Trust Management of Services in Cloud Environments: Obstacles and Solutions

TALAL H. NOOR[1], QUAN Z. SHENG[1]
The University of Adelaide
and
SHERALI ZEADALLY[2]
University of the District of Columbia
and
JIAN YU[3]
Swinburne University of Technology

Trust management is one of the most challenging issues in the emerging cloud computing area. Over the past few years, many studies have proposed different techniques to address trust management issues. However, despite these past efforts, several trust management issues such as identification, privacy, personalization, integration, security, and scalability have been mostly neglected and need to be addressed before cloud computing can be fully embraced. In this article, we present an overview of the cloud service models and we survey the main techniques and research prototypes that efficiently support trust management of services in cloud environments. We present a generic analytical framework that assesses existing trust management research prototypes in cloud computing and relevant areas using a set of assessment criteria. Open research issues for trust management in cloud environments are also discussed.

Categories and Subject Descriptors: K.6.5 [**Management of Computing and Information Systems**]: Security and Protection (D.4.6, K.4.2); H.3.5 [**Information Storage and Retrieval**]: On-line Information Services—*Web-based services*

General Terms: Reliability, Security, Theory

Additional Key Words and Phrases: Trust management, cloud computing, service-oriented computing, credentials, policy, credibility, reputation, trust prediction, security, privacy

1. INTRODUCTION

Over the past few years, cloud computing has been receiving much attention as a new computing paradigm for providing flexible and on-demand infrastructures,

Authors' addresses: [1]School of Computer Science, The University of Adelaide, Adelaide SA 5005, Australia; emails: {talal, qsheng}@cs.adelaide.edu.au. [2]Department of Computer Science and Information Technology, The University of the District of Columbia, Washington, DC 20008; email: szeadally@udc.edu. [3]Faculty of Information and Communication Technologies, Swinburne University of Technology, John Street, Hawthorn, Melbourne, Victoria 3122, Australia; email: jianyu@swin.edu.au.

Permission to make digital/hard copy of all or part of this material without fee for personal or classroom use provided that the copies are not made or distributed for profit or commercial advantage, the ACM copyright/server notice, the title of the publication, and its date appear, and notice is given that copying is by permission of the ACM, Inc. To copy otherwise, to republish, to post on servers, or to redistribute to lists requires prior specific permission and/or a fee.
© 2013 ACM 0360-0300/13/00-0001 $5.00

platforms and software as services. Cloud computing has emerged as a result of combining the benefits of grid computing [Foster et al. 2008] with those of service-oriented computing [Wei and Blake 2010] to utilize computer resources (*data centers*) and deliver computer resources as services. In the case of grid computing, computer resources are combined from several virtual organizations to achieve a certain goal (e.g., high performance and reduced costs), while in the case of service-oriented computing, computer software is designed and governed in the form of services. With cloud computing, computer resources are designed and governed in the form of services using virtualization techniques (e.g., the creation of virtual instances of the hardware platform, the operating system or the storage of network resources) to automate business logics since distributed systems are available for both public and private sectors. Cloud environments promise several benefits such as reduced expenses and simplicity to service providers and service requesters [Foster et al. 2008; Sotomayor et al. 2009]. For instance, it only took 24 hours, at the cost of merely $240, for the New York Times to archive its 11 million articles (1851-1980) using a cloud service named Amazon Web Services [Gottfrid 2007].

Given the accelerated adoption of cloud computing in the industry, trust management is still considered as one of the key challenges in the adoption of cloud computing. Indeed, according to the researchers at UC Berkeley [Armbrust et al. 2010], trust management and security are ranked among the top 10 obstacles for adopting cloud computing. This is because of challenging issues such as privacy [Cavoukian 2008; Bertino et al. 2009] (e.g., the leakage of Apple's iPad subscribers' information[1]), security [Hwang and Li 2010; Viega 2009] (e.g., the mass email deletions of Gmail[2]), and dependability [Hwang et al. 2009] (e.g., Amazon Web Services outage that took down lots of business web sites[3]). In addition, the highly dynamic, distributed, and non-transparent nature of cloud services makes trust management even more challenging [Armbrust et al. 2010; Hwang and Li 2010; Noor and Sheng 2011b; Pearson and Benameur 2010].

