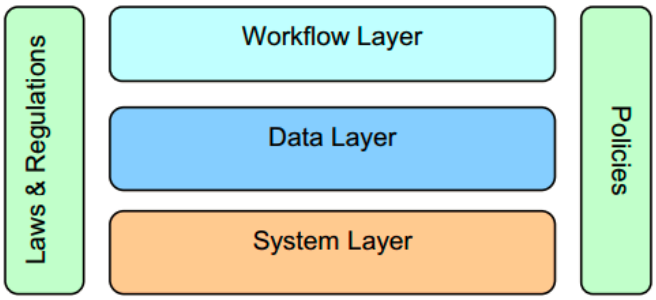

Figure 4.45. Abstraction layers of accountability in cloud computing [137]

- **Almorsy et al. (2011) [138]**

The authors proposed a trust model named Security Management Framework Based on Aligning (FISMA), which is a new cloud security management framework based on aligning the FISMA standard to fit with the cloud computing model, enabling CSPs and consumers to be security certified. This framework is based on improving collaboration between CSPs, service providers and service consumers in managing the security of the cloud platform and the hosted services [138].

The model framework architecture consists of three main layers: a management layer, an enforcement layer, and a feedback layer. Management layer which is responsible for capturing security specifications of the CPs, SPs, and CCs. It consists of: (a) The security categorization service used by the hosted services' tenants to specify security categorization of their information maintained by the cloud services; (b) The collaborative risk assessment service where all the cloud platform stakeholders participate in the risk assessment process with the knowledge they possess.

(c) The security controls manager service is used to register security controls, their mappings to the FISMA security controls' templates, and their log files structure and locations. (d) The security metrics manager service is used by the cloud stakeholders to register security metrics they need to measure about the platform security. (e) The multi-tenant security plan (SLA) viewer service is used to reflect the tenant security agreement. This shows the tenant-service security categorization, vulnerabilities, threats, risks, the selected mitigation controls and the required metrics. (f) The multi-tenant security status viewer. This reflects the current values of the security metrics and their trends. Enforcement layer which is responsible for security planning and security controls selection based on the identified risks. The selected security controls are documented in the security management plan. The implementation service then uses this plan for maintaining security control configuration parameters and the mapping of such parameters to the corresponding security controls. Feedback layer which has two key services: the monitoring service which is responsible for collecting measures defined in the security metrics manager and storing it in the security management repository to be used by the analysis service and by the multi-tenant security status reporting service. The analysis service analyses the collected measures to make sure that the system is operating within the defined boundaries for each metric. If there is a deviation from the predefined limits, the analysis service will give alerts to update the current configurations [138].

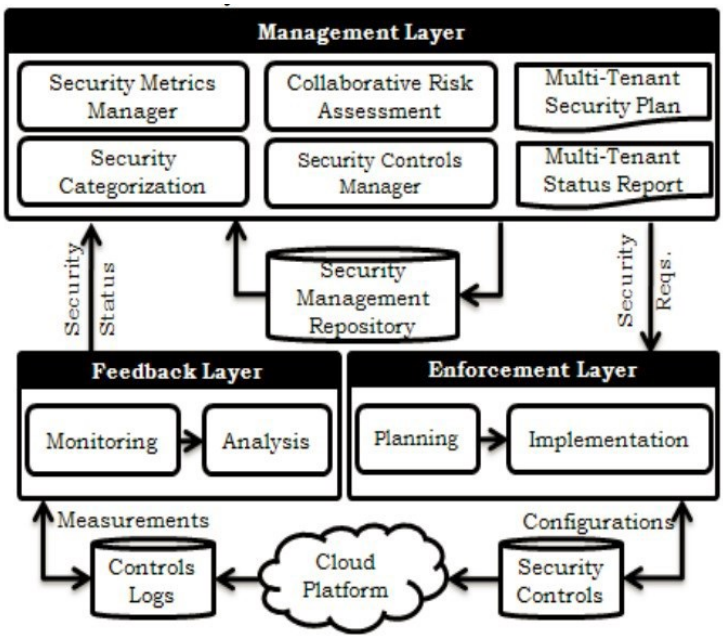

Figure 4.46. The collaboration-based framework architecture [138]

4.5.1. Analyzing Platform Based Trust

Tables 4.13, 4.14, and 4.15 represent the analysis of domain based trust models based on the selected assessment criteria defined in chapter 3. The assessment criteria and the cloud computing challenges that each trust feature address are indicated in the tables. In the tables × means the trust model does not evaluate the specified assessment criteria and ✓ means that it does. Also, there are some assessment criteria that if the trust model evaluates them instead of having ✓ as an indicator, it is mentioned how it support that criteria such as attack resistance, trust evaluation, trust scope and trust aggregation.

Based on table 4.13, none of the selected platform based trust models evaluates encryption that address lack of confidentiality challenge. In order to find the trust models that address lack of reliability, Habib (2011) and FISMA are the two trust models that evaluate attack resistance assessment criteria under this category and in addition none of the trust models evaluates punishment of agreement violation.

Table 4.13. Evaluating Platform Based Trust Models – Part A

Challenges	Lack of Confidentiality	Lack of Reliability	
	Encryption	Attack resistance	Punishment of Agreement Violation
Cloud-Trust	×	×	×
BTC	×	×	×
Habib (2011)	×	Sybil attack	×
Zhexuan (2010)	✓	×	×
Sato (2010)	✓	×	×
TrustCloud	×	×	×
FISMA	×	Malicious user	×

Table 4.14 analyzes trust models that evaluate trust based on lack of identity management and lack of privacy assessment criteria. Considering lack of identity management, BTC and Zhexuan, Sato and TrustCloud are the trust models that evaluate identity management.

