A Novel Trust Management Framework for Multi-cloud Environments Based on Trust Service Providers

Journal:	*IET Information Security*
Manuscript ID:	Draft
Manuscript Type:	Research Paper
Date Submitted by the Author:	n/a
Complete List of Authors:	Fan, Wenjuan; Hefei University of Technology, School of Management; North Carolina State University, Department of Computer Science Perros, Harry; North Carolina State University, Department of Computer Science
Keyword:	Trust and Reputation Management, Trust Models, System and architecture design

A Novel Trust Management Framework for Multi-cloud Environments Based on Trust Service Providers

Wenjuan Fan[1, 2]
Harry Perros[2], IEEE Fellow
[1]School of Management, Hefei University of Technology, Hefei, Anhui, China
wfan3@ncsu.edu
[2]Computer Science Department, NC State University, Raleigh, NC, USA
hp@ncsu.edu

Abstract: In this paper, we address the problem of trust management in multi-cloud environments using a trust management architecture based on a group of distributed Trust Service Providers (TSPs). These are independent third-party providers, trusted by Cloud Providers (CPs), Cloud Service Providers (CSPs) and Cloud Service Users (CSUs), that provide trust related services to cloud participants. TSPs are distributed over the clouds, and they elicit raw trust evidence from different sources and in different formats, i.e., adherence of a CSP to a Service Level Agreement (SLA) for a cloud-based service and the feedback sent by CSUs. Using this information, they evaluate the objective trust and subjective trust of CSPs respectively. Furthermore, we introduce a trust propagation network among TSPs across different clouds, which is used by a TSP to obtain trust information about a service from other TSPs. The proposed trust management framework for a multi-cloud environment is based on the proposed trust evaluation model and the trust propagation network. Experiments show that our proposed framework is effective in differentiating trustworthy and untrustworthy CSPs in a multi-cloud environment.

Keywords: trust management, trust service provider, multi-cloud, subjective trust, objective trust, trust propagation

1 Introduction

Cloud computing is a popular paradigm for providing software applications, platforms and infrastructure resources (Buyya et al., 2008; Armbrust et al., 2010), and it has given rise to important trust-related problems (Khan and Malluhi, 2010; Monsef and Gidado, 2011; Abbadi and Martin, 2011). In the last few years, research has been extended to multi-cloud infrastructures (Grozev and Buyya, 2012; Ngo et al., 2012), federated cloud computing environments (Buyya, et al., 2010), because of the big benefit in solving large-scale computational and data intensive problems. Trust-related issue in multi-clouds involve more complicated content and new problems (Abwajy, 2009; Bernstein and Vij, 2010; Abwajy, 2011), since cloud services are running on distributed computing resources which are integrated through a federation of the computing clouds. In addition, due to the complexity of the service delivery models of multi-cloud applications, trust management becomes especially important and complicated. For example, a physicist may process scientific data hosted by one institution on a remote application server for data mining run by another, and then store the results on a public cloud data service. A solid trust relationship is required among Cloud Service Users (CSUs), Cloud Service Providers (CSPs), and Cloud Providers (CPs) in such open, dynamic and uncertain environments.

For a successful multi-cloud implementation, trust relationships among participants have to be reliably elicited, aggregated, and propagated. As a result, on one hand, from the perspective of CSUs, they can build confidence in adopting cloud-based services, selecting appropriate and reliable CSPs, and stimulate positive cooperation with trustworthy multi-cloud CSPs; and on the other hand, from the perspective of CSPs, it is important for them to compose services seamlessly and dynamically across organization boundaries so that to construct composed cloud services. That is, a CSP also has to assess the trustworthiness of other CSPs to identify reliable ones. Thus, the trustworthiness of involved entities across different clouds needs to be evaluated, maintained and updated. However, to the best of our knowledge, there is a lack of comprehensive research work on establishing a systematic trust management framework for multi-cloud environments.

Toward a robust and effective trust management for multi-cloud environments, we propose a trust management architecture based on Trust Service Providers (TSPs). These are trust-broker agents, trusted by different CPs, CSPs and CSUs and distributed over the multi-cloud. They are independent third-party providers that provide trust-related services for the cloud participants (both CSUs and CSPs). For example, they can provide services which differentiate malicious CSPs from good ones, select trustworthy CSPs for CSUs/CSPs, and make recommendations to CSUs/CSPs with personalized requirements (these services could act as value-added services). In order to successfully offer trust-related services, the TSPs need to have agreements with the CSPs/CPs, so that to be able to monitor their services and/or have access to the monitors deployed by the CSPs/CPs, in order to observe the actual transaction process and get the real-time trust information. CSPs/CPs, on the other hand, are motivated to cooperate with TSPs, since through them they can build a high reputation and gain a better trust level.

Based on the proposed TSP-based trust management architecture, we use a novel two-layer trust evaluation model, consisting of a subjective trust model and an objective trust model, which measures the trustworthiness of a target CSP from two different perspectives. The subjective trust model is based on feedback information received from the CSUs, and the objective trust model is based on actual observations as to how well a CSP/CP adheres to

the agreed upon SLAs. Both the objective and subjective trust evaluation models address the trustworthiness, untrustworthiness, and uncertainty values of the target entities, which constitute the basic trust evaluation framework. Furthermore, we extend the two-layer trust evaluation model by differentiating between local trust (which is based on the interactions of one particular CSU) and global trust (which is based on all the interactions of all CSUs). More particularly, for local trust, there also subdivides the local subjective and local objective trust values so that different CSUs can derive different subjective/objective trust values for the same target based on the CSUs' subjective feedback or objective SLA information, respectively; and for global trust, there also subdivides the global subjective and global objective trust values, so that the same CSP can have two different overall trust values based on the total CSUs' subjective feedback or objective SLA information, respectively. Providing these different trust values can help CSUs to make a better selection.

In order to utilize effectively the trust information from multiple TSPs hosted in different clouds, we developed a trust propagation model based on the notion of TSPs Path of Trust (TPoT). Trust information from CSPs is propagated through the TPoTs after a TSP has flooded a trust request to all the TSPs.

The major contributions of this paper are as follows:

1) A trust management framework based on TSPs is proposed. To the best of our knowledge, this framework is proposed for the first time for multi-cloud environments.
2) The trust evaluation model consists of an objective and subjective trust evaluation models, based on different trust information sources and trust context, which can better formulate the trust relationship based on different sources and format of trust information.
3) Using the objective and subjective trust models, we extended it to a combination of the local objective trust model (*LOT*), the local subjective trust model (*LST*), the global objective trust model (*GOT*), and the global subjective trust model (*GST*). The *LOT* and *LST* are concentrated on the personalized objective and subjective trust respectively, and *GOT* and *GST* focus on the aggregated overall objective and subjective trust respectively, so that personalized and optimized trust decision can be made for cloud service requesters.
4) A trust network of TSPs for trust sharing is proposed, where TSPs establish trust paths to other TSPs. A trust request from a TSP is propagated to the other TSPs by flooding the message over the trust paths.

The rest of the paper is organized as follows. In Section 2, we give a literature review of related work. In Section 3, we describe the proposed conceptual trust management framework for multi-cloud environment. In Section 4, the trust modeling methodology including the subjective and objective trust evaluation models is formalized, followed by an illustration of the trust propagation model in Section 5. The results of simulation experiments that we carried out are reported in Section 6, and the paper is concluded in Section 7.

2. Literature review

Trust has attracted much attention from different research domains, such as psychology, sociology, economics, philosophy, computer science, and management science, while gaining little consensus. There is no general definition of trust, and the definitions are usually discipline-specific (Ozaa et al., 2006). Broadly speaking, trust

means an act of faith, confidence, and reliance in something that is expected to behave or deliver as promised (Khaled and Qutaibah, 2010).

2.1 Trust broker mechanism

During the last decade, a number of studies on trust broker mechanisms for different computing scenarios were reported. Azzedin and Maheswaran (2004) proposed a network of trust brokers for peer-to peer scenarios for providing peer recommendations. The authors applied the trust brokering system to a resource manager in a public resource grid environment. Lin et al. (2005) introduced a broker framework for web applications, where service brokers manage trust information for their respective users. By delegating trust management to brokers, individual users only need to ask their brokers about the reputation of a service before any transaction with a server. Azzedin and Hsu et al. (2006) proposed a trust broker mechanism where users rely on their broker to provide reputation ratings about service providers, and brokers can leverage their concerned partners to aggregate the reputation of unfamiliar service providers. Varalakshmi et al. (2007) proposed a reputation-based trust management architecture to support Service Provider (SP) selection based on trust values passing through immediate nodes and brokers. Trust values of SPs and consumers are evaluated and updated after the completion of each transaction. A set of trust parameters is considered for trust evaluation. The problem of unreliable ratings was not considered in the above papers. Zhang et al. (2006) proposed a distributed reputation and trust management broker framework called DIRECT for e-commerce environment, where malicious attacks are detected by local and cross-broker check mechanisms. Only one attribute is used, i.e., the transaction size (or value), as a factor that determines the probability with which servers may faithfully deliver requested services (Lee and Lin, 2008).

2.2 Subjective trust and objective trust evaluation

Many trust and reputation models have been proposed to evaluate entities based on historical interaction records, feedback, and other recommendations. These models can be classified into two categories: subjective trust and objective trust. Zhang et al. (2004) gave a conceptual classification of the trust functions. They defined that “if the quality of a service can be objectively measured, then an entity’s trustworthiness for that service is called objective trust”, and an entity’s trustworthiness should be “independent of the source of the trust evaluation”. They also argued that “an entity’s subjective trust may vary greatly when different sources of trust evaluation are considered”. However, the situation may be changed given a different context. In this case, the condition can be relaxed, i.e., the objective trust should be perception-independent but not necessarily source-independent. For example, in this paper, the objective trust of a multi-cloud CSP which is evaluated by different TSPs may also vary. This is because one TSP can only capture the SLA monitoring data of a single cloud, and as a result, different TSPs may have different objective trust measurements to the same CSP.

In open and dynamic systems, the subject trust can be viewed as the “subjective probability with which an entity (trustor, which can be a service user or a provider) believes another entity (trustee) will perform an action that has an influence on the trustor’s goal”. There are many classic subjective trust models which address the trust and reputation evaluation. Beth et al. (1994), proposed a formal method for the evaluation of trustworthiness for the

authorization of sensitive tasks. Jøsang (2001) proposed a model based on subjective logic that deals with uncertainty explicitly. Trust is a relationship between two entities for a specific statement (a, b, u), represented using degrees of belief, disbelief, and uncertainty. Wang and Vassileva (2003) provided a Bayesian network-based trust model, which is a flexible method to present differentiated trust and combine different aspects of trust. He et al. (2004) proposed a cloud theory-based trust model to describe the uncertainty concepts, that is, they regard trust between entities as a cloud. Jameel et al. (2005) proposed a trust model that changes with the context and time, and the trust values are updated according to historical records and other factors. Song and Hwang (2005) proposed a dynamic fuzzy logic trust model based on grid computing, which includes the definition and description of trust, fuzzy reasoning and evaluation of trust relationships, and updating and evolution of trust values.