An effective trust management system helps cloud service providers and consumers consumers reap the benefits brought about by cloud computing technologies. Despite the benefits of trust management, several issues related to general trust assessment mechanisms, distrusted feedbacks, poor identification of feedbacks, privacy of participants and the lack of feedbacks integration still need to be addressed. Traditional trust management approaches such as the use of Service Level Agreement (SLA) are inadequate for complex cloud environments. The vague clauses and unclear technical specifications of SLAs can lead cloud service consumers to be unable to identify trustworthy cloud services [Habib et al. 2011].

To the best of our knowledge, this is the first comprehensive survey that focuses on the trust management of services in cloud environments. In this work, we survey the main techniques, frameworks, and research prototypes on trust management in cloud computing and its most relevant areas. We propose a generic framework that considers a holistic view of the issues related to the trust management for interactions in cloud environments. In particular, we differentiate the trust management

[1]http://techcrunch.com/2010/06/15/ipad-breach-personal-data/
[2]http://www.techcrunch.com/2006/12/28/gmail-disaster-reports-of-mass-email-deletions/
[3]http://bits.blogs.nytimes.com/2011/04/21/amazon-cloud-failure-takes-down-web-sites/

perspectives and classify trust management techniques into four categories. We compare thirty representative trust management research prototypes in cloud computing and the relevant research areas using the proposed analytical framework. The framework consists of three layers and for each layer, we identify a set of dimensions (i.e., assessment criteria), which are used as a benchmark, to study these research prototypes. Several major cloud service providers are also compared.

The remainder of the article is organized as follows. The related work is discussed in Section 2. In Section 3 and Section 4, we present an overview of cloud services and their deployment models, and trust management techniques, respectively. In Section 5, we propose an analytical framework for trust management and identify a set of dimensions for each layer in the framework, which are used for comparing trust management solutions. In Section 6, we discuss and evaluate thirty representative research prototypes. In Section 7, we also compare several major cloud service providers from a trust perspective. In Section 8, we highlight some future directions for research and development. Finally, we offer some concluding remarks in Section 9.

2. RELATED WORK

Trust management is one of the most important issues in the area of information security and several surveys have been conducted. One of the first few surveys that address trust issues is done by Grandison and Sloman [Grandison and Sloman 2000]. This survey overviews trust definitions from *computer science*, *economic*, and *social psychology* perspectives. It also summarizes the trust relationship properties and trust classes that represent different types of trust. Suryanarayana and Taylor [Suryanarayana and Taylor 2004] classify trust management into three types, namely *policy-based*, *reputation-based*, and *social network-based*. The authors compare nine trust management systems based on eleven different criteria parameters. Ruohomaa and Kutvonen [Ruohomaa and Kutvonen 2005] overviews several trust models. They define trust actors and classify trust management into three tasks, including i) initialization of trust relationships, ii) behavior observation and iii) actions after a new experience. Artz and Gil [Artz and Gil 2007] compare several trust definitions for different research areas in the field of computer science. In particular, the authors discuss the relevance of trust and the semantic Web and point out some unique trust management challenges for the area. Finally, Fernandez-Gago et al. [Fernandez-Gago et al. 2007] perform a trust management survey focusing on wireless sensor networks. The survey overviews existing trust management solutions for ad-hoc and the peer-to-peer (P2P) wireless sensor networks.