Table 4.14. Evaluating Platform Based Trust Models – Part B

Challenges	Lack of Identity Management		Lack of Privacy		
Assessment criteria / trust Models	Identity Management	Trust Bootstrapping	Transferability	Heterogeneity	Security Enabled Trust
Cloud-Trust	×	✓	×	×	✓
BTC	✓	×	✓	✓	✓
Habib (2011)	×	×	×	×	✓
Zhexuan (2010)	✓	×	×	×	✓
Sato (2010)	✓	×	×	×	✓
TrustCloud	✓	✓	×	×	✓
FISMA	×	×	×	×	✓

In addition, Li (2013) and TrustCloud evaluate trust bootstrapping trust feature. So, TrustCloud evaluate both identity management and trust bootstrapping assessment criteria.

Therefore, based on the CC concern in case of lack of identity management they can choose the trust model that does their required evaluation. Table 4.14 also analyzes selected platform based trust models based on three assessment criteria that address lack of privacy. None of the trust models evaluate transferability. Just BTC evaluate heterogeneity support. In addition, all of the studied platform based trust models evaluate security enabled trust.

Table 4.15 analyzes trust models that evaluate trust based on lack of trust on CSPs, lack of SLA Support and lack of transparency. Considering lack of reputation, just Li (2013) and FISMA are evaluating reputation enabled trust and both of them evaluate global trust and evaluate trust based on outside in approach. By considering lack of SLA support, all of the studied platform based trust models have SLA verification mechanism except Zhexuan (2010) and TrustCloud. In addition, regarding lack of transparency, BTC and Li (2013) evaluate transparency in trust management systems.

Table 4.15. Evaluating Platform Based Trust Models – Part C

Challenges	Lack of Trust on CSPs					Lack of SLA Support	Lack Of Transparency
	Trust evaluation	Local/Global	Reputation Enabled Trust	Trust Aggregation	Trust Evidence	SLA Verification	Transparency
Cloud-Trust	Outside in	Global	✓	Centralize	Direct	✓	×
BTC	Inside out	-	×	-	Direct	✓	✓
Habib (2011)	Inside out	-	-	-	Direct	✓	✓
Zhexuan (2010)	Inside out	-	×	-	Indirect	×	×
Sato (2010)	Inside out	-	×	-	Direct	✓	×
TrustCloud	Inside out	-	×	-	Direct	×	×
FISMA	Outside in	Global	✓	Centralize	Direct	✓	×

CHAPTER 5. DISCUSSION

Cloud computing created many new opportunities for organizations across the world. For example, with cloud computing, cloud services are available at affordable prices. Cloud services help CCs to access everything from software to operating platforms without investing in hardware or software. Also, for storing a massive amount of data, cloud computing is very useful. CCs can easily add or subtract storage at any time, with no financial penalties. There are a wide variety of opportunities that cloud computing offers to the enterprises. Despite all these benefits, there are a number of concerns that make the adoption of cloud computing a little challenging and hard for the enterprises to find the best CSP that satisfies their concerns. In order to help the enterprises select the best CSP, the researchers have develop some trust models. Trust models are tools to evaluate the trustworthiness of the cloud services and cloud service providers. The output of a trust model is a score that determines how well a trust management system satisfies a set of parameters. Each trust model can just evaluate trust based on a specific set of criteria. The enterprises should find the trust model that can evaluate the criteria that address their concerns. For example, if one of the enterprises concerns about the confidentiality of the data stored in cloud, then it should find a trust model that can evaluate encryption in trust management systems. However, there are many trust models and it is very hard for the enterprises to choose the one that best evaluate their considered criteria. This thesis aims to help the enterprises find a good trust model. Chapter two has studied the concerns that the enterprises have to adopt cloud computing and categorized them into seven challenges. In chapter three, fourteen assessment criteria have been selected to address each of these challenges. The enterprises need to determine the services they want to use in cloud and the characteristics of those services. For example, one enterprise might want a cloud service that is very reliable and is always available, while another enterprise is just concern about a good

contract with the CSP that also include some rights for the cloud consumer. Therefore, based on the challenges mentioned in chapter two, the former enterprise is concerned about lack of reliability while the latter is concerned about lack of SLA. After detecting the challenges, the enterprise can check the assessment criteria for each challenge in chapter three. They need to know which of those criteria under their selected challenges is more important for them since one trust model may not evaluate all of those assessment criteria. Finally, they can review the trust models studied in chapter four to find the best trust model that help them find the best CSP. The tables in chapter four help a lot in comparing and finding the best trust model. After the enterprise choose the right trust model, it can run that trust model on the trust management system of a specific cloud service, to find the goodness of the services offered. In order to make the process a little more clear, two scenarios is discussed below.

The first scenario is about a company provides a drive-through grocery service in which customers shop online and drive to neighborhood pickup points where their purchases are loaded into their cars. This service is hosted different grocery stores in different part of world. It needs up to date knowledge of what is selling in each store so that it can order the right inventory for the next day. Also, it needs to have an up to date knowledge of the amount of stuff stocked at each store so that it does not let the customers buy the items that are not available in a specific store. In addition, it has a supply-chain management application used for forecasting, demand planning, and inventory control. Therefore, if a store in Italy decide to run a special on tomatoes that suddenly become available due to sunny weather, the company's IT organization can ensure that their software orders the extra tomatoes and that drivers deliver them for the following day. Or, if a heavy snowstorm causes a store to close, the company can halt the following day's orders so the store does not have to pay for unneeded merchandise.

This company needs a high performance storage system to help them complete its daily calculation of products sold, then reorder products needed for the following day by its 6:00 P.M. This company is willing to migrate it's entire on-premises datacenter to the cloud. It is looking for a public cloud environment that provides compute, storage, networking, and other services for creating and hosting applications in datacenters. It needs a high performance reliable system that has a very low failure rate. Also, the system should be very secure since the customers will share their credit card information to submit their purchase. In addition, since it is a very competitive market, they do not want any of their competitors learn about their forecasting application. Therefore, considering the challenges that discussed in chapter 3, this company cares about three challenges. One of them is lack of reliability. The system should be reliable and attack resistance and no attack should be able to affect the system performance. Attack resistance is the assessment criteria that this company want the trust model evaluate in cloud system. The second challenge that this company cares about is lack of confidentiality. Both the customer information and the result of the supply chain management application should be store securely. Therefore, they choose encryption as the assessment criteria. Third challenge is lack of privacy. The cloud system should have a high privacy protection against the company's applications and data. They found that transferability is the assessment criteria that they would like the trust model evaluate for them. There are a number of CSPs that can serve this company but the managers of the company are very indecisive on choosing the best cloud service providers. They need to find the best trust model that best analyzes the cloud services based on their reliability, privacy and confidentiality concerns. By referring to the tables provided in chapter 4, the managers can find the trust model that help them to evaluate their criteria in cloud services. For instance, based on the criteria this company has, TREASURE trust model best satisfies their requirements. They can run this trust model on

trust management systems and the returned value would show the goodness of the cloud service and CSP.