Objective trust management is very important, because reputation-based trust management suffers from attacks, such as, bad mouthing, on-off, conflicting behavior, and newcomer attacks (Buchegger and Le Boudec, 2004; Sun et al., 2006; Li et al., 2008). Witkowski and Pitt (2000) developed the notion of "objective trust" for software agents, which address the trust of agents or between agents based on actual experience. The trust an agent places on another is dynamically updated in the light of new experience. The authors proposed an objective trust-based agent mechanism which is only a direct interaction-derived reputation scheme. Similar to the model by Witkowski et al., Sabater et al. (2002) proposed a model which not only concerns on the overall performance to the agent's direct perception, but they also evaluate the agent's behavior with other agents in the system. Li et al. (2007) proposed an objective trust management framework for MANETs, by which one node evaluates objectively the trustworthiness of another node based on direct observations and also on second-hand information. The authors extended their work (Li et al., 2008) to an attack-resistant objective trust management framework, where different weights are put on different information related to observations of behaviors according to the time of occurrence and providers. Besides, the trust and confidence value are considered and combined into trustworthiness metric. Other studies on objective trust management are also related to the security, which is referred to the "hard trust".

There is limited literature concerning the combination of objective and subjective trust. Yang et al. (2012) proposed a trust evaluation framework combining objective and subjective means, which calculates a degree of trust based on the combination of certified trust and the user's reputation. Tong et al. (2013) studied the relationship between objective trust and subjective trust and defined some properties of each type of trust. Thus considering the limitation of these papers, in this paper we provide a two-layer trust model which includes the subjective trust and objective trust to evaluate the trustworthiness of a CSP/service.

2.3 Trust network and trust propagation

A trust network is a conceptual network that shows trust relations between entities. It can be depicted by a directed acyclic graph where a vertex indicates an entity and an edge between two entities indicates that there is a trust relationship between the entities. Trust networks have been used in many areas, such as, peer-to-peer (Chen, et al., 2009), mobile ad hoc networks (Cho et al., 2011), wireless sensor networks (Singh, 2012), and social networks (Fogel and Nehmad, 2009).

With regard to trust propagation in trust networks, Guha et al. (2004) addressed the problem of transitivity of distrust, that is, if A distrusts B and B distrusts C, then we cannot say that A trusts C. The authors also evaluated and ranked several methods for propagating trust and distrust in a given web of trust. Jøsang et al. (2006) analyzed topologies of trust propagation and proposed the use of subjective logic for modeling trust relationships. De Cock and Da Silva (2006) modeled a trust network as an intuitive fuzzy relation to address the problem of ignorance and vagueness, and derived trust information through a trusted third party. Hang et al. (2009) investigated the operators for trust propagation in social networks, including concatenation and aggregation. Yuan et al. (2010) verified that a trust network is a small-world network, so that it is possible to build a trust relationship between two randomly selected users of the trust network within a limited number of hops.

2.4 Cloud trust

The problem of trust in cloud computing has not been adequately addressed. Pearson and Benameur (2010) studied security, trust and privacy issues in the context of cloud computing and discussed feasible ways that these issues may be addressed. Fan et al. (2013) introduced a two-stage process to evaluate the trustworthiness of cloud services. Habib et al. (2011) proposed a multi-faceted trust management system architecture for a cloud computing marketplace, which aims to identify trustworthy cloud providers in terms of different attributes. Ko et al. (2011) proposed a framework to address accountability issues in cloud computing. Muchahari and Sinha (2012) introduced a trust management architecture that maintains a registry of cloud providers and their respective trust values, and calculates the CSP's trust based on feedback regarding various SLAs and QoS attributes. Mohammed et al. (2010) proposed an SLA-based trust model to evaluate cloud services in order to help cloud users select the most reliable resources, which combines a conceptual SLA framework for cloud computing with a proposed trust management model. Goyal et al. (2012) proposed a trust model that is suitable for Infrastructure as a Service (IaaS) schemes. Chandrasekar et al. (2012) introduced a trust model which monitors the QoS, based on which the trust is established dynamically by making use of a Markov chain model. Li and Du (2013) proposed an adaptive trust management model, which are rough set and induced ordered weighted averaging operator for evaluating the performance of cloud services based on multiple attributes.

With regard to the studies on the trust of a multi-cloud environment, the following is a formal definition of multi-cloud computing.

A cloud model that, for the purpose of guaranteeing service quality, such as the performance and availability of each service, allows on-demand reassignment of resources and transfer of workload through a [sic] interworking of cloud systems of different CPs based on coordination of each consumers requirements for service quality with each providers SLA and use of standard interfaces (Global Inter-Cloud Technology Forum, 2010.).

Some inter-cloud trust management papers have been published in the open literature. Abawaj (2009) proposed a distributed framework which allows entities to determine the trustworthiness of federated cloud computing entities. Ngo et al. (2012) proposed a dynamic trust establishment approach incorporated into cloud services provisioning lifecycles for the multi-provider in the inter-cloud environment. The approach is attribute-based, and the attribute

semantics can be transformed between domains. It also involves multiple levels of delegation and dynamic trust chain establishment.

There is still lack of comprehensive research work on establishing a systematic trust management framework for the multi-cloud environment, which integrates objective and subjective trust evaluation, trust propagation, and multi-attribute aspects of trust issues.

3. Trust issues analysis and proposed trust management framework

3.1 Trust issues analysis for multi-cloud environment

In a multi-cloud environment, there are a lot of CSPs offering a large variety of services. Consequently, it is desirable that CSUs are able to select the most trustworthy CSPs for a particular service. Therefore, functionalities to manage the flow of the trust information (i.e., risk analysis, monitoring information, attribute date, user feedback, etc) across clouds are required. For this reason, a robust trust management must be put in place for cloud deployment and interaction in an effective and secure way. However, due to gaps in trust mechanisms and protocols over different clouds, there is still a lack of a dynamic federal cloud service trust management framework.

The trustworthiness of cloud services is also related to the QoS, security, privacy protection, and other parameters associated with a service. The trustworthiness and the QoS of services can be seen as the objective trust, and it can be measured under a uniformed framework by using parameters related to the context of a service. Last but not least, at the service interaction layer, trust is also a subjective concept, i.e., a subjective perception related to the entities' preference, requirements, profile, etc. This kind of trust can also be affected by many factors, such as the direct interaction experience, and recommendations from other entities.

Therefore, one of the most important issues in a trust management framework is trust evaluation. The trust level of an entity in a system is quantified as trustworthiness. Cloud services should be evaluated based on fine-grained QoS parameters together with customer's feedback, recommendations, and further specific requirements related to the cloud computing environment. In order to specify the trust factors involved in a cloud computing scenario, a set of such attributes is given in Habib et al., 2010 and in the Cloud Controls Matrix (CCM) by Cloud Security Alliance (CSA) (2011). According to the literature and industry practice, many aspects of attributes need to be considered when deriving the trustworthiness of a cloud-based service, such as, the availability, reliability, response time, security, privacy, transparency, and customer support.

Based on the above analysis, we can infer that the trustworthiness of cloud services depends on two aspects of trust information sources, that is, the system performance records and the trust information feedback from CSUs. Thus, we concentrate on the two categories of trust values, i.e., the objective trust and the subjective trust. In particular, the objective trust evaluation model measures the trustworthiness of multi-cloud services from an objective perspective, and the application runtime performance is a source of intuitive evidence and can further serve as the basis for calculating the objective trust evaluation. On the other hand, the subjective trust evaluation model measures the trustworthiness from the perspective of CSUs' perception, based on past service interactions.

Besides the trust evaluation model, we also take into account the propagation of trust relationships in our proposed trust management framework. Like in a real social scenario, trust relationships in the multi-cloud

environment can also be propagated through some mechanism. Trust relationships, which relate trustors and trustees, exists in the whole computing environment and it can form a trust network. In our proposed framework, there are mainly three kinds of nodes in the trust network: CSUs, CSPs, and TSPs either as trustors or trustees. By connecting the nodes through a trust network, the trust information can be shared across clouds and can lower the computation burden of collecting and aggregating the global data. However, due to the large size of the network, the reliability of the trust propagation path (from the source node to the target node) is very important.

3.2. Conceptual system model

3.2.1 Multi-cloud framework

Based on the above trust issues analysis for multi-cloud services, we propose a TSP-based federal trust management framework which can address the challenges in managing trust in multi-cloud services. This is shown in Figure 1.

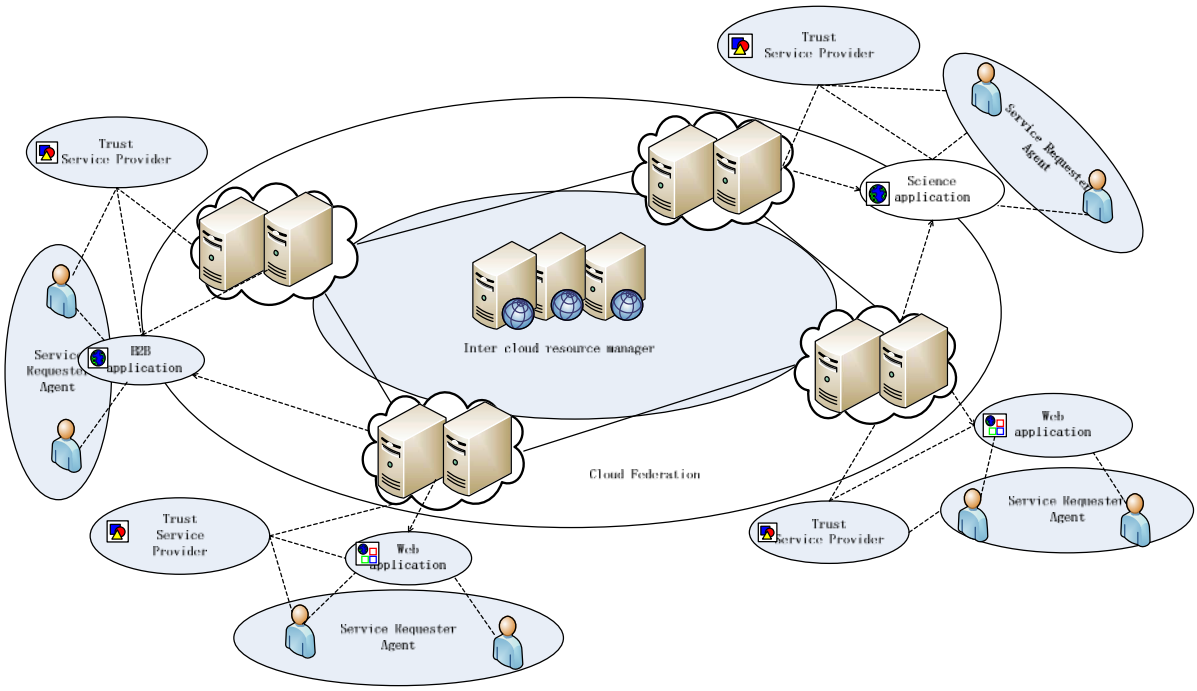


Figure 1: An overall framework for multi-cloud service environment

As shown in Figure 1, a federation of trust management maintained by the multiple CSPs is crucial to allow flexible cloud-based service composition and service integration. In order to achieve scalability, trust relationships between the involved actors should be created on-demand, instead of being statically defined prior to service interaction. However, there is a high uncertainty component when deciding whether or not to cooperate with unknown parties. Thus, every actor that participates in the multi-cloud environment has to make decisions with

some form of risk. A cloud service requester may assess that if it is secure to collaborate with a particular unknown CSP. Similarly, a CSP will have to decide if it is secure to authorize the access from a specific service requester.