A few surveys focus on reputation-based trust management systems. For instance, Marti and Garcia-Molina [Marti and Garcia-Molina 2006] exploit a taxonomy technique to classify different reputation-based trust management systems. Sabater and Sierra [Sabater and Sierra 2005] overview the reputation-based trust management and investigate the relationship between existing solutions and agent-based perspective. Agent-based or multi-agent trust and reputation systems use an artificial intelligence approach where autonomous and intelligent software agents are used to observe and search for trustworthy entities in order to make better decisions. Jøsang et al. [Jøsang et al. 2007] discuss general ideas of trust (e.g.,

trust classes and trust purpose) and explain the overlapping notions between trust and reputation terms. A few trust models are compared in the survey. Silaghi et al. [Silaghi et al. 2007] investigate whether existing trust management approaches can be applied to Grid environments. A few guidelines are given in the survey that may be useful to future research and the development of trust management systems in Grids. Wang and Vassileva [Wang and Vassileva 2007] present a systematic review of several trust and reputation systems. They classify these systems into three categories including centralized versus decentralized, persons/agents versus resources, and global versus personalized. A few potential research directions are given in the survey that help develop trustworthy Web services. In [Hoffman et al. 2009], Hoffman et al. survey several attacks and defense mechanisms of reputation systems, particularly in P2P environments. They specify the reputation system's components and classify attacks against each component. Various defense mechanisms are also proposed.

Most of the recent surveys lack a holistic view on trust management techniques (e.g., policy, reputation, recommendation, prediction). In particular, trust management issues such as distrusted feedbacks, poor identification of trust feedbacks, privacy of trust participants, and the lack of trust feedbacks integration have not been fully discussed. In contrast, our survey compares thirty representative trust management research prototypes based on fourteen different dimensions (i.e., assessment parameters). Our work specifically focuses on trust management issues in cloud environments, which makes original contributions by presenting trust management perspectives, a classification of various trust management techniques and an analytical framework for trust management prototypes assessment.

3. OVERVIEW OF SERVICES IN CLOUD ENVIRONMENTS

Cloud services are established based on five essential characteristics [Mell and Grance 2011], namely, i) *on-demand self-service* where cloud service consumers are able to automatically provision computing resources without the need for human interaction with each cloud service provider, ii) *broad network access* where cloud service consumers can access available computing resources over the network, iii) *resource pooling* where computing resources are pooled to serve multiple cloud service consumers based on a multi-tenant model where physical and virtual computing resources are dynamically reassigned on-demand, iv) *rapid elasticity* where computing resources are elastically provisioned to scale rapidly based on the cloud service consumers need, and v) *measured service* where computing resources usage is monitored, metered (i.e., using pay as you go mechanism), controlled and reported to provide transparency for both cloud service providers and consumers.

3.1 Cloud Service Models

Cloud services have three different models, including *Infrastructure as a Service* (IaaS), *Platform as a Service* (PaaS), and *Software as a Service* (SaaS) based on different Service Level Agreements (SLAs) between a cloud service provider and a cloud service consumer [Brandic et al. 2010; Clark et al. 2010; Mell and Grance 2011]. Figure 1 depicts the structured layers of cloud services:

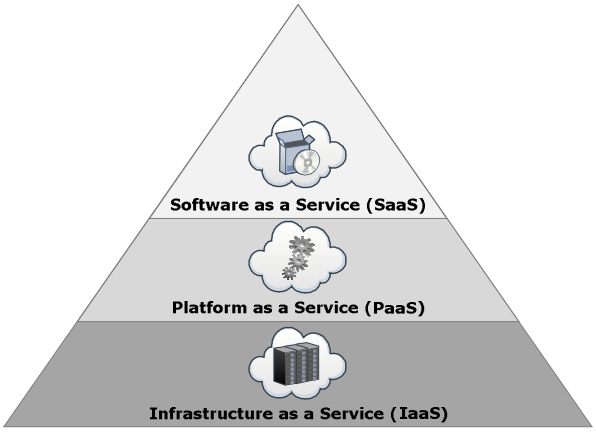


Fig. 1. Cloud Service Models

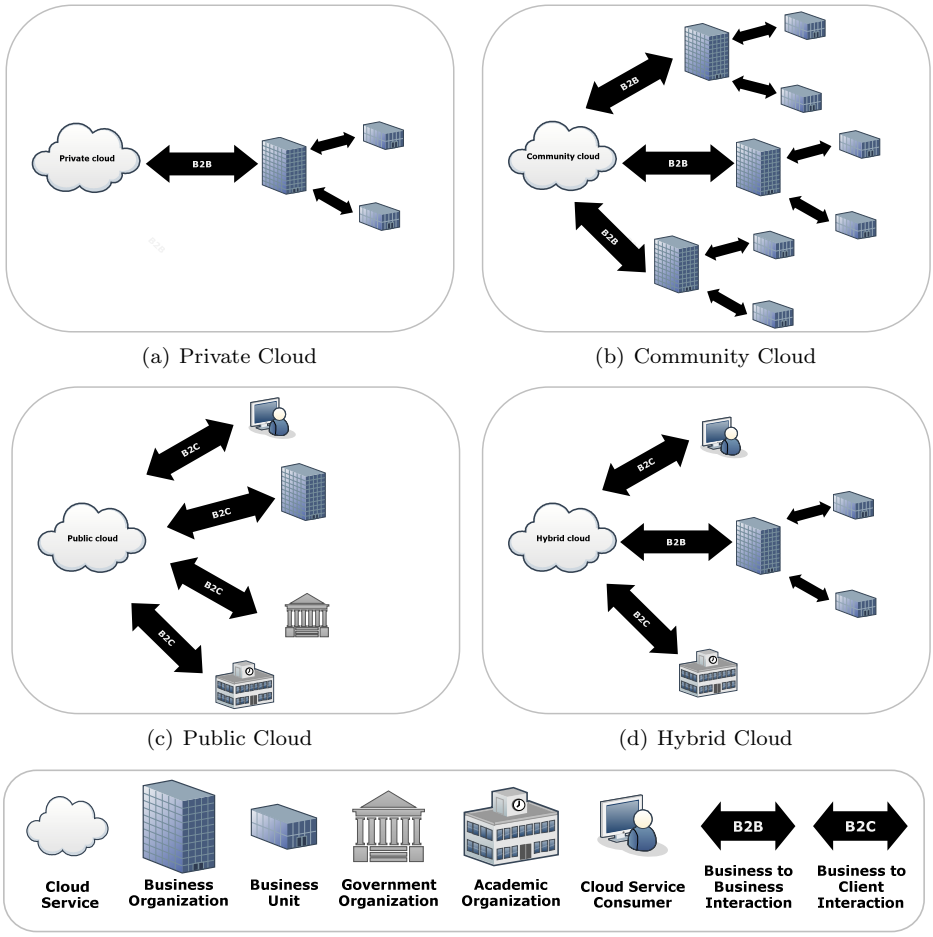
—*Infrastructure as a Service (IaaS).* This model represents the foundation part of the cloud environment where a cloud service consumer can rent the storage, the processing and the communication through virtual machines provided by a cloud service provider (e.g., Amazon's Elastic Compute Cloud (EC2) [Amazon-EC2 2011] and Simple Storage Service (S3) [Amazon-S3 2011]). In this model, the cloud service provider controls and manages the underlying cloud environment, whereas the cloud service consumer has control over his/her virtual machine which includes the storage, the processing and can even select some network components for communication.

—*Platform as a Service (PaaS).* This model represents the integration part of the cloud environment and resides above the IaaS layer to support system integration and virtualization middleware. The PaaS allows a cloud service consumer to develop his/her own software where the cloud service provider provisions the software development tools and programming languages (e.g., Google App [Google-Apps 2011]). In this model, the cloud service consumer has no control over the underlying cloud infrastructure (e.g., storage network, operating systems, etc.) but has control over the deployed applications.

—*Software as a Service (SaaS).* This model represents the application part of the cloud environment and resides above the PaaS layer to support remote accessibility where cloud service consumers can remotely access their data which is stored in the underlying cloud infrastructure using applications provided by the cloud service provider (e.g., Google Docs [Google-Docs 2011], Windows Live Mesh [Microsoft 2011]). Similarly, in this model, the cloud service consumer has no control over the underlying cloud infrastructure (e.g., storage network, operating systems, etc.) but has control over his/her data.