The next scenario is about a company that has a solution for establishing communication between clinicians, patients, and hospital administrators. This solution provides medical imaging, medical diagnostics, patient monitoring systems, performance improvement, drug discovery, and biopharmaceutical manufacturing technologies for the customers from all around the world. For instance, with this solution, doctors working in different locations can all look at the same diagnostic images simultaneously and collaborate with each other easily to find the best treatment plans for a patient. This technology reduces information silos and help doctors collaborate more easily and quickly on treatment plans.

This company needs to provide a secure, centralized, real-time access to diagnostic scans and reports, therefore, it requires a flexible, highly scalable platform that could deliver a wide range of solutions and services to help imagine new ways to diagnose and treat cancer, heart disease, neurological diseases and other conditions earlier, more cost-effectively, and more efficiently. The managers of this company decided to migrate their software to the cloud. However, they have some concern regarding the service they get. The first one is that they are concerned about lack of identity management challenge in cloud computing. Since the patient data are confidential and just doctors should have access to them, they like the cloud service provide an effective identity management system. The second challenge that they are concerned about is lack of reliability. The system should be available at any time in order to provide a very effective and fast diagnostic services and it should be resistance to any attack. The third challenge in cloud computing that this company is concerned about is lack of confidentiality. This company expects that all the data that they will store in the cloud be secure and confidential. In addition, the

managers of this company are a little conservative so they would like to have a good contract with the CSP, therefore, how the SLA is written is very important for them. Based on this discussion, this company can choose one of the trust models that we discussed in chapter four to help them evaluate the CSPs. Based on the tables provided in chapter four, MTCEM is a good trust model that can help this company to find the best CSP. Furthermore, if the company wants to know how trustworthy the CSP is, they can also consider this criteria and find a trust model that satisfy this challenge as well.

As it is explained here, the tables in chapter four help enterprises find a trust model that best evaluate the trust management systems for them. Since each trust model evaluates a limited number of criteria, an enterprise might need to use multiple trust models to evaluate the trust management systems based on all of the criteria that it considers.

CHAPTER 6. CONCLUSION

In recent years, cloud computing has become a vibrant and rapidly expanding area of research and development. In today's competitive environment, the service dynamism, elasticity, and choices offered by highly scalable cloud computing technology are too attractive for enterprises to ignore. These opportunities, however, don't come without challenges. Because CSP controls the data, enterprises are concerned about different challenges on data confidentiality, privacy, integrity and availability for CCs. Today, the problem of trusting cloud computing is a paramount concern for most enterprises in such a way that trust is widely regarded as one of the top obstacles for the adoption and growth of cloud computing. In order to evaluate trust management systems, trust models have been developed. However, each developed trust model evaluates limited number of assessment criteria and it is hard for enterprises to use these trust models in their decision making process. In this work, a comprehensive survey has been presented that is, to the best of our knowledge, the first in cloud computing to focus on the evaluation of trust management of services and trust models usefulness, comprehensively. Six trust management challenges have been recognized to overcome trust issues in cloud computing in Chapter 2. Fourteen assessment criteria have been selected for trust based on existing trust models and then they have been categorized into six introduced trust challenges in Chapter 3. Fifty three trust models have been overviewed and compared in Chapter 4 to determine related assessment criteria for each of them. A generic analytical framework has been proposed in Chapter 4 that can be used to compare different trust models based on a set of assessment criteria. Chapter 5 provides a discussion on how to use the suggested analytical framework. Along with the current research efforts, we encourage more insight and development of innovative solutions to address the various open research issues that we have identified in this work.

REFERENCES

[1] P. Mell and T.Grance. “The NIST definition of cloud computing,” National Institute of Standards and Technology, 2009.

[2] M. Firdhous, O. Ghazali and S. Hassan. “Trust Management in Cloud Computing: A Critical Review,” International Journal on Advances in ICT for Emerging Regions, Vol. 4 no. 2, pp. 24-36, 2011.

[3] J. Staten. “Is cloud computing ready for the enterprise?” Forrester Research, 2008.

[4] X. Li and L. Liu. "A Reputation-based Trust Model for Peer-to-Peer E-Commerce Communities," in IEEE International Conference on E-Commerce Technology, 2003, pp. 275-284.

[5] D. Zhou. "Security Issues in Ad-hoc Networks," The Handbook of Ad-hoc Wireless Networks Boca Raton, FL, USA: CRC Press, Inc, 2003, pp. 569 – 582.

[6] O. Malik. “Amazon S3 Storage Service Does Down, Still Not Up,” Internet: https://gigaom.com/2008/02/15/amazon-s3-service-goes-down/, Feb 15, 2008 [Jan. 24, 2016].

[7] N. Gohring. “Amazon’s S3 Down for Several Hours,” Internet: http://www.pcworld.com/article/142549/article.html, Feb 15, 2008 [Jan. 24, 2016].

[8] Wikipedia. “Patrior Act,” Internet: https://en.wikipedia.org/wiki/Patriot_Act. [Feb. 13, 2016].

[9] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica and M. Zaharia. “A View of Cloud Computing.” Communications of the ACM, vol. 53, no. 4, pp. 50-58, April 2010.

[10] S. Gupta, P. Kumar and A. Abraham. “Cloud computing: Trust issues, challenges, and solutions,” in Managing Trust in Cyberspace, CRC press by Taylor and Francis Group, 2013, pp. 13-39.