3.2 Main actors and their activities

The main actors (entities) in a multi-cloud service federal trust management scenario are: (1) a CSP, which provides services to customers or end users for profit. In the particular context of cloud computing, the CSPs provide a wide range of services in different service delivery models, i.e., XaaS; (2) a CSU, which uses a service offered by a CSP, and can also requests a TSP for the trust value of CSPs so that to select the most trustworthy CSP; and (3) a TSP, which vouches for the trustworthiness of the CSPs that it has agreements with and publishes/updates/shares their global/local trust values. In order to enable trust management in a multi-cloud computing environment, those main actors need implement several types of agents (modules) for trust establishing and evaluation processes. The trust management scheme for multi-cloud service is illustrated in Figure 2.

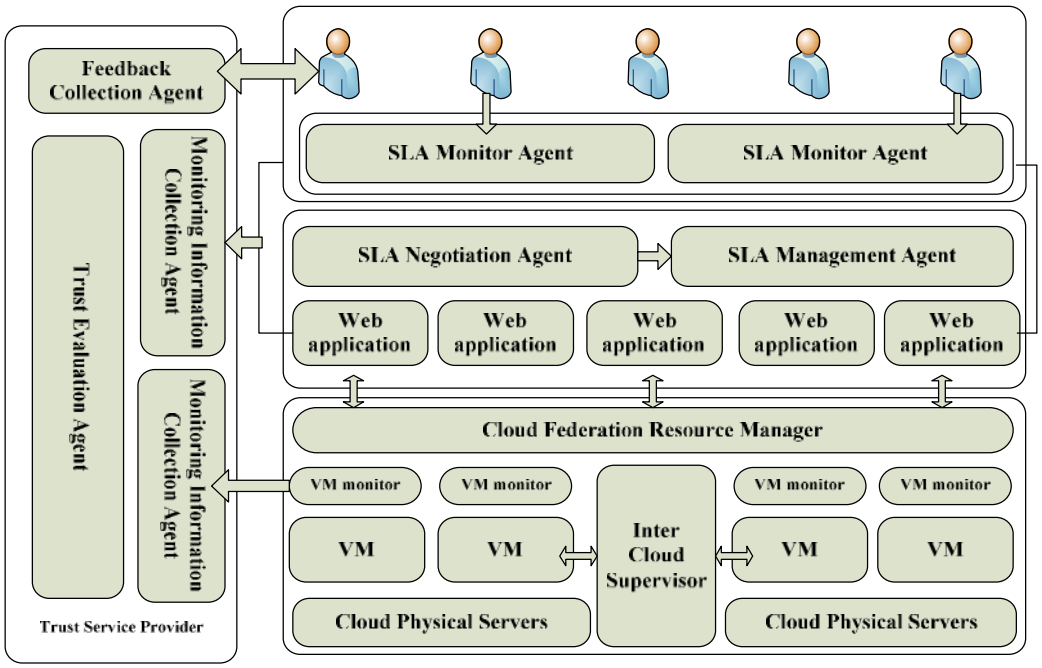


Figure 2: Trust management mechanism for multi-cloud service environment

Below we describe the functions and activities of the main actors.

3.2.1 CSPs

SLA negotiation agents

An SLA negotiation agent is capable of negotiating an SLA between a CSP and a CSU. A CSP has to register its services with the trust management system through an SLA negotiation agent. A CSU can obtain SLA details about a CSP through the CSP's SLA negotiation agent. After a successful negotiation between a CSP and a CSU, the

contracted SLA and the process of negotiation are recorded by the SLA negotiation agent for future validation and auditing. If the negotiation fails, then the SLA negotiation agent only keeps the process of negotiation in record for a certain period of time. An SLA negotiate agent denoted by SNA_i maintains the following information:

- A services directory of CSP_i; (typically, CSP_i may deliver multiple services across different clouds, but in order to simplify the problem, we assume that one CSP offers one multi-cloud service in the system model).
- The service requester catalog of CSP_i;
- The service negotiation record for each CSU_j, including the negotiation requester, negotiation start time, negotiation clauses, negotiation result, and negotiation end time.

SLA management agents

An SLA management agent mainly performs the function of making sure that SLAs are satisfied. For example, if it happens that the SLA of a service application currently being executed is not satisfied, then the SLA management agent adjusts the resource allocation, reschedules the task execution, or migrates the application to another VM or even another cloud, etc, so as to acquire maximum profit and resource utilization. Thus, an SLA management agent may end up violating some part of the SLA of a particular CSU. Therefore, it need to keep a record all the changes in the service deployment in the runtime process, for the purpose of trust management and also for future validation and auditing. An SLA management agent denoted by SMMA for a CSP maintains the following information:

- The current directory of contracted SLAs for all the CSUs that a CSP is interacting with;
- The original task allocation for each CSU. That is, the mapping from the CSU's task to the cloud resource, including the configuration of VM, network, and other important system parameters;
- The reallocation records if there are any changes to the original task allocation, labeled by the time and CSU.

3.2.2 CSUs

SLA monitor agents

On the side of CSUs, monitoring the behavior and performance of services to verify whether they are in compliance with SLAs is an essential issue. This is done using SLA monitor agents. First of all, an agent should not be biased towards a CSP or a CSU. An SLA monitor agent captures data regarding the interaction process between a CSU and a CSP and it also responds to requests for monitoring data from TSPs. The agent collects monitoring information from the server side continuously, which involves all the performance parameters included in an SLA. A single SLA monitor agent can monitor the runtime of all applications with which a specific CSU is in the process of interaction, and a single application can also be monitored by all the SLA monitor agents associated with the CSUs currently interacting with the service. An SLA monitor agent denoted by SMA of a CSU maintains the following:

- The applications that SMA is currently monitoring;
- The set of TSPs with which the applications that SMA is monitoring have cooperation agreements;
- The performance attributes/factors in an SLA that SMA is responsible to monitor;
- The SLA accomplishment reports on the set of the performance attributes/factors of an SLA. This information is collected for each application within a fixed window determined by SMA.

3.2.3 TSPs

Generally speaking, a TSP is a mediation agent between CSPs and CSUs, and we can also call it a trust broker. A set of TSPs are distributed over the Internet and delegated by different CSPs in different clouds to provide trust-related information services. TSPs are invoked when CSUs request cloud services with trust requirements. One TSP can represent multiple CSPs, and one CSP can also delegate multiple TSPs. Like search engine sites and portals, TSPs are independently maintained and operated. CSUs are free to choose among many TSPs available either free or through a paid membership (Lin et al., 2005). A TSP derives the objective trust of CSPs based on trust information sent by monitoring information process agents. It also collects the trust feedback ratings sent by CSUs on services they used, in order to build up the subjective trust about each service. Furthermore, TSPs can also interact with each other in order to exchange and propagate trust information. A TSP_i maintains the following information:

- The entrusted/ delegation relationship with other entities in the cloud:
  a. The set of N_i CSPs that TSP_i represents;
  b. The set of trusted neighbor TSPs of TSP_i;
  c. The set of SLA monitor agents of the CSUs that TSP_i can get access. The SLA monitoring information is not open to all the TSPs, and only those TSPs that have an agreement with a CSU can request it.
- The trust management mechanism applied by the TSP:
  a. The trust inference model/algorithm for TSP_i used to evaluate the trustworthiness of CSPs and the other TSPs;
  b. The trust policy set that TSP_i applies to different contexts of cloud services due to the different service delivery models and service deployment models;
  c. Trust propagation model to select trusted neighbor TSPs and share trust information.

The following agents are implemented in a TSP:

Monitoring information collection agent (*MCA*)

The MCS of a TSP collects information from the SLA monitor agents with which the TSP has agreements. This information is used in the evaluation of the objective trust of a specific service. The monitoring information collection agent of TSP_i, denoted by MCA_i, maintains the following information:

- The list of CSPs that TSP_i has the authority to monitor;
- The SLA monitoring information from the SLA monitor agents with which TSP_i has an agreement.

Feedback collection agent (*FCA*)

The FCA of a TSP is responsible for collecting the subjective feedback from the CSUs who have interacted with the concerned CSPs. A feedback information collection agent denoted by FCA_i for TSP_i maintains the following:

- A list of CSUs whose feedback TSP_i is collecting;
- The actual feedback data from the CSUs in the list.

Trust evaluation agent (*TEA*)

The trust evaluation agent is responsible for the calculation of the subjective and objective trust values of CSPs, based on the information collected from the MCA and FCA. The trust evaluation agent TEA_i for TSP_i maintains the following:

- The trust inference algorithms that TSP_i uses to evaluate the trust of CSPs and the other TSPs;
- The trust policies that TSP_i applies to different contexts of cloud services;
- The data processing and trust calculation module.

The trust value database (*TVD*)

The TVD contains the past calculated trust values of CSPs. These values can further serve as trust evidence for future trust-related decision.

The next Figure 3 illustrates the trust management mechanism of a TSP.

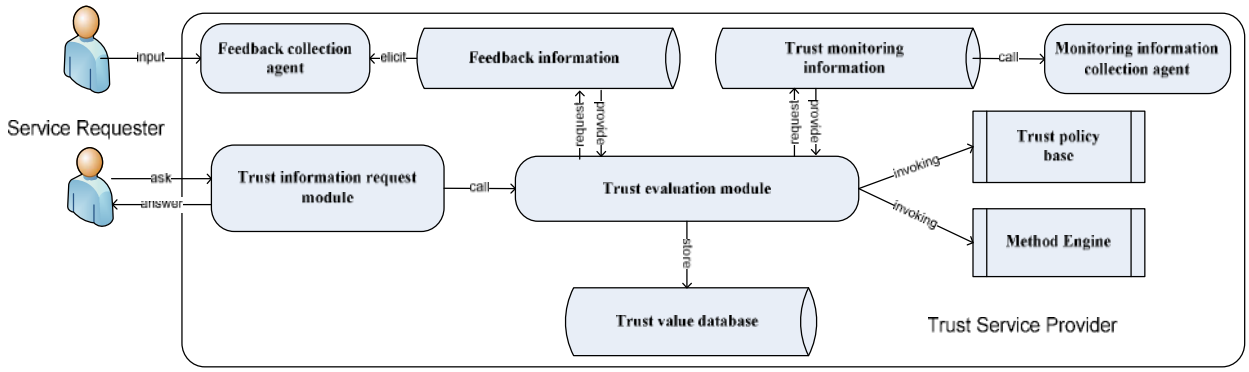


Figure 3: Trust management mechanism scheme of TSP

4. Trust modeling methodology

Following Jøsang's theory, in our modeling methodology, we use a triple consisting of three scalars including belief (i.e., positive trust or belief about trustworthy), disbelief (i.e., distrust or negative trust or belief of untrustworthy), and uncertainty, to model one entity's trust in another entity.