3.2 Cloud Service Deployment Models

Based on the Service Level Agreement (SLA), all cloud service models (i.e., IaaS, PaaS, SaaS) can be provisioned through four different cloud service deployment



(a) Private Cloud (b) Community Cloud

(c) Public Cloud (d) Hybrid Cloud

Fig. 2. Cloud Service Deployment Models

models, namely *Private*, *Community*, *Public*, and *Hybrid* [Mell and Grance 2011; Sotomayor et al. 2009] depending on the cloud service consumer's needs. Figure 2 depicts how cloud services are arranged to support these four cloud services deployment models and shows different interactions between cloud service providers and consumers. The interactions include business-to-business (B2B) and business-to-client (B2C).

—*Private Cloud.* In this deployment model, computing resources are provisioned for a particular organization (e.g., a business organization as shown in Figure 2(a)), which involves several consumers (e.g., several business units). Essentially, interactions in this deployment model are considered as B2B interactions where the computing resources can be owned, governed, and operated by the same organization, a third party, or both.

—*Community Cloud.* In this deployment model, computing resources are provisioned for a community of organizations, as shown in Figure 2(b), to achieve

a certain goal (e.g., high performance, security requirements, or reduced costs). Basically, interactions in this model are considered as B2B interactions where the computing resources can be owned, governed, and operated by the community (i.e., one or several organizations in the community), a third party, or both.

—*Public Cloud.* In this deployment model, computing resources are provisioned for the public (e.g., an individual cloud service consumer, academic, government, business organizations or a combination of these cloud service consumer types as shown in Figure 2(c)). Essentially, interactions in this model are considered as B2C where the computing resources can be owned, governed, and operated by an academic, government, or business organization, or a combination of them.

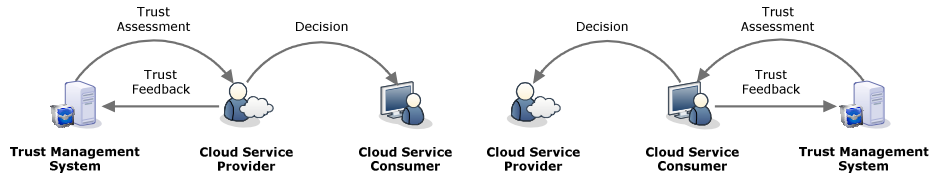
—*Hybrid Cloud.* In this deployment model, computing resources are provisioned using two or more deployment models (e.g., private and public clouds can be deployed together using a hybrid deployment model as shown in Figure 2(d)). Basically, interactions in this model include B2B and B2C interactions where computing resources are bound together by different clouds (e.g., private and public clouds) using portability techniques (e.g., data and application portability such as cloud bursting for load balancing between clouds).

Given all possible service and deployment models and interactions in cloud environments, we argue that there is no one trust management solution that fits all cloud services. A trust management service may be independent of cloud services but the trust techniques and assessment functions need to suit the underlying cloud service models. We believe that it is vital to know what are the possible trust management techniques and to identify which types of cloud services these techniques support well in order to give insights on how to develop the most suitable trust management solution for each type of cloud services. In the following section, we differentiate the trust management perspectives, classify the trust management techniques and present several examples for trust management systems in cloud environments.

4. OVERVIEW OF TRUST MANAGEMENT

Trust management is originally developed by Blaze et. al [Blaze et al. 1996] to overcome the issues of centralized security systems, such as centralized control of trust relationships (i.e., global certifying authorities), inflexibility to support complex trust relationships in large-scale networks, and the heterogeneity of policy languages. Policy languages in trust management are responsible for setting authorization roles and implementing security policies. Authorization roles are satisfied through a set of security policies, which themselves are satisfied through a set of credentials. Some early attempts to implementing the trust management are PolicyMaker and KeyNote [Blaze et al. 1998; Blaze et al. 1998; Blaze et al. 1999; Blaze et al. 2000]. These techniques are considered as policy-based trust management because they rely on policy roles to provide automated authorizations. Later, trust management inspired many researchers to specify the same concept in different environments such as e-commerce, P2P systems, Web services, wireless sensor networks, grid computing, and most recently cloud computing.