[11] X. Li and J. Du. “Adaptive and Attribute based Trust model for service-level agreement guarantee in cloud computing.” IET Information Security, vol. 7, no. 1, pp. 39-50, March 2013.

[12] K. M. Khan and Q. Malluhi. “Establishing Trust in Cloud Computing.” IT Professional, vol. 12, no. 5, pp. 20-27, Sept-Oct 2010.

[13] P. I. Bhosle and S. A. Kasurkar. “Trust in Cloud Computing.” International Journal of Advanced Research in Computer Engineering & Technology (IJARCET), vol. 2, no. 4, pp. 1541-1548, April 2013.

[14] A. S. Horvat and R. Agrawal, “Trust in cloud computing,” in SoutheastCon2015, 2015, pp. 1-8.

[15] A. Kanwal, R. Masood, U. E Ghazia, M. A. Shibli and A. G. Abbasi. “Assessment Criteria for Trust Models in Cloud Computing,” in Green Computing and Communications (GreenCom), 2013 IEEE and Internet of Things (iThings/CPSCom), IEEE International Conference on and IEEE Cyber, Physical and Social Computing, 2013, pp. 254-261.

[16] F. Corradini, F. Angelis, F. Ippoliti and F. Marcantoni. “A Survey of Trust Management Models for Cloud Computing,” in 5th International Conference on Cloud Computing and Services Science, 2015, pp. 158-162.

[17] K. Rathi, D. Chhotu and M. S. Kumari. “A Survey on Trust in Cloud Computing.” International Journal of Engineering Technology, Management and Applied Sciences, vol. 3, no. 1, pp. 175-181, Jan. 2015.

[18] K. Rathi and S. Kumari. “Analyzing and Surveying Trust in Cloud Computing Environment.” IOSR Journal of Computer Engineering, vol. 17, no. 3, pp. 66-70, May-Jun 2015.

[19] J. Huang and D. M. Nicol. “Trust mechanisms for cloud computing.” Journal of Cloud Computing: Advances, Systems and Applications, vol. 2, no. 9, pp. 2-14, Dec 2013.

[20] K. Hwang and D. Li. “Trusted Cloud Computing with Secure Resources and Data Coloring.” IEEE Internet Computing, vol. 14, no. 5, pp. 14-22, Sept–Oct. 2010.

[21] S. M. Habib, S. Ries, M. Mühlhäuser and P. Varikkattu. “Towards a trust management system for cloud computing marketplaces: using CAIQ as a trust information source.” in Security and Communication Networks, John Wiley and Sons, Ltd., vol. 7, no. 11, pp. 2185-2200, Nov. 2014.

[22] T. Noor, Q. Sheng, L. Yao, S. Dustdar and A. Ngu. “CloudArmor: Supporting Reputation-based Trust Management for Cloud Services.” IEEE transactions on parallel and distributed systems, vol. 27, no. 2, pp. 367-380, March 2015.

[23] R. Ko, P. Jagadpramana, M. Mowbray, S. Pearson, M. Kirchberg, Q. Liang and B. S. Lee. “TrustCloud: A framework for accountability and trust in cloud computing,” in 2011 IEEE World Congress on Services (SERVICES’11), 2011, pp. 584-588.

[24] M. Ahmed and Y. Xiang. “Trust ticket deployment: a notion of a data owner’s trust in cloud computing,” in 2011IEEE 10th International Conference on Trust, Security and Privacy in Computing and Communications, 2011, pp. 111–117.

[25] K. Hwang, S. Kulkarni and Y. Hu. “Cloud Security with Virtualized Defense and Reputation-based Trust Management,” In Proc. of IEEE 8th Int. Conf. on Dependable, Autonomic and Secure Computing (DASC’09), 2009, pp. 717-722.

[26] I. Brandic, S. Dustdar, T. Anstett, D. Schumm and F. Leymann. “Compliant cloud computing (c3): Architecture and language support for user-driven compliance management in clouds,” in Proceedings of IEEE 3rd International Conference on Cloud Computing (CLOUD’10), 2010, pp. 244-251.

[27] M. Alhamad, T. Dillon and E. Chang. “Sla-based trust model for cloud computing,” in Network-Based Information Systems (NBiS), 2010 13th IEEE International Conference, 2010, pp. 321–324.

[28] P. Pawar, M. Rajarajan, S. Nair, and A. Zisman. “Trust model for optimized cloud services,” in Trust Management VI. Springer, 2012, pp. 97–112.

[29] S. Chakraborty and K. Roy. “An SLA-based framework for estimating trustworthiness of a cloud,” in Trust, Security and Privacy in Computing and Communications (TrustCom), 2012 IEEE 11th International Conference, 2012, pp. 937–942.

[30] D. Marudhadevi, V. N. Dhatchayani, and V. S. Sriram, “A trust evaluation model for cloud computing using service level agreement.” The Computer Journal, bxu129, Nov 2014.

[31] Z. Yang, L. Qiao, C. Liu, C. Yang, and G. Wan. "A Collaborative Trust Model of Firewall-through based on Cloud Computing," in 14th International Conference on Computer Supported Cooperative Work in Design (CSCWD), 2010, pp. 329 - 334.

[32] W. Li, L. Ping and X. Pan. "Use trust management module to achieve effective security mechanisms in cloud environment," in International Conference on Electronics and Information Engineering (ICEIE), 2010, pp. 14-19.

[33] Z. Song, J. Molina, and C. Strong. "Trusted Anonymous Execution: A Model to RaiseTrust in Cloud," in 9th International Conference on Grid and Cooperative Computing (GCC), 2010, pp. 133 - 138.

[34] X. Y. Li, L. T. Zhou, Y. Shi, and Y. Guo. "A trusted computing environment model in cloudarchitecture," in Ninth International Conference on Machine Learning and Cybernetics (ICMLC), 2010, pp. 2843-2848.

[35] I. Foster., Y. Zhao, I. Raicu and S. LU. "Cloud computing and grid computing 360-degree compared," in Proceedings of the Grid Computing Environments Workshop (GCE'08), 2008, pp. 1-10.