4.1 An SLA-based objective trust evaluation model

We consider the objective trust that is related to multi-attribute trust factors. This type of trust is independent from a CSU's preference. In order to derive the objective trust, TSPs need to collect the execution data and system log records, which are released by CPs and/or CSPs. In view of this, transparency of cloud services is important in the process of objective trust evaluation. In order to process the objective trust evaluation, TSPs need to come to an agreement that authorizes them to monitor the service parameters that are specified in the SLA contracted between CSUs and CSPs. The CSPs usually provide SLAs to CSUs that guarantee a certain level of functional performance of the services, such as, response time, percentage of availability, reliability, security, consistency, flexibility, etc. Given a service, the trust monitors can decide whether it has satisfied the SLA requirements after each transaction and use this information to establishing trust for CSPs.

4.1.1 Local objective trust evaluation (LOT)

In this paper, the objective and subjective trust evaluation is carried out within fixed-sized time windows. We assume a total of *W* windows. Let $P_{i,j} = \{P_{i,j}(k), k = 1, \cdots M\}$ be the set of parameters in an SLA between CSU_i and CSP_j , where M is the total number of parameters, and let $h_{ij}(t)$ be the total number of interactions in window *t*, *t*≤W. For each interaction, a TSP can get the performance records regarding all the parameters in an SLA, and then make a decision as to whether the requirements for these parameters have been a) satisfied (*T*), b) not satisfied (-*T*), or c) uncertain (*U*). Let Θ be the set of possible responses, i.e., $\Theta = \{T, -T, U\}$. A violation of an SLA (i.e., a -*T* in the record) means a negative evidence, a satisfaction of an SLA (i.e., a *T* in the record) means a positive evidence of the function trust on the service application, and an uncertain interaction regarding the SLA (i.e., a *U* in the record) refers to an uncertain evidence. Obviously, the objective trust will increase with the addition of positive evidence and decrease with the addition of negative evidence. The uncertainty of the objective trust will also increase with the addition of uncertain evidence. Let $n_{i,j,k}^{suc}(t)$, $n_{i,j,k}^{failed}(t)$, and $n_{i,j,k}^{un}(t)$ be the total number of interactions during window *t* between CSU_i and CSP_j which classified as satisfied, not satisfied, and uncertain respectively, for each parameter *k* of the SLA between CSU_i and CSP_j For *t*=0, there is no interaction between CSU_i and CSP_j, so $n_{i,j,k}^{suc}(0)=n_{i,j,k}^{failed}(0)=n_{i,j,k}^{un}(0)=0$. For window *t* and for each parameter *k* of the SLA, the local objective trust value is expressed as $LOT_{i,j,k}^{t} = \left(lom_{i,j,k}^{t}\{T\}, lom_{i,j,k}^{t}\{-T\}, lom_{i,j,k}^{t}\{U\}\right)$, where the basic probabilities assignment (BPA) of $lom_{i,j,k}^{t}\{.\}$ are calculated as follows:

$$LOT_{i,j,k}^{t} = \begin{cases} lom_{i,j,k}^{t}\{T\} = \mu \times lom_{i,j,k}^{t-1}(T) + (1-\mu) \times \dfrac{n_{i,j,k}^{suc}(t)}{n_{i,j,k}^{suc}(t) + n_{i,j,k}^{failed}(t) + n_{i,j,k}^{un}(t)} \\ lom_{i,j,k}^{t}\{-T\} = \mu \times lom_{i,j,k}^{t-1}(-T) + (1-\mu) \times \dfrac{n_{i,j,k}^{falied}(t)}{n_{i,j,k}^{suc}(t) + n_{i,j,k}^{failed}(t) + n_{i,j,k}^{un}(t)} \\ lom_{i,j,k}^{t}\{U\} = 1 - lom_{i,j,k}^{t}(T) - lom_{i,j,k}^{t}(-T) \end{cases}$$

where, $0 \leq \mu \leq 1$ is a weight factor. For t=0, we set $LOT_{i,j,k}^{0} = (0.5, 0.5, 1)$. Summing up the probabilities over all parameters of the SLA gives:

$$LOT_{i,j}^{t} = \begin{cases} lom_{i,j}^{t}\{T\} = \sum_{k=1}^{M} \omega_k lom_{i,j,k}^{t}(T) \\ lom_{i,j}^{t}\{-T\} = \sum_{k=1}^{M} \omega_k lom_{i,j,k}^{t}(-T) \\ lom_{i,j}^{t}\{U\} = 1 - lom_{i,j}^{t}(T) - lom_{i,j}^{t}(-T) \end{cases}$$

Where ω_k is a weight vector used by CSU_i to weight the *M* parameters, with $\sum_{k=1}^{M} \omega_k = 1$. $lom_{i,j}^{t}\{T\}$, $lom_{i,j}^{t}\{-T\}$, and $lom_{i,j}^{t}\{U\}$ gives the probability that the objective trust evidence of CSU_i for CSP_j in time window *t* is trustworthy, untrustworthy, and uncertain, respectively.

If there is no enough interaction between CSU_i and CSP_j, i.e., the number of historical interactions is smaller than a minimum number ζ, or the service interactions happened before the time window concerned, then other users' interaction with CSP_j will be taken into consideration.

4.1.2 Global objective trust evaluation

The combination of all the local objective trusts for CSP_j forms its global objective trust. The global objective trust of CSP_j at time t is given by $GOT_j^t = \left(gom_j^t\{T\}, gom_j^t\{-T\}, gom_j^t\{U\}\right)$, where $gom_j^t\{T\}$ is the global objective trustworthiness value, $gom_j^t\{-T\}$ is the global objective untrustworthiness value, and $gom_j^t\{U\}$ global objective uncertainty value. They are calculated as follows:

$$GOT_j^t = \begin{cases} gom_j^t\{T\} = lom_{1,j}^t(T) \oplus lom_{2,j}^t(T) \oplus \cdots lom_{n_j(t),j}^t(T) \\ gom_j^t\{-T\} = lom_{1,j}^t(-T) \oplus lom_{2,j}^t(-T) \oplus \cdots lom_{n_j(t),j}^t(-T) \\ gom_j^t\{U\} = 1 - gom_{1,j}^t(T) - gom_{1,j}^t(-T) \end{cases}$$

Where $n_j(t)$ is the number of CSUs who have interacted with CSP_j in time t. The operator $\oplus$ means (weighted) average operation to all the local objective trust values from all the CSUs.

4.2 Feedback based subjective trust evaluation model

A CSU's trust feedback on a service is a subjective evaluation of the perceived trustworthiness. A CSU tends to trust a service because of a good interaction experience, otherwise it tends to distrust it. Below we describe the subjective trust evaluation model.

4.2.1 Local subjective trust evaluation (LST)

Let $\varphi_{i,j}(t)$ be the total number of historical trust feedbacks of CSU_i for CSP_j during window t, t = 1,2,...,W, where $\sum_{t=1}^{W} \varphi_{i,j}(t) = \varphi_{i,j}$. Let a trust feedback rating be denoted by $f_{i,j}$, $0 \le f_{i,j} \le 1$, where 0 means untrustworthy and 1 means trustworthy, and let $\bar{f}_{i,j}^t$ denote the average trust feedback ratings of CSU_i for CSP_j during window t.

Here we have the following assumptions: If $\bar{f}_{i,j}^t$=0.5, then the evaluation has the highest uncertainty which is equal to 1; if $\bar{f}_{i,j}^t$<0.5 or $\bar{f}_{i,j}^t$>0.5, then the evaluation only has trust and uncertainty. We transform the $\bar{f}_{i,j}^t$ values to the probabilities that the subjective trust evidence of CSU_i for CSP_j in time window t is trustworthy $m_{i,j}^t\{T\}$, untrustworthy $m_{i,j}^t\{\text{-}T\}$, and uncertain $m_{i,j}^t\{U\}$, as follows:

$$\begin{cases} sm_{i,j}^t\{T\} = \begin{cases} \dfrac{\bar{f}_{i,j}^t - 0.5}{0.5} & if\ \bar{f}_{i,j}^t \in [0.5,1] \\ 0 & if\ \bar{f}_{i,j}^t \in [0,0.5] \end{cases} \\ sm_{i,j}^t\{-T\} = \begin{cases} \dfrac{0.5 - \bar{f}_{i,j}^t}{0.5} & if\ \bar{f}_{i,j}^t \in [0,0.5] \\ 0 & if\ \bar{f}_{i,j}^t \in [0.5,1] \end{cases} \\ sm_{i,j}^t\{U\} = 1 - sm_{i,j}^t\{T\} - sm_{i,j}^t\{-T\} \end{cases}$$

The local subjective trust value of CSU_i for CSP_j in time window t is expressed as follows:, $LST_{i,j}^t = (lsm_{i,j}^t\{T\}, lsm_{i,j}^t\{-T\}, lsm_{i,j}^t\{U\})$, where $lsm_{i,j}^t\{T\}$, $lsm_{i,j}^t\{-T\}$, and $lsm_{i,j}^t\{U\}$ denote the probability of the local subjective trustworthiness, local subjective untrustworthiness, and local subjective uncertainty, respectively, and they are calculated as follows:

$$LST_{i,j}^{t} = \begin{cases} lsm_{i,j}^{t}\{T\} = \mu \times sm_{i,j}^{t-1}(T) + (1-\mu) \times \dfrac{\bar{f}_{i,j}^{t} - 0.5}{0.5} \\ lsm_{i,j}^{t}\{-T\} = \mu \times sm_{i,j}^{t-1}(-T) + (1-\mu) \times \dfrac{0.5 - \bar{f}_{i,j}^{t}}{0.5} \\ lsm_{i,j}^{t}\{U\} = 1 - sm_{i,j}^{t-1}(T) - sm_{i,j}^{t-1}(-T) \end{cases}$$

where, $0 \leq \mu \leq 1$ is a weight factor. For t=0, we set $LST_{i,j}^{0} = (0.5,0,1)$.