Trust management is an effective approach to assess and establish *trusted relationships*. Several approaches have been proposed for managing and assessing trust based on different perspectives. We classify trust management using two



(a) Service Provider's Perspective (SPP) (b) Service Requester's Perspective (SRP)

Fig. 3. Trust Management Perspectives

different perspectives, namely: *Service Provider Perspective* (SPP) and *Service Requester Perspective* (SRP). In SPP, the service provider is the main driver of the trust management system where service requesters' trustworthiness is assessed (Figure 3(a)). On the other hand, in SRP, the service requester is the one who assesses the trustworthiness of the service provider (Figure 3(b)).

4.1 Trust Management Techniques

Different trust management techniques have been reported in the literature, which can be classified into four different categories: *Policy*, *Recommendation*, *Reputation*, and *Prediction*. To ease the discussion, we focus on explaining these trust management techniques using the service requester perspective (i.e., cloud service consumers perspective). The same techniques can be applied to the other perspective (i.e., cloud service providers perspective).

Figure 4 depicts the four trust management techniques. Cloud service consumers and providers are connected with lines representing trusted relations between them (denoted $\mathcal{T}_r$). The values of $\mathcal{T}_r$ can be either 0 (the trusted relationship does not exist) or 1 (the trusted relationship exists). An unrecognized relation, denoted in a dashed line, occurs when a cloud service consumer x approaches a cloud service provider y for the first time.

4.1.1 *Policy as a Trust Management Technique (PocT).* Policy as a trust management technique (PocT) is one of the most popular and traditional ways to establish trust among parties and has been used in cloud environments [Yao et al. 2010; Santos et al. 2009; Alhamad et al. 2010], the grid [Song et al. 2005], P2P systems [Song et al. 2005], Web applications [De Capitani di Vimercati et al. 2012] and the service- oriented environment [Skogsrud et al. 2007; Skogsrud et al. 2009]. PocT uses a set of policies and each of which assumes several roles that control authorization levels and specifies a minimum trust threshold in order to authorize access. The trust thresholds are based on the trust results or the credentials.

For the trust results-based threshold, several approaches can be used. For instance, the *monitoring and auditing* approach proves Service Level Agreement (SLA) violations in cloud services (i.e., if the SLA is satisfied, then the cloud service is considered as trustworthy and vise versa). The *entities credibility* approach specifies a set of parameters to measure the credibility of parties [Huynh et al. 2006] while the *feedback credibility* approach considers a set of factors to measure the credibility of feedbacks. SLA can be considered as a service plan (i.e., where the service level is specified) and as a service assurance where penalties can be