[36] Y. Wei and M. B. Blake, "Service-oriented computing and cloud computing: challenges and opportunities." IEEE Internet Computing, vol. 14, no. 6, pp. 72–75, Nov-Dec 2010.

[37] B. Sotomayor, R. S. Montero and I. Foster, "Virtual infrastructure management in private and hybrid clouds." IEEE Internet Computing, vol. 13, no. 5, pp. 14–22, Sept-Oct 2009.

[38] D. Gotfrid. "Self-service, prorated supercomputing fun." Internet: http://open.blogs.nytimes.com/2007/11/01/self-service-prorated-super-computing-fun/, Nov 1, 2007 [Aug. 17 2015].

[39] S. Pearson and Azzedine Benameur. "Privacy, Security and Trust Issues Arising from Cloud Computing," 2nd IEEE International Conference on Cloud Computing Technology and Science, 2007, pp. 693-702.

[40] T. Muller. “Formal Aspects of Security and Trust”, Springer Berlin Heidelberg, 2011, pp. 141-156.

[41] C. P. Pfleeger and S. L. Pfleeger. Security in Computing (4th Edition). Prentice Hall PTR, Upper Saddle River, NJ, USA, 2006, pp. 603-647.

[42] S. Subashini and V. Kavitha. “A survey on security issues in service delivery models of cloud computing.” Journal of Network and Computer Applications, vol. 34, pp. 1-11, Jan 2011.

[43] S. Person. “Taking account of privacy when designing cloud computing services.” Hewlett-Packard Development Company, L.P, pp. 1-10, March 2009.

[44] C. P. Pfleeger and S. Lawrence Pfleeger, Security in Computing (5th Edition). Prentice Hall PTR, Upper Saddle River, NJ, USA, 2015, pp. 432- 496.

[45] O. M. willu Sangupamba, N. Prat and I. Comyn-Wattiau. “Advanced in Conceptual Modeling” in lecture notes in computer science, Atlanta, GA, 2014, vol. 8823, pp. 75-84.

[46] ENISA. “An SME perspective on cloud computing-survey,” Internet: http://www.enisa.europa.eu/, Nov. 20, 2009 [Oct. 12 2015].

[47] C. Torode. “Beware these risks of cloud computing, from no SLAs to vendor lock-in,” Internet: http://searchcio.techtarget.com/news/1363821/Beware-these-risks-of-cloud-computing-from-no-SLAs-to-vendor-lock-in, Aug. 6 2009 [Oct. 12 2015].

[48] Sun Systems. “Building Customer Trust in Cloud Computing with Transparent Security.” Sum Microsystems, Inc. pp. 1-26, Nov 2009.

[49] D. H. Mcknight,N and L. Chervany. “What trust means in e-commerce customer relationships: an interdisciplinary conceptual typology.” International Journal of Electronic Commerce, vol. 6, no. 2, pp. 35–59, Winter 2001/2002.

[50] J. Huang and D. Nicol. “A formal-semantics-based calculus of trust.” IEEE internet computing, vol. 14, no. 5, pp. 38–46, June 2010.

[51] H. Zhu, F. Bao and R. H. Deng, “Computing of trust in distributed networks,” in IACR Cryptology ePrint Archive, 2003.

[52] V. Kumar, B. Chejerla, S. Madria and M. Mohania. “A survey of trust and trust management in cloud computing,” Managing Trust in Cyberspace, CRC press, 2013, pp. 41.

[53] R. Lee and M. Madden. “The State of Music Downloading and File-Sharing Online.” Internet: http://www.pewtrusts.org/en/research-and-analysis/reports/2004/04/25/the-state-of-music-downloading-and-filesharing-online, April 25, 2014 [May 27, 2015].

[54] J. D. Lewis and A. J. Weigert. “Trust as a social reality.” Oxford Journals, Social forces, vol. 63, no. 4, pp. 967–985, Jun 1985.

[55] A. S. Horvat and R. Agrawal. “Trust in cloud computing,” in SoutheastCon2015, 2015, pp. 1-8.

[56] A. Kanwal, R. Masood, U. E Ghazia, M. A. Shibli and A. G. Abbasi. “Assessment Criteria for Trust Models in Cloud Computing,” in IEEE International Conference on Green Computing and Communications and IEEE Internet of Things and IEEE Cyber, Physical and Social Computing, 2013, pp. 254-261.

[57] F. Corradini, F. D. Angelis, F. Ippoliti and F. Marcantoni. “A Survey of Trust Management Models for Cloud Computing,” in 5th International Conference on Cloud Computing and Services Science, 2015, pp. 158-162.

[58] K. Rathi, D. Chhotu and M. S. Kumari, “A Survey on Trust in Cloud Computing.” International Journal of Engineering Technology, Management and Applied Sciences, vol. 3, no. 1, pp. 175-181, Jan 2015.

[59] K. Rathi and S. Kumari. “Analyzing and Surveying Trust in Cloud Computing Environment.” IOSR Journal of Computer Engineering, vol. 17, no. 3, pp. 66-70, Jun 2015.

[60] J. Huang and D. M. Nicol. “Trust mechanisms for cloud computing.” Journal of Cloud Computing: Advances, Systems and Applications, vol. 2, no. 9, pp. 2-14, Dec 2013.

[61] M. Blaze, J. Feigenbaum and J. Lacy. "Decentralized Trust Managament," in IEEE Symposium on Security and Privacy, 1996, pp. 164-173.

[62] S. Ganeriwal, L. K. Balzano and M. B. Srivastava. "Reputation-based Framework for High Integrity Sensor Networks." ACM Transactions on Sensor Networks, vol. 4, no. 15, pp. 1-38, May 2008.

[63] J. Huang and D. M. Nicol. “Trust mechanisms for cloud computing.” Journal of Cloud Computing: Advances, Systems and Applications, vol. 2, no. 9, pp. 2-14, April 2013.