4.2.2 Global subjective trust reference (GST)

The global subjective trust is an aggregate of the trust that all the CSUs for a specific CSP$_j$. For time window t, it is expressed as $GST_j^t = (gsm_j^t\{T\}, gsm_j^t\{-T\}, gsm_j^t\{U\})$, where $gsm_j^t\{T\}$, $gsm_j^t\{-T\}$, and $gsm_j^t\{U\}$ are the mean global subjective probabilities for trustworthiness, untrustworthiness, and uncertainty respectively, calculated as follows:

$$GST_j^t = \begin{cases} gsm_j^t\{T\} = \dfrac{\sum_{i=1}^{N} c_{i,j} \cdot lsm_{i,j}^t\{T\}}{\sum_{i=1}^{N} c_i} \\ gsm_j^t\{-T\} = \dfrac{\sum_{i=1}^{N} c_{i,j} \cdot lsm_{i,j}^t\{-T\}}{\sum_{i=1}^{N} c_i} \\ gsm_j^t\{TU\} = 1 - gsm_j^t\{T\} - gsm_j^t\{-T\} \end{cases}$$

where N is the number of CSUs who have returned their feedback to TSP in the t time windows. The weight $c_{i,j}$, i=1,2,…,N, is the confidence value regarding CSU$_i$'s trust valuation to CSP$_j$, and it is calculated using the following factors:

- *The frequency of transactions between* CSU$_i$ and CSP$_j$, $\gamma_{i,j}(t)$, in the last t time window. More feedback from CSU$_i$ indicates more confidence on the personal trust value. Therefore, $\gamma_{i,j}(t)$ is expressed as: $\gamma_{i,j}(t) = 1 - e^{-h_{i,j}(t)}$.
- *The mean deviation* $v_{i,j}^t$ *of the feedback*. The larger the mean deviation of all the feedback the less is the confidence on personal trust value. We have: $v_{i,j}^t = \frac{\sum_{w=1}^{\varphi_{i,j}(t)} |f_w - \bar{f}_{i,j}|}{\varphi_{i,j}(t)}$
- *The certainty for derived trust value* $LST_{i,j}^t$

  In general, the uncertainty of a CSU's trust evaluation increases as the trust rating approaches 0.5 and decreases as the trust rating approaches 0 or 1. We assume the function shown in Figure 4 to express this behavior.

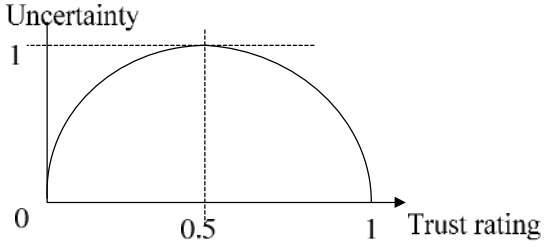


Figure 4: Uncertainty regarding trust ratings

The closer $LST_{i,j}^t$ is to 0 or 1, the more certain it is. Otherwise, the trust value has more uncertainty, with the highest uncertainty achieved when $LST_{i,j}^t$=0.5. Let the certainty of a derived trust value be denoted by $\pi_{i,j}(t)$. Then, we have: $\pi_{i,j}(t)$=1 if $LST_{i,j}^t$=0 or 1, and $\pi_{i,j}(t)$=0 if $LST_{i,j}^t$=0.5. The following certainty function captures these properties: $\pi_{i,j}^t = 1 - \left(-4\left(LST_{i,j}^t\right)^2 + 4LST_{i,j}^t\right)$

Based on the above three factors, we have $c_{i,j}(t) = \omega_1\gamma_{i,j}(t) + \omega_2\pi_{i,j}(t) + \omega_3 e^{-v_{i,j}(t)}$, where $\omega_1 + \omega_2 + \omega_3 = 1$, and $\omega_1, \omega_2, \omega_3 \in (0,1)$.

5 A trust propagation network of TSPs

5.1 TPoT establishment and trust propagation

In this section, we present a trust propagation network of TSPs that can be used by a TSP to get trust values about a CSP from other TSPs. An example of a trust propagation network is shown in Figure 5. Nodes A, B, C, D, and E are TSPs, and a solid line between two nodes indicates that the nodes trust each other. The trust propagation network is formed by each node establishing a trust relation with its neighbors. This relation is binary, i.e., "trust" or "do not trust", and symmetric. In the example in Figure 5, A has established a trust relation with its neighbors B, D, and E. Likewise, node D has established a relationship with neighbors A, E and C, node E with neighbors D, A, and C, node B with neighbors A and C, and node C with neighbors B and E. Now let us assume that TSP A receives a trust request for an unknown CSP. TSP A floods this request to its neighbors B, D, and E. If node B has trust information about the CSP in question, then it will respond back to A, otherwise it will flood the request to its neighbors. Flooding continues and it is possible that it covers all the TSPs in the network. Let us assume that only C has the trust information for the CSP in question, then C will receive the request message from all its neighbors through the paths: ABC, AEC, ADC, and ADEC. Each of these paths is referred to as a TSP Path of Trust (TPoT). For each path, C responds with the trust information that travels in the opposite direction of the path.

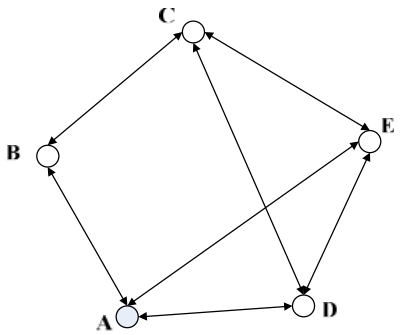


Figure 5: An Example of a trust propagation network of TSPs

We will make the following assumptions regarding TPoTs:

Assumption 1: The reliability of a received recommendation decreases as the length of a TPoT increases. For instance, if TSPs B and C have information about the CSP in question, then they will both respond to A. In this case,

the response from B is considered more reliable than that from C since it is a one-hop TPoT as opposed to at least a two-hop TPoT from C.

Assumption 2: If there is more than one TPoT from an originating TSP to a TSP that has the information, then the shortest path TPoT is used. For instance, let us assume that only TSP E has the information requested by A. As can be seen there are two paths between A and E, i.e., AE and ADE, and in this case the shortest path AE, will be selected.

Assumption 3: If there are multiple TPoTs from the originating TSP to the target TSP with minimum number of intermediary nodes, then one of them is randomly selected. For instance, if we assume that only C has the information, then we have the following four paths: ABC, ADC, AEC, and ADEC. In this case, one of the three paths ABC, ADC, AEC will be randomly selected.

A service may run on several clouds, and in this case it is monitored by different TSPs. There may be more than one TSP associated with a given cloud that monitors the same service. When a TSP floods a request for trust information of a service, more than one TSP associated with the same cloud may respond, of which we chose the one with the shortest TPoT. The answers obtained from the selected TSP from each cloud are aggregated into a single overall result as discussed in the following section. Finally, we note that the establishment of a trust relation between two TSPs can be done using different methods, not considered in this paper.

5.2 Problem formulation

Based on the established TPoT and the basic trust modeling methodology, which are respectively trust metrics, i.e., the local objective trust model (LOT), local subjective trust model (LST), global subjective trust model (GST), and global objective trust model (GOT), each TSP can get the GoT and GST trust values on specific CSPs with respect to CSUs' requests. In our proposed structure of the trust management framework, all these trust values of a specific CSP are stored in the trust value database (TVD) of corresponding TSPs. All TSPs are distributed across different clouds to collect the trust monitoring and feedback information. Each TSP collects these trust information from the certain CSPs they are connected with (here we assume that each TSP only exists in a single cloud, but one cloud can have multiple TSPs).

In Figure 6, the trust propagation structure of the distributed TSPs in a multi-cloud service environment is illustrated. We assume that a service S is deployed in three different clouds, i.e., Cloud 1, Cloud 2, and Cloud 3. There are three TSPs that can collect trust monitoring information and trust feedback from these three clouds, i.e., TSP_1, TSP_2, and TSP_3, which are deployed in Cloud 1, Cloud 2, and Cloud 3 respectively. Let us assume now that a CSU served by another TSP_u requests the trust value of S. TSP_u has no information of this service S, and in this case it will flood the trust request to its neighbors. Through the flooding process, TSP_u can get trust information about S from TSP_1, TSP_2, and TSP_3. These three TSPs may have derived different trust values regarding the service S due to the discrepancy of trustworthiness in the different clouds and the perceptive trust of different CSUs.

Now let us assume that TSP_u, has trust information about this service, but the trust information is only for the cloud TSP_u, is monitoring. Then, in this case it will also flood a trust query to its neighbors TSPs, so that to get trust information about the service running in other clouds.

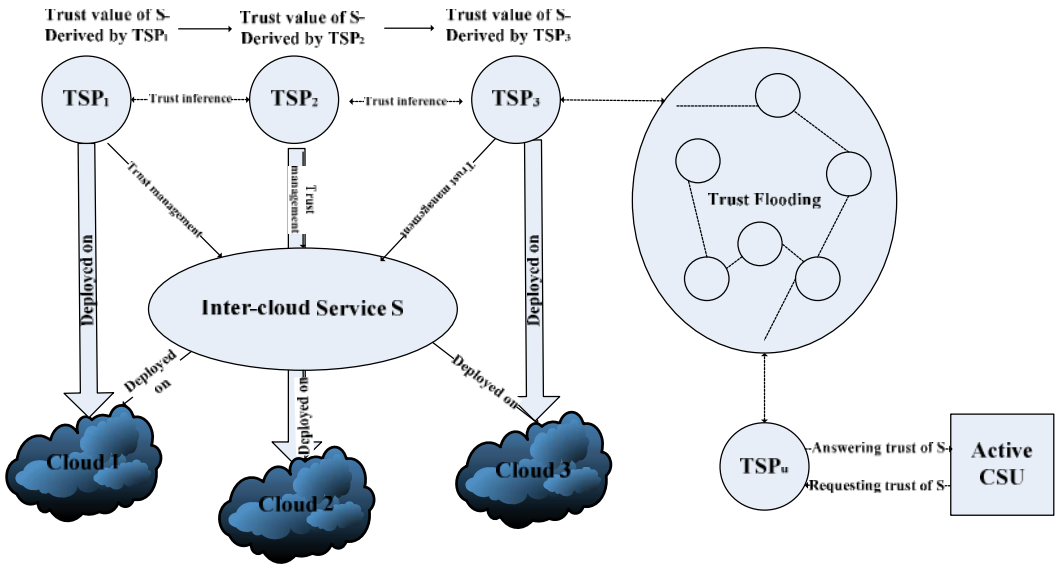


Figure 6: Diagram of trust propagation structure for a multi-cloud service

We assume that a service S is deployed in N clouds. Therefore, at least N TSPs will return the trust value of S from N clouds respectively. The information received from each of these TSPs on the shortest TPoT is aggregated to a final trust value of S. The basic probability assignment of global objective and subjective trust values $\mathrm{GOT}_{i\to S}^{t}$ and $\mathrm{GST}_{i\to S}^{t}$, of S which are derived and returned by TSP_i, i=1,…,N, for time t are as follows:

$$\mathrm{GOT}_{i\to S}^{t} = \left(gom_{i\to S}^{t}(T), gom_{i\to S}^{t}(-T), gom_{i\to S}^{t}(U)\right)$$

$$\mathrm{GST}_{i\to S}^{t} = \left(gsm_{i\to S}^{t}(T), gsm_{i\to S}^{t}(-T), gsm_{i\to S}^{t}(U)\right)$$

where $gom_{i\to S}^{t}(.)$ and $gsm_{i\to S}^{t}(.)$ are the global objective and subjective trust values (trustworthiness, untrustworthiness, or uncertainty) evaluated by TSP_i for service S in time window t.

According to Assumption 1, a TPoT with n intermediate nodes (TSPs) is more valid than a TPoT with $n' \geq n$. We model this using a long-path-attenuation factor $\theta(n) = \theta^{n-1}, \theta \in (0,1)$ to reflect the decay impact of a long path on the reliability of the propagated trust information.