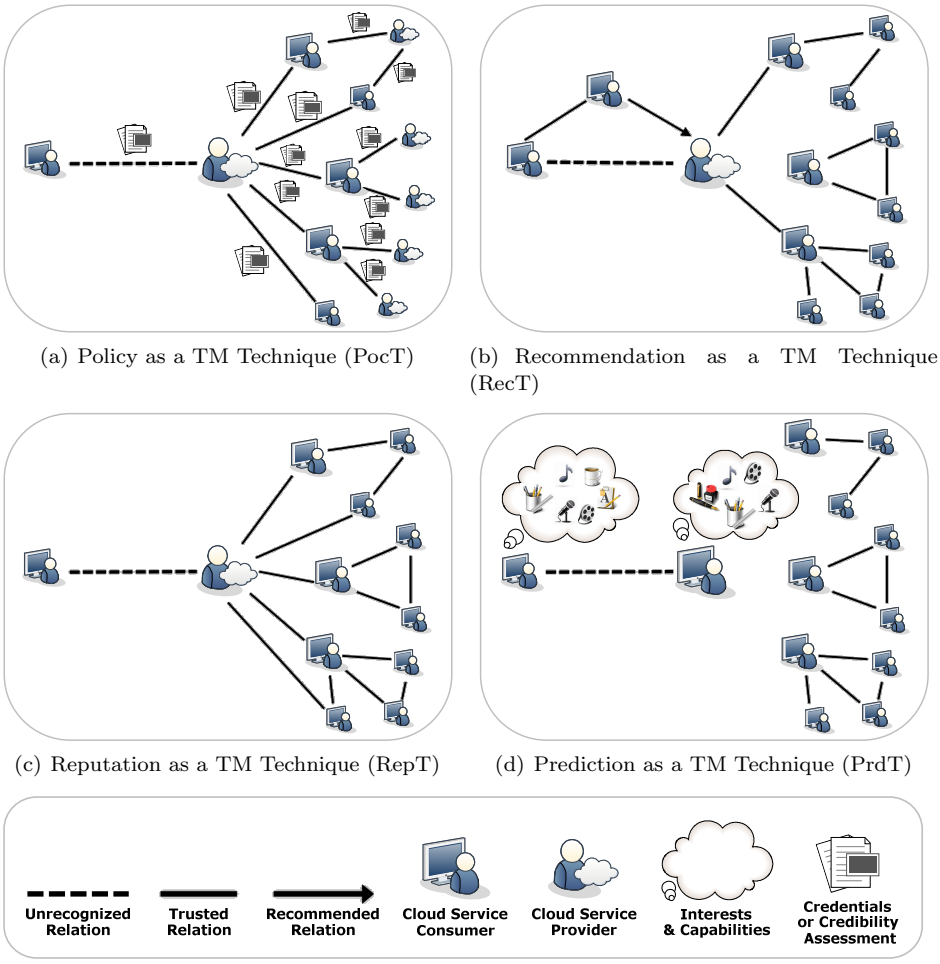
assigned to the cloud service provider if there is a service level violation in the provisioned cloud services. SLA can establish trust between cloud service consumers and providers by specifying technical and functional descriptions with strict clauses. The entities credibility (i.e., the credibility of cloud services) can be measured from qualitative and quantitative attributes such as security, availability, response time, and customer support [Habib et al. 2011]. The feedback credibility [Xiong and Liu 2003] can be measured using several factors such as cloud service consumers' experience (i.e., the quality of feedbacks differs from one person to another [Noor and Sheng 2011a]). Many researchers identify two features of credibility including trustworthiness and expertise [Xiong and Liu 2004; Srivatsa et al. 2005; Malik and Bouguettaya 2009a; Al-Sharawneh and Williams 2010; Noor and Sheng 2011a].

For credential-based threshold, PocT follows either the Single- Sign-On (SSO) approach [Pashalidis and Mitchell 2003] where the credentials disclosure and authentication take place once and then the cloud service consumers have an access approval for several cloud services, or the state machine approach [Thomas and Hun 2002] where the credentials disclosure and authentication take place for each state of the execution of cloud services. Credentials are generally established based on standards such as the X.509v3 [Cooper et al. 2008], the Simple Public Key Infrastructure (SPKI) [Ellison et al. 1999], or the Security Assertion Markup Language (SAML) [Cantor et al. 2005]. Many researchers use the digital certificates perspective to define the credential term [Seamons et al. 2001; Camenisch and Van Herreweghen 2002; Bertino et al. 2004] where a trusted third party (i.e., certificate authority) is required to certify the credential. However, not all credentials require a trusted certificate authority for establishing identities such as the Simple Public Key Infrastructure (SPKI) credentials [Ellison 1996] where the certificate authority is not required.

Figure 4(a) depicts how PocT is arranged to support trust management in the cloud environment. A cloud service consumer x has certain policies $\mathcal{P}_x$ to control the disclosure of its own credentials $\mathcal{C}_x$ and contains the minimum trust threshold $\mathcal{T}_x$. $\mathcal{T}_x$ can either follow the credentials approach or the credibility approach, depending on the credibility assessment of the cloud service provider y (denoted $\mathcal{R}_y$) to determine whether to proceed with the transaction. In contrast, the cloud service provider y also has certain policies $\mathcal{P}_y$ to regulate access to its cloud services (e.g., IaaS, PaaS, SaaS), to control the disclosure of its own credentials $\mathcal{C}_y$ and contains the minimum trust threshold $\mathcal{T}_y$. Similarly, $\mathcal{T}_y$ can either follow the credential approach or the credibility approach, depending on the credibility assessment of the cloud service consumer x (denoted $\mathcal{R}_x$). If both trust thresholds are satisfied (i.e., $\mathcal{T}_x$ and $\mathcal{T}_y$), the relation between the cloud service consumer x and provider y is considered as a trusted relation (i.e., $\mathcal{T}r\,(x,\,y) = 1$ as shown in Equation 1).