[64] S. X. Wang, L. Zhang, S. Wang and X. Qiu. “A cloud-based trust model for evaluating quality of web services.” Journal of computer science and technology, vol. 25, pp. 1130–1142, Nov 2013.

[65] I. U. Haq, I. Brandic and E. Schikuta. “Sla validation in layered cloud infrastructures” in Economics of Grids, Clouds, Systems and Services, Springer-Verlag, Berlin, Heidelberg, 2010, Vol. 6296, pp. 153–164.

[66] T. Applogic, “3tera’s Cloud Computing SLA goes live” Internet: http://blog.3tera.com/computing/175/, 2009 [Jun. 11 2015].

[67] "Cloud computing use cases white paper," Cloud Computing Use Cases Discussion Group, July 2009.

[68] A. Kanwal, R. Masood, U. E. Ghazia, M. A. Shibli and A. G. Abbasi. "Assessment criteria for trust models in cloud computing," In Green Computing and Communications (GreenCom), 2013 IEEE and Internet of Things (iThings/CPSCom), IEEE International Conference on and IEEE Cyber, Physical and Social Computing, 2013, pp. 254–261.

[69] S. M. Habib, S. Ries and M. Muhlhauser. "Cloud computing landscape and research challenges regarding trust and reputation," Ubiquitous Autonomic and Trusted Computing, Symposia and Workshops, 2010, pp. 410–415.

[70] Cloud Computing Use Cases Discussion Group. "Cloud computing use cases white paper Version 4.0." Cloud Computing Use Cases Discussion Group, 2010.

[71] CSA: Cloud Security Aliance, "Domain 12: Guidance for Identity & Access Management V2.1", https://cloudsecurityalliance.org/guidance/csaguide-dom12-v2.10.pdf, April 2010 [July. 2 2015].

[72] P. Membrey, K. C. C. Chan, N. Canh, Y. Demchenko and C. de Laat. "Trusted Virtual Infrastructure Bootstrapping for On Demand Services, Availability, Reliability and Security (ARES)," 2012 Seventh International Conference, 2012, pp. 350-357.

[73] Y. Demchenko, C. Ngo, C. de Laat, J. Rodriguez, L. M. Contreras, J. A. Garcia-Espin, S. Figuerola, G. Landi and N. Ciulli. "Intercloud Architecture Framework for Heterogeneous Cloud Based Infrastructure Services Provisioning On-Demand, Advanced Information Networking and Applications Workshops (WAINA)," 2013 27th International Conference, 2013, pp. 777-784.

[74] Z. Malik and A. Bouguettaya. “Reputation Bootstrapping for Trust Establishment among Web Services.” IEEE Internet Computing, vol. 13, no. 1, pp. 40-47, Jan-Feb 2009.

[75] A. Josang, R. Ismail and C. Boyd. “A survey of trust and reputation systems for online service provision.” Decision Support Systems, vol. 43, no. 2, pp. 618–644, March 2007.

[76] “SSH from the browser” internet: https://cloud.google.com/compute/docs/ssh-in-browser, [Feb. 2 2016].

[77] Y. Wang and K. Lin. “Reputation-Oriented Trustworthy Computing in E-Commerce Environments.” IEEE Internet Computing, vol. 12, no. 4, pp. 55–59, July-Aug 2008.

[78] R. Buyya, C.S. Yeo and S. Venugopal. “Market-oriented cloud computing: Vision, hype, and reality for delivering it services as computing utilities,” in High Performance Computing and Communications, 2008. HPCC’08. 10th IEEE International Conference, 2008, pp. 5–13.

[79] S. Singh and C. Morley. “Young Australians privacy, security and trust in internet banking,” in Proceedings of the 21st Annual Conference of the Australian Computer-Human interaction Special interest Group, 2009, pp. 121-128.

[80] Y. Wang and J. Vassileva. “A review on trust and reputation for web service selection,” Distributed Computing Systems Workshops, 2007. ICDCSW '07. 27th International Conference, 2007, pp 25.

[81] S. M. Habib, S. Ries and M. Mühlhäuser. “Cloud Computing Landscape and Research Challenges regarding Trust and Reputation,” Ubiquitous Intelligence & Computing and 7th International Conference on Autonomic & Trusted Computing (UIC/ATC), 2010 7th International Conference, 2010, pp. 410-415.

[82] D. Osterwalder. “Trust Through Evaluation and Certification,” Social Science Computer Review - The digital imperative of social sciences in the new millennium, vol. 19, no. 1, pp. 32-46, Mar 2001.

[83] W. Li and L. Ping. “Trust model to enhance security and interoperability of cloud environment,” Cloud Computing: First International Conference, CloudCom 2009, 2009 pp. 69-79.

[84] M. Wanga, G. Wangb, J. Tianb, H. Zhanga and Y. Zhanga. “An Accurate and Multi-faceted Reputation Scheme for Cloud Computing,” The 11th International Conference on Mobile Systems and Pervasive Computing (MobiSPC-2014) Procedia Computer Science, 2014, vol. 34, pp. 466 – 473.

[85] W.n Tan, Y. Sun, L. X. Li, G. Lu, and T. Wang. “A Trust Service-Oriented Scheduling Model for Workflow Applications in Cloud Computing.” IEEE SYSTEMS JOURNAL, vol. 8, no. 3, pp. 686-878, June 2013,

[86] M. K. Muchahari and S. K. Sinha. “A New Trust Management Architecture for Cloud Computing Environment,” in IEEE Cloud Society International Symposium on Cloud and Services Computing, 2012, pp. 136–140.

[87] Z. Raghebi and M. Hashemi. “A New Trust Evaluation Method based on Reliability of Customer Feedback for Cloud Computing,” in Information Security and Cryptology (ISCISC), 2013 10th International ISC Conference, 2013, pp. 1-6.

[88] W. Fan and H. Perros. “A Reliability-based Trust Management Mechanism for Cloud Services,” in 12th IEEE International Conference on Trust, Security and Privacy in Computing and Communications, 2013, pp. 1581-1586.