The GST values sent from the different TSPs are aggregated to derive an overall subjective trust value of the multi-cloud CSP, as follows:

$$\mathrm{GST}_{S}^{t} = \begin{cases} gsm_{S}^{t}(T) = \dfrac{\sum_{i=1}^{N} \theta(R_i) \cdot gsm_{i\to S}^{t}(T)}{\sum_{i=1}^{N} \theta(R_i)} \\ gsm_{S}^{t}(-T) = \dfrac{\sum_{i=1}^{N} \theta(R_i) \cdot gsm_{i\to S}^{t}(-T)}{\sum_{i=1}^{N} \theta(R_i)} \\ gsm_{S}^{t}(U) = 1 - gsm_{S}^{t}(T) - gsm_{S}^{t}(-T) \end{cases}$$

where R_i is the number of intermediate TSPs between TSP_u and TSP_i, i=1,…,N. The GOT values provided by the different TSPs are not aggregated, because these objective trust values reflect the trustworthiness of the cloud that the service is deployed on in addition to the trustworthiness of the CSP.

6 Simulation experiments

In order to evaluate our proposed trust model, we simulate a multi-cloud environment with multiple CSPs and CSUs. The trustworthiness of a CSP is specified in advance as trustworthy, or untrustworthy, or it may randomly fluctuate between being trustworthy and untrustworthy during a simulation experiment. CSUs give their feedback to these CSPs, and the SLA monitoring data is randomly generated from a certain range of values that depend on the pre-specified trust level of the CSPs. We first evaluate our model assuming a single cloud with a single TSP, and then we extend our evaluation to a multi-cloud environment.

6.1 Numerical results based on a single cloud with a single TSP

We simulate three kinds of CSPs in the architecture: trustworthy CSPs, untrustworthy CSPs, and random CSPs. Trustworthy CSPs provide trustworthy services in most transactions, untrustworthy CSPs provide untrustworthy services in most transactions, and random CSPs provide trustworthy or untrustworthy services randomly. We assume that each CSP provides a single service on the same cloud. They all have initial trustworthy and untrustworthy degrees of 0.5 at time t_0 with highest uncertainty 1. We simulate 10000 CSUs, of which 80% are trustworthy, 10% are untrustworthy, and 10% are random. Trustworthy CSUs return true feedback for most transactions, untrustworthy CSUs return false feedback for most transactions, and random CSUs return true or untrue feedback randomly for all transactions. We carried all the simulations for 100 time window. The number of interactions of a CSU is uniformly distributed in [0, 20] for each time window. For each type of CSP we have the following assumptions.

Trustworthy CSP:

- The percentage of successful interactions in each time window: 90%,
- The percentage of failed interactions in each time window: 5%
- The percentage of uncertain interactions in each time window: 5%
- Trustworthy CSUs' rating: [0.8, 1]
- Untrustworthy CSUs' rating: [0, 0.5]
- Random CSUs' rating: [0, 1]

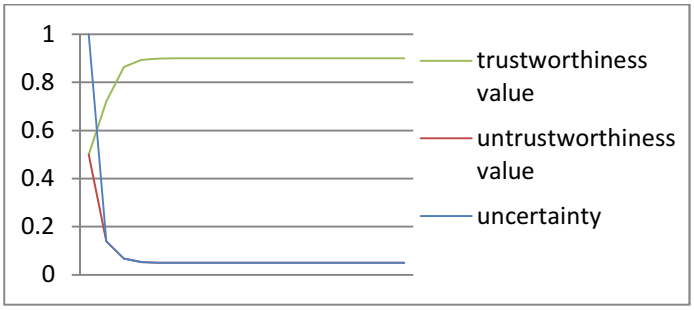
Untrustworthy CSP:

- The percentage of successful interactions in each time window: [0, 50%]
- The percentage of uncertain interactions in each time window: 10%
- The percentage of failed interactions in each time window: rest ones
- Trustworthy CSUs' rating: [0, 0.5]
- Untrustworthy CSUs' rating: [0.5, 1]
- Random CSUs' rating: [0, 1]

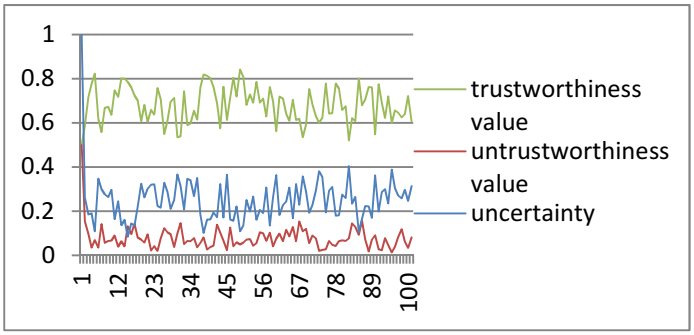
Random CSPs:

- The percentage of successful interactions in each time window: a= [0, 100%]

- The percentage of uncertain interactions in each time window: rand (a, 1)
- The percentage of failed interactions in each time window: rest ones
- Trustworthy CSUs' rating: [0.25, 0.75]
- Random CSU's rating: [0, 1]

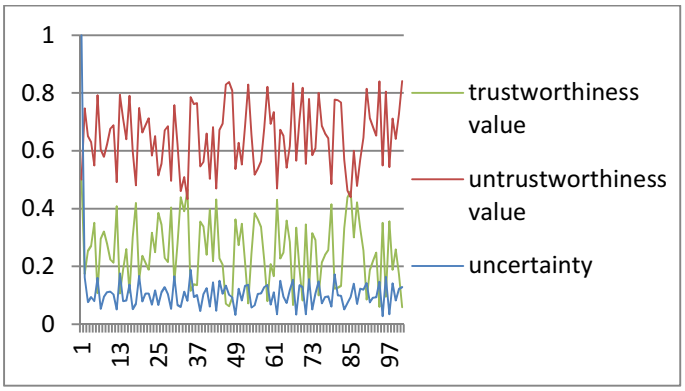


(a) GOT

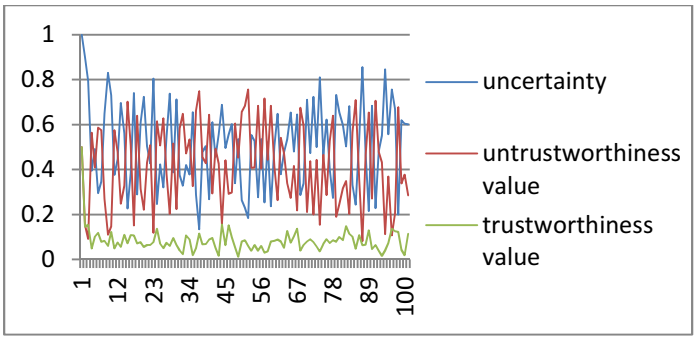


(b) GST

Figure 7: Evaluation results for trustworthy CSPs

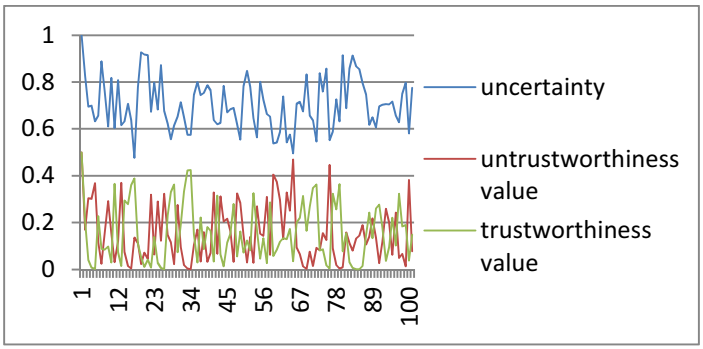


(a) GOT

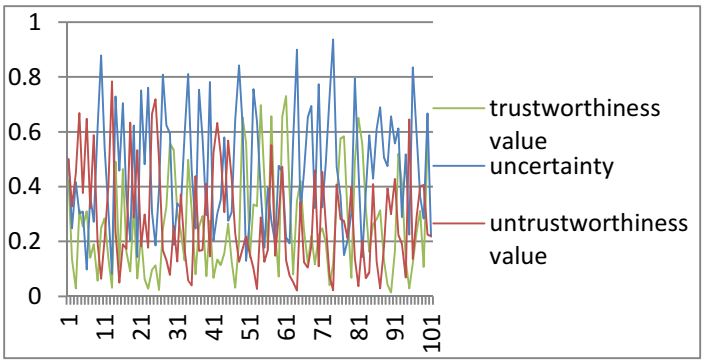


(b) GST

Figure 8: Evaluation results for untrustworthy CSPs



(a) GOT



(b) GST

Figure 9: Evaluation results for random CSPs

The results are shown in Figure 7, 8, and 9. Our goal is to show the effectiveness of our model to differentiate the trust levels of differerent types of CSPs. The results show that the trustworthy CSPs have relatively higher trustworthiness values and lower untrustworthiness and uncertainty values; the untrustworthy CSPs have relatively higher untrustworthiness values and lower trustworthiness and uncertainty values; and the random CSPs have relatively higher uncertainty values and lower trustworthiness and untrustworthiness values.

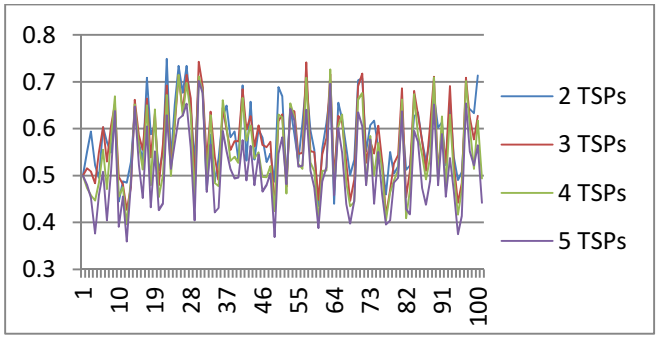
6.2 The numerical results based on multiple TSPs in a multi-cloud environment

In this part, we consider multiple clouds and TSPs and we assume that each cloud has a TSP and each TSP is only deployed in one cloud. For trustworthy and untrustworthy CSPs we considered three types of CSUs, i.e., trustworthy, untrustworthy, and random, but for random CSPs we only considered two types of CSUs, i.e., trustworthy and random ones. Each CSP provides the same service in more than one cloud. In this experiment, we simulate 5 clouds and 5 TSPs. All services are deployed on at least two clouds. They all have an initial trustworthy and untrustworthy degree of 0.5 at t_0 with highest uncertainty 1. For each cloud, we simulate 10000 CSUs. The simulation were carried out over 100 time windows, and the number of interactions of a CSU is uniformly distributed in [0,20] for each time window.

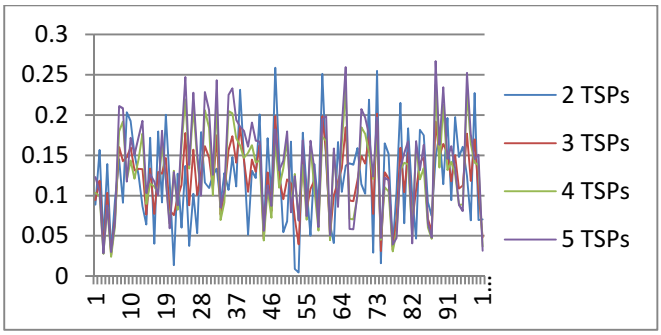
Figure 10 gives the GST values of a trustworthy CSP deployed on different numbers of clouds. The basic assumptions are shown in Table 1.

Table 1: The basic settings of trustworthy CSPs in different clouds

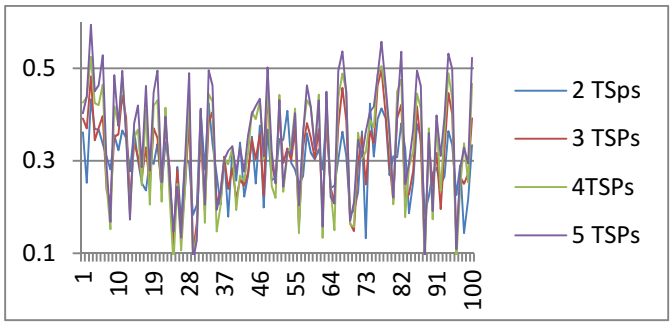
	Cloud 1	Cloud 2	Cloud 3	Cloud 4	Cloud 5
Range of ratings from trustworthy CSUs	[0.8, 1]	[0.8,0.9]	[0.7,1]	[0.7,0.9]	[0.8,1]
Range of ratings from untrustworthy CSUs	[0, 0.5]	[0.1, 0.5]	[0.1, 0.5]	[0, 0.5]	[0.1, 0.5]
Range of ratings from random CSUs	[0, 1]	[0, 1]	[0, 1]	[0, 1]	[0, 1]
Number of hops	0	1	2	3	4
Distribution of trustworthy, untrustworthy, and random CSUs	(80%, 10%, 10%)	(60%, 20%, 20%)	(70%, 20%, 10%)	(70%, 10%, 20%)	(80%, 10%, 10%)



(a) Trustworthiness values of a trustworthy CSP



(b) Untrustworthiness values of a trustworthy CSP



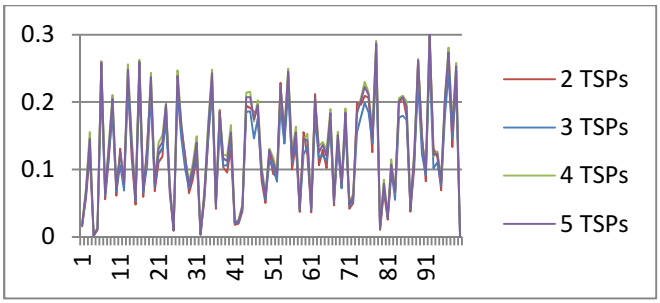
(c) Uncertainty values of a trustworthy CSP

Figure 10: GST values of a trustworthy CSP deployed on different numbers of clouds

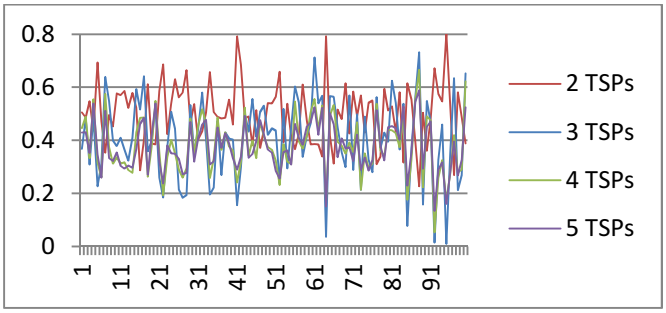
From Figure 10, we can see that a trustworthy CSP has a higher trustworthiness value and lower untrustworthiness and uncertainty values. And there is a slightly impact of the number of host clouds on the GST evaluation for the trustworthy CSP. That is, the trustworthiness value of the same trustworthy CSP decreases slightly as the number of host clouds increases. Also, the untrustworthiness and uncertainty value increases slightly as the number of host clouds increases.

Table 2: The Basic Settings to untrustworthy CSPs in different clouds

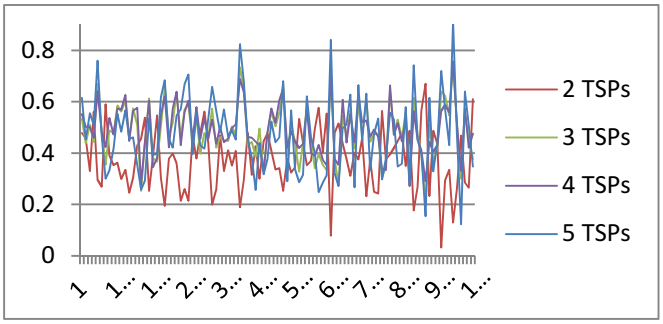
	Cloud 1	Cloud 2	Cloud 3	Cloud 4	Cloud 5
Range of ratings from Trustworthy CSUs	[0, 0.5]	[0, 0.5]	[0.1, 0.5]	[0, 0.4]	[0.1, 0.5]
Range of ratings from Untrustworthy CSUs	[0.5, 1]	[0.6, 1]	[0.5, 1]	[0.6, 1]	[0.6, 1]
Range of ratings from Random CSUs	[0, 1]	[0, 1]	[0, 1]	[0, 1]	[0, 1]
Number of hops	0	1	2	3	4
Distribution of Trustworthy, Untrustworthy, and Random CSUs	(80%, 10%, 10%)	(60%, 20%, 20%)	(70%, 20%, 10%)	(70%, 10%, 20%)	(80%, 10%, 10%)



(a) Trustworthiness value of an untrustworthy CSP



(b) Untrustworthiness value of an untrustworthy CSP



(c) Uncertainty value of an untrustworthy CSP

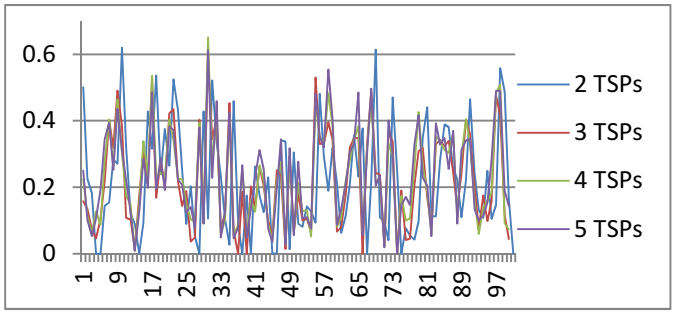
Figure 11: GST values of an untrustworthy CSP deployed on different numbers of clouds

Figure 11 gives the GST values of an untrustworthy CSP deployed on different numbers of clouds. The basic assumptions are as shown in Table 2. We can see that an untrustworthy CSP has a higher untrustworthiness value and lower trustworthiness and uncertainty values. There is a slightly impact of the number of host clouds on the GST evaluation for the untrustworthy CSP. There is even no change on the trustworthiness value as the number of host clouds increase, but the untrustworthiness value decreases slightly with the number of host clouds increasing, and the uncertainty value increases slightly with the number of host clouds increasing. From the above results we can see that as the number of host clouds increases, the uncertainty of the trust evaluation of untrustworthy CSPs increases, and the accuracy of the trustworthiness and untrustworthiness evaluation will somewhat deteriorate, but overall the model is effective and robust to these influences.

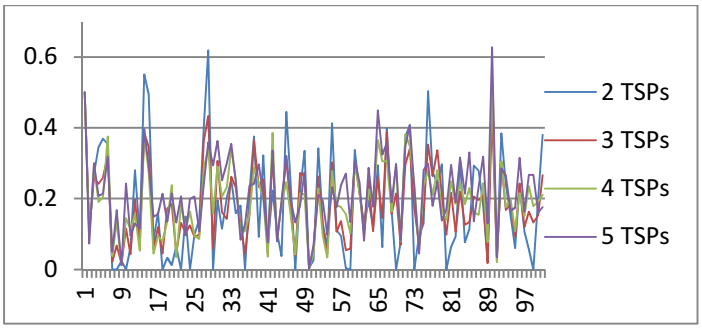
Figure 12 gives the GST values of a random CSP deployed on different numbers of clouds. The basic assumptions are as shown in Table 3.

Table 3: The basic settings to random CSPs in different clouds

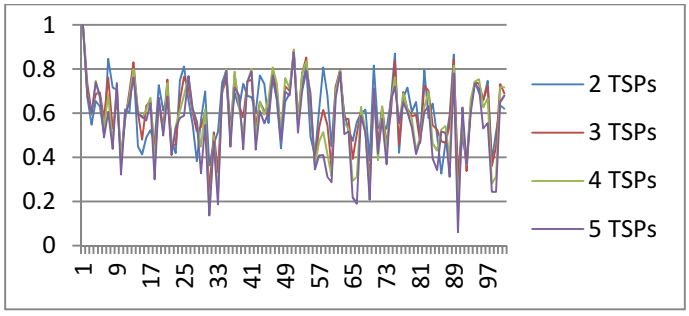
	Cloud 1	Cloud 2	Cloud 3	Cloud 4	Cloud 5
Range of ratings from Trustworthy CSUs	[0,1]	[0.25,0.75]	[0.1,0.9]	[0.2,0.8]	[0.15,0.85]
Range of ratings from random CSUs	[0, 1]	[0, 1]	[0, 1]	[0, 1]	[0, 1]
Number of hops	0	1	2	3	4
Distribution of Trustworthy and Random CSUs	(80%, 20%)	(60%, 30%)	(75%, 25%)	(70%, 30%)	(85%, 15%)



(a) Trustworthiness value of a random CSP



(b) Untrustworthiness value of a random CSP



(c) Uncertainty value of a random CSP

Figure 12: GST values of a random CSP deployed on different numbers of clouds

From Figure 12, we can see that a random CSP has lower trustworthiness and untrustworthiness values, which are around (0, 0.4), and higher uncertainty values, which are around 0.6 on average. For random CSPs, there is no obviously regular trend in the trust evaluation as the number of host clouds changes.

7 Conclusion

In this paper, we develop a novel trust management framework for a multi-cloud environment to effectively evaluate the trustworthiness of CSPs using subjective and objective trust. We propose a TSP-based trust management architecture, which is designed to perform the tasks of trust information eliciting, processing, and evaluation of CSPs in a multi-cloud environment. TSPs can derive the LST and LOT from a single CSU's perspective or the GST and GOT from the whole CSUs' aggregated perspective. In order to share the trust information of multi-cloud services across different clouds, a trust propagation network of TSPs is established. This is formed with each node establishing a trust relation with its neighbors. When A TSP receives a trust request for an unknown CSP, it floods this request through the trust propagation network. A TSP that has the required trust information responds to the originating TSP through the paths over which it received the trust request, referred to as TPoTs. In order to test the effectiveness of the proposed framework we conducted simulation experiments for a single TSP with a single cloud and for multiple TSPs in a multi-cloud environment. The experiments show that the proposed trust management framework is effective and robust scheme for differentiating trustworthy and untrustworthy CSPs.

Acknowledgment

This research work is supported by projects of Nature Science Foundation of China (Nos. 71131002, 71071045, and 71201042), and the Fifth Group of Special Postdoctoral Science Foundation Projects of China (No. 2012T50571).