[89] T. H. Noor, Q. Z. Sheng, L. Yao, Schahram Dustdar and Anne H.H. Ngu. “CloudArmor: Supporting Reputation-based Trust Management for Cloud Services.” IEEE transactions on parallel and distributed systems, pp. 367-380, Mar 2015.

[90] X. Li, H. Ma, W. Yao and X. Gui. “Data-driven and Feedback-Enhanced Trust Computing Pattern for Large-scale Multi-Cloud Collaborative Services.” IEEE Transactions on Services Computing, vol. PP, no. 99, pp. 1, Sept. 2015.

[91] S. Rizvi, J. Ryoo, Y. Liu, D. Zazworsky and A. Cappeta. “A Centralized Trust Model Approach for Cloud Computing,” in IEEE Wireless and Optical Communication Conference (WOCC), 2014, pp. 1-6.

[92] A. Bradai, W. Ben-Ameur and H. Afifi. “Byzantine Resistant Reputation-based Trust Management,” in Collaborative Computing: Networking, Applications and Worksharing (Collaboratecom), 9th International Conference Conference, 2013, pp. 269-278.

[93] N. Agheli, B. Hosseini and A. Shojaee. “A Trust Evaluation Model for Selecting Service Provider in Cloud Environment,” in Computer and Knowledge Engineering (ICCKE), 2014 4th International eConference, 2014, pp. 251-255.

[94] X. Zhang, H. Liu, B. Li, X. Wang, H. Chen and S. Wu. “Application-Oriented Remote Verification Trust Model in Cloud Computing,” in 2nd IEEE International Conference on Cloud Computing Technology and Science, 2010, pp. 405-408.

[95] Y. Liu, Y. Ma, H. Zhang, D. Li and G. Chen. “A Method for Trust Management in Cloud Computing: Data Coloring by Cloud Watermarking.” International Journal of Automation and Computing, vol. 3, no. 8, pp. 280-285, Aug 2011.

[96] D. Kong and Y. Zhai. “Trust Based Recommendation System in Service Oriented Cloud Computing,” in Cloud and Service Computing (CSC), 2012 international conference, 2012, pp. 176-179.

[97] S. Singh and D. Chand. “Trust evaluation in cloud based on friends and third party’s recommendations,” in Engineering and Computational Sciences (RAECS), 2014 Recent Advances, 2014, pp. 1–6.

[98] S. Han, M. Hassan, C. Yoon and E. Huh. “Efficient service recommendation system for cloud computing market,” in 2nd international conference on interaction sciences: information technology, culture and human, 2009, pp. 839-845.

[99] T. H. Noor and Q. Z. Sheng. “Trust as a service: A framework for trust management in cloud environments,” in Web Information System Engineering–WISE 2011, 2011, pp. 314–321.

[100] M. Firdhous, O. Ghazali and S. Hassan. “A trust computing mechanism for cloud computing,” in Kaleidoscope 2011: The Fully Networked Human Innovations for Future Networks and Services (K-2011), proceedings of ITU, 2011, pp. 1–7.

[101] M. Firdhous, O. Ghazali and S. Hassan. “A trust computing mechanism for cloud computing with multilevel thresholding,” in Industrial and Information Systems (ICIIS), 2011 6th IEEE International Conference, 2011, pp. 457–461.

[102] S. Wang, J. Wei, L. Sun, Q. Sun and F. Yan. “Reputation Measurement of Cloud Services based on Unstable Feedback Ratings,” in IEEE International Conference on Parallel and Distributed Systems, 2013, pp. 474- 479.

[103] F. J. Krautheim, D. S. Phatak and A. T. Sherman. “Introducing the Trusted Virtual Environment Module: A New Mechanism for Rooting Trust in Cloud Computing,” in

TRUST'10 Proceedings of the 3rd international conference on Trust and trustworthy computing, 2010, p. 211-227.

[104] L. Guoyuan, W. Danru, B. Yuyu and L. Min. “MTBAC: A Mutual Trust Based Access Control Model in Cloud Computing.” China Communications, vol. 11, no. 4, pp. 154-162, April 2014.

[105] P. D. Manuel, T. Selve, M. Ibrahim and A. Barr. "Trust management system for grid and cloud resources," in First International Conference on Advanced Computing (ICAC 2009), 2009, pp. 176-181.

[106] T. Wang, B. Ye, Y. Li and Y. Yang. “Family Gene Based Cloud Trust Model,” in Educational and Network Technology (ICENT), 2010 International Conference, 2010, pp. 540-544.

[107] Z. Wan, J. Liu and R. H. Deng. “HASBE: A Hierarchical Attribute-Based Solution for Flexible and Scalable Access Control in Cloud Computing”, Information Forensics and Security, IEEE Transactions, vol. 7, no. 2, pp. 743-754, Oct 2011.

[108] Z. Shen, L. Li, F. Yan and X. Wu. "Cloud Computing System Based on Trusted Computing Platform," in International Conference on Intelligent Computation Technology and Automation (ICICTA), 2010, pp. 942 - 945.

[109] Z. Shen and Q. Tong. "The security of cloud computing system enabled by trusted computing technology," in 2nd International Conference on Signal Processing Systems (ICSPS), 2010, pp. 11-15.

[110] X. Y. Li, L. T. Zhou, Y. Shi, and Y. Guo. "A trusted computing environment model in cloud architecture," in Ninth International Conference on Machine Learning and Cybernetics (ICMLC), 2010, pp. 2843-2848.

[111] W. Ma, X. Li, Y. Shi and Y. Guo. “TVMCM: A Trusted VM Clone Model in Cloud Computing,” in Information Science and Service Science and Data Mining (ISSDM), 2012 6th International Conference, 2012, pp. 607-611.

[112] U. A. Kashif, Z. A. Memon, A. R. Balouch and J. A. Chandio. “Distributed Trust Protocol for IaaS Cloud Computing,” in 12th International Bhurban Conference on Applied Sciences & Technology (IBCAST), 2015, pp. 257-279.

[113] W. H. zhang and H. L. sheng. “An improved trusted cloud computing platform model based on DAA and Privacy CA scheme,” in 2010 International Conference on Computer Application and System Modeling (ICCASM 2010), 2010, pp. V13-33-V13-39.

[114] N. Singh, S. Matele and S. Singh. “An Efficient Approach for Security of Cloud Using Watermarking Technique,” in ICCCT '15 Proceedings of the Sixth International Conference on Computer and Communication Technology 2015, 2015, pp. 31-35.

[115] I. Sudha, N. Vijayalashmy and G. Rupavani. “Implementation of HASH Algorithm in Cloud Watermarking Techniques.” IJREAT International Journal of Research in Engineering & Advanced Technology, vol. 2, no. 2, pp. 1-6, May 2014.

[116] K. Hwang and D. Li. “Trusted Cloud Computing with Secure Resources and Data Coloring.” IEEE Internet Computing, vol. 14, no. 5, pp. 14-22, Sept-Oct 2010.

[117] M. Alhamad, T. Dillon and E. Chang. “SLA-Based Trust Model for Cloud Computing,” in Network-Based Information Systems (NBiS), 2010 13th International Conference, 2010, pp. 321-324.

[118] A. M. Hammadi and O. Hussain. “A Framework for SLA Assurance in Cloud Computing,” in 26th International Conference on Advanced Information Networking and Applications Workshops, 2012, pp. 394-398.

[119] S. Chakraborty. “An SLA-based Framework for Estimating Trustworthiness of a Cloud,” in Trust, Security and Privacy in Computing and Communications (TrustCom), 2012 IEEE 11th International Conference, 2012, pp. 937-942.

[120] O. Jules, A. Hafid and M. A. Serhani. “Bayesian Network, and Probabilistic Ontology Driven Trust Model for SLA Management of Cloud Services,” in Third International Conference, ACIIDS, 2014, pp. 77-83.

[121] L. Qi, W. Dou, J, Ni, X. Xia, C. Ma and J. Liu. “A Trust Evaluation Method for Cloud Services with Fluctuant QoS and Flexible SLA,” in 2014 IEEE International Conference on Web Services, 2014, pp. 345-352.

[122] Z. Yang, X. Qin, Y. Yang and T. Yagnik. “A Hybrid Trust Service Architecture for Cloud Computing,” in 2013 International Conference on Computer Sciences and Applications, 2013, pp. 674-680.

[123] S. M. Habib, S. Ries and M. Muhlhauser. “Towards a Trust Management System for Cloud Computing.” Security and Communication Networks, vol. 7, no. 11, pp. 1641–2236, Nov 2014.

[124] A. Kanwal, R. Masood and M. A. Shibli. “Evaluation and Establishment of Trust in Cloud Federation”, in 8th International Conference on Ubiquitous Information Management and Communication, 2014, pp. 12-25.

[125] D. Marudhadevi, V. N. Dhatchayani and V. S. Shankar Sriram. “A Trust Evaluation Model for Cloud Computing Using Service Level Agreement.” Oxford Computer Journal, vol. 58, no. 10, pp. 2225-2232, Nov 2014.

[126] P. S. Pawar, M. Rajarajan, S. K. Nair and A. Zisman. “Trust model for optimized cloud services.” Trust Management VI, vol. 374, pp. 97–112, 2012.

[127] X. Wu. “A Fuzzy Reputation-based Trust Management Scheme for Cloud Computing.” International Journal of Digital Content Technology and its Applications (JDCTA), vol. 6, no. 17, pp. 437-445, Sept 2012.

[128] Z. Yang, L. Qiao, C. Liu, C. Yang and G. Wan. “A Collaborative Trust Model of Firewall-through based on Cloud Computing,” in Computer Supported Cooperative Work in Design (CSCWD), 2010 14th International Conference, 2010, pp. 329-334.

[129] V. Varadharajan and U. Tupakula. “TREASURE: Trust Enhanced Security for Cloud Environments,” in 2012 IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications, 2012, pp. 145-152.

[130] W. Li, L. Ping and X. Pan. "Use trust management module to achieve effective security mechanisms in cloud environment," in International Conference on Electronics and Information Engineering (ICEIE), 2010, pp. 14-19.

[131] W. Li and L. Ping. “Trust Model to Enhance Security and Interoperability of Cloud Environment,” in CloudCom ’09 proceeding of the 1st international conference on cloud computing, Springerm, 2009, pp. 69-79.

[132] X. Li and J. Du. “Adaptive and attribute-based trust model for service level agreement guarantee in cloud computing.” Information Security, IET, vol. 7, no. 1, pp. 39-50, Mar 2013

[133] B. K. Raju and G. Geethakumari. “A Model for Trust Enhancement in Cloud Computing,” in Computer and Communications Technologies (ICCCT), 2014 International Conference, 2014, pp. 1-5.

[134] S. M. Habib, S. Ries and M. Muhlhauser. “Towards a Trust Management System for Cloud Computing,” in 2011 International Joint Conference of IEEE TrustCom-11/IEEE ICESS-11/FCST-11, 2011, pp. 933-939.

[135] Z. Song, J. Molina and C. Strong. "Trusted Anonymous Execution: A Model to RaiseTrust in Cloud," in 9th International Conference on Grid and Cooperative Computing (GCC), Nanjing, China, 2010, 2010, pp. 133 - 138.

[136] H. Sato, A. Kanai and S. Tanimoto. "A Cloud Trust Model in a Security Aware Cloud," in 10th IEEE/IPSJ International Symposium on Applications and the Internet (SAINT), 2010, pp. 121 - 124.

[137] R. K. L. Ko, P. Jagadpramana, M. Mowbray, S. Pearson, M. Kirchberg, Q. Liang and B. S. Lee. “TrustCloud: A Framework for Accountability and Trust in Cloud Computing,” in 2011 IEEE World Congress on Services, 2011, pp. 584-588.

[138] M. Almorsy, J. Grundy and A. S. Ibrahim. “Collaboration-Based Cloud Computing Security Management Framework,” in IEEE 4th International Conference on Cloud Computing, 2011, pp. 364-371.