References

3Tera Applogic, 3tera's Cloud Computing SLA goes live, March 31, 2009.

Abawajy, J., Determining service trustworthiness in intercloud computing environments. In Proceedings of 2009 10th International Symposium on Pervasive Systems, Algorithms, and Networks (ISPAN), pp. 784-788. 2009.

Abawajy, J., Establishing trust in hybrid cloud computing environments. In Proceedings of 2011 IEEE 10th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), pp. 118-125. 2011.

Abbadi, I. M. and Martin A., Trust in the Cloud, Information Security Technology Report, 16(3-4), pp. 108-114, 2011.

Armbrust, M., Fox, A., Griffith, R., Joseph, A. D., Katz, R., Konwinski, A., and Zaharia, M., A view of cloud computing. Communications of the ACM, 53(4), pp. 50-58. 2010.

Azzedin, F. and Maheswaran, M., A trust brokering system and its application to resource management in public-resource grids, In Proceedings of IEEE Parallel and Distributed Processing Symposium, 2004.

Bernstein, D. and Vij, D., Intercloud security considerations, In Proceedings of 2010 IEEE Second International Conference on Cloud Computing Technology and Science (CloudCom), pp. 537-544. 2010.

Beth, M. B. and Klein, B., Valuation of Trust in Open Network, In Proceeding of the European Symposium on Research in Security (ESORICS), Brighton: Springer-Verlag, pp. 3-18, 1994.

Buchegger, S. and Le Boudec, J. Y., A Robust Reputation System for P2P and Mobile Ad-Hoc Networks, In Proceedings of P2PEcon, 2004.

Buyya, R., Yeo, C., and Venugopal, S., Market-oriented Cloud Computing: Vision, Hype, and Reality for Delivering it Services as Computing Utilities. In Proceedings of IEEE 10th International Conference on High Performance Computing and Communications, pp. 5-13. 2008.

Buyya, R., Ranjan, R., and Calheiros, R. N., Intercloud: Utility-oriented federation of cloud computing environments for scaling of application services. In Proceedings of Algorithms and architectures for parallel, Springer Berlin Heidelberg, pp. 13-31. 2010.

Chandrasekar, A., Chandrasekar, K., Mahadevan, M., and Varalakshmi, P., QoS monitoring and dynamic trust establishment in the cloud. In Advances in Grid and Pervasive Computing, Springer Berlin Heidelberg, pp. 289-301. 2012.

Chen, K., Hwang, K., and Chen, G., Heuristic discovery of role-based trust chains in peer-to-peer networks. IEEE Transactions on Parallel and Distributed Systems, 20 (1), pp. 83-96. 2009.

Cho, J. H., Swami, A., and Chen, R., A survey on trust management for mobile ad hoc networks. Communications Surveys and Tutorials, IEEE 13, no. 4, pp. 562-583. 2011.

CSA, Cloud Controls Matrix, https://cloudsecurityalliance.org/research/initiatives/cloud-controls-matrix/, 2011.

De Cock, M. and Da Silva, P. P., A many valued representation and propagation of trust and distrust. In Fuzzy Logic and Applications, Springer Berlin Heidelberg, pp. 114-120. 2006.

Fan, W., Yang, S., and Pei, J., A novel two-stage model for cloud service trustworthiness evaluation, Expert systems, DOI: 10.1111/exsy.12017, 2013.

Fogel, J. and Nehmad, E., Internet social network communities: Risk taking, trust, and privacy concerns. Computers in Human Behavior, 25(1), pp. 153-160. 2009.

Global Inter-Cloud Technology Forum. Use Cases and Functional requirements for Inter-Cloud Computing. Technical Report, 2010.

Goyal, M. K., Aggarwal, A., Gupta, P., and Kumar, P., QoS based trust management model for Cloud IaaS. In Proceedings of Second IEEE International Conference on Parallel Distributed and Grid Computing (PDGC), pp. 843-847. 2012.

Grozev, N. and Buyya, R., Inter-Cloud architectures and application brokering: taxonomy and survey. Software: Practice and Experience, pp. 1-22. 2012.

Guha, R., Kumar, R., Raghavan, P., and Tomkins, A., Propagation of trust and distrust. In Proceedings of the 13th International Conference on World Wide Web, ACM, pp. 403-412. 2004.

Habib, S. M., Ries, S., and Muhlhauser, M., Cloud computing landscape and research challenges regarding trust and reputation, In proceedings of 7th International Conference on Ubiquitous Intelligence & Computing and Autonomic & Trusted Computing (UIC/ATC), pp. 410-415. 2010.

Habib, S. M., Ries, S., and Muhlhauser, M., Towards a trust management system for cloud computing. In Proceedings of 2011 IEEE 10th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), pp. 933-939. 2011.

Hang, C. W., Wang, Y., & Singh, M. P. Operators for propagating trust and their evaluation in social networks. In Proceedings of he 8th International Conference on Autonomous Agents and Multiagent Systems-Volume 2, pp. 1025-1032. 2009.

Haq, I. U., Brandic, I., and Schikuta, E., SLA validation in layered cloud infrastructures, Economics of Grids, Clouds, Systems, and Services. Springer-Verlag, pp. 153-164. 2010.

He, R., Hu, J., Niu, J., and Yuan, M., A novel cloud-based trust model for pervasive computing. In Proceedings of Fourth International Conference on Computer and Information Technology (CIT'04), pp. 693-700. IEEE Computer Society, 2004.

Hsu, J. Y. J., Lin, K. J., Chang, T. H., Ho, C. J., Huang, H. S., and Jih, W. R., Parameter learning of personalized trust models in broker-based distributed trust management. Information Systems Frontiers, 8 (4), pp. 321-333. 2010.

Jameel, H., Kalim, U., Sajjad, A., Lee, S., and Lee, Y. K., A trust model for ubiquitous systems based on vectors of trust values. In Proceedings of Seventh IEEE International Symposium on Multimedia, 2005.

Jøsang, A., A Logic for uncertain probabilities. International Journal of Uncertainty, Fuzziness and Knowledge based Systems, 9 (03), pp. 279-311. 2001.

Jøsang, A., Marsh, S., and Pope, S., Exploring different types of trust propagation. Trust management, Springer Berlin Heidelberg, pp. 179-192. 2006.

Khaled, M. K. and Qutaibah, M., Establishing trust in cloud computing, cloud computing, IT Professional, 12(5), pp. 20-27, 2010.

Ko, R. K., Jagadpramana, P., Mowbray, M., Pearson, S., Kirchberg, M., Liang, Q., and Lee, B. S., TrustCloud: A framework for accountability and trust in cloud computing. In Proceedings of 2011 IEEE World Congress on Services (SERVICES), pp. 584-588. 2011.

Lee, J. and Lin, K., Context-aware distributed reputation management system. In Proceedings of 2008 IEEE International Conference on e-Business Engineering. ICEBE'08, pp. 61-68. 2008.

Li, J., Li, R., and Jien K., Future trust management framework for mobile ad hoc networks. Communications Magazine, 46 (4), pp. 108-114. 2008.

Li, R., Li J., Liu P., and Chen H., An objective trust management framework for mobile ad hoc networks. In Proceedings of Vehicular Technology Conference, 2007. VTC2007-Spring., pp. 56-60. 2007.

Li, X. and Du, J., Adaptive and attribute-based trust model for service-level agreement guarantee in cloud computing, IET INFORMATION SECURITY, 7(2), pp: 144-154. 2013.

Lin, K. J., Lu, H., Yu, T., and Tai, C. E., A reputation and trust management broker framework for web applications. In Proceedings of the IEEE International Conference on e-Technology, e-Commerce, and e-Service, pp. 262–269, 2005.

Mohammed, A., Dillon, T., and Chang, E., SLA-based trust model for cloud computing. In Proceedings of 2010 13th International Conference on Network-Based Information Systems (NBiS), pp. 321-324. 2010.

Monsef, M. and Gidado, N., Trust and privacy concern in the Cloud. In Proceedings of 2011 European Cup, IT Security for the Next Generation, pp. 1-15, 2011.

Muchahari, M. K. and Sinha, S. K., A New Trust Management Architecture for Cloud Computing Environment. In Proceedings of 2012 International Symposium on Cloud and Services Computing (ISCOS), pp. 136-140. 2012.

Ngo, C., Demchenko, Y., and de Laat, C., Toward a Dynamic Trust Establishment approach for multi-provider Intercloud environment. In Proceedings of 2012 IEEE 4th International Conference on Cloud Computing Technology and Science (CloudCom), pp. 532-538. 2012.

Ozaa, N. V., Halla, T., and Rainera, A., Trust in software outsourcing relationships: An empirical investigation of Indian software companies. Information and Software Technology, 48(5), pp. 245-354, 2006.

Sabater, J. and Sierra, C., Regret: a reputation model for gregarious societies. In Proceedings of the 1st International Joint Conference on Autonomous Agents and Multi-Agents Systems, 2002.

Singh, A., Trust and Trust Management Models for Ecommerce & Sensor Network. International Journal of Engineering Research and Applications (IJERA), 2012.

Song, S. and Hwang, K., Fuzzy trust integration for security enforcement in grid computing, In Proceedings of the Int'l Symposium on Network and Parallel Computing. LNCS 3222, Berlin: Springer-Verlag, pp. 9-21, 2005.

Sun, Y. L., Han, Z., Yu, W., and Liu, K., R., A Trust Evaluation Framework in Distributed Networks: Vulnerability Analysis and Defense Against Attacks, In Proceedings of 2006 IEEE INFOCOM, 2006.

Teacy, W. T., Luck, M., Rogers, A., and Jennings, N. R., An efficient and versatile approach to trust and reputation using hierarchical Bayesian modeling. Artificial Intelligence, 2012.

Tong, X., Zhang W., Yu, L., and Huang, H., Subjectivity and Objectivity of Trust. In Proceedings of Agents and Data Mining Interaction, Springer Berlin Heidelberg, pp. 105-114. 2013.

Varalakshmi, P., Selvi, S. T., and Pradeep, M., A multi-broker trust management framework for resource selection in grid. In Proceedings of 2nd International Conference on Communication Systems Software and Middleware, pp. 1-6, 2007.

Wang, W., Zeng, G., Zhang, J., and Tang, D., Dynamic trust evaluation and scheduling framework for cloud computing. Security and Communication Networks, 5 (3), pp. 311-318, 2012.

Wang, Y. and Vassileva, J., Bayesian network-based trust model. In Proceedings of IEEE/WIC International Conference on Web Intelligence, pp. 372-378, 2003.

Witkowski, M. and Pitt, J., Objective trust-based agents: Trust and trustworthiness in a multi-agent trading society, In Proceedings of Fourth International Conference on Multi Agent Systems, pp. 463-464, 2000.

Yuan, W., Guan, D., Lee, Y., Lee, S., and Sung, J., Improved trust-aware recommender system using small-worldness of trust networks. Knowledge-Based Systems, 23 (3), pp. 232-238, 2010.

Zhang, Q., Yu, T., and Keith, I., A Classification Scheme for Trust Functions in Reputation-Based Trust Management. In Proceedings of ISWC Workshop on Trust, Security, and Reputation on the Semantic Web. 2004.