$$\mathcal{T}r\,(x,\,y) = \begin{cases} 1 & if \quad \mathcal{C}_x \geq \mathcal{T}_y \Leftrightarrow \mathcal{C}_y \geq \mathcal{T}_x \;\; or \;\; \mathcal{R}_y \geq \mathcal{T}_x \Leftrightarrow \mathcal{R}_x \geq \mathcal{T}_y \\ 0 & otherwise \end{cases} \tag{1}$$

The literature reports some efforts of PocT in cloud computing. For example, Brandic et al. [Brandic et al. 2010] propose a novel language for specifying compliance requirements based on a model-driven technique and Ko et al. [Ko et al. 2011] present a TrustCloud framework that uses SLA detective controls and monitoring



(a) Policy as a TM Technique (PocT)

(b) Recommendation as a TM Technique (RecT)

(c) Reputation as a TM Technique (RepT)

(d) Prediction as a TM Technique (PrdT)

Fig. 4. Trust Management (TM) Techniques

techniques for achieving trusted cloud services. Hwang et al. [Hwang et al. 2009; Hwang and Li 2010] propose a security aware cloud architecture that uses predefined policies to evaluate the credibility of cloud services and Habib et al. [Habib et al. 2011] develop a multi-faceted Trust Management (TM) system to measure the credibility of cloud services based on quality of service (QoS) attributes such as security, latency, availability, and customer support. Finally, Noor and Sheng [Noor and Sheng 2011b; 2011a] propose a credibility model that distinguishes credible feedbacks from the misleading ones. PocT is applicable for all three cloud service models.

4.1.2 *Recommendation as a Trust Management Technique (RecT).* Recommendation as a trust management technique (RecT) has been widely used in the cloud environment [Habib et al. 2011; Krautheim et al. 2010], the grid [Domingues et al. 2007], and the service oriented environment [Skopik et al. 2009; Park et al. 2005].

Recommendations take advantage of participants knowledge about the trusted parties, especially given that the party at least knows the source of the trust feedback. It is well known in the social psychology theory that the role of a person has a considerable influence on another person's trust assessment if a recommendation is given [Liu et al. 2009]. Recommendations can appear in different forms such as the *explicit recommendation* or the *transitive recommendation*. An explicit recommendation happens when a cloud service consumer clearly recommends a certain cloud service to her well-established and trusted relations (e.g., friends). A transitive recommendation happens, on the other hand, when a cloud service consumer trusts a certain cloud service because at least one of her trusted relations trust the service.

Figure 4(b) depicts the RecT approach where the cloud service consumer x has a trusted relation with another cloud service consumer z. Essentially, the cloud service consumer z recommends consumer x to cloud service provider y, or x transitively trusts y because there is a trusted relation between z and y. In other words, because the cloud service consumer x trusts the other cloud service consumer z, it is more likely that x will trust the recommended relation (i.e., the cloud service provider y), $\mathcal{T}r\,(x,\, y \mid \mathcal{T}r\,(z,\, y)) = 1$ as shown in Equation 2.

$$\mathcal{T}r\,(x,\, y \mid \mathcal{T}r\,(z,\, y)) = \begin{cases} 1 & if \quad \mathcal{T}r\,(z,\, y) = 1 \\ 0 & otherwise \end{cases} \tag{2}$$

One of the recent efforts using RecT in cloud computing is reported in [Habib et al. 2011]. In the work, trust is derived from recommendations using several operations including *consensus* (i.e., where trust feedbacks are aggregated from different cloud service consumers) and *discounting* (i.e., where trust feedbacks are weighted based on the trustworthiness of cloud service consumers). In [Krautheim et al. 2010], a cloud trust model is proposed based on transitive trust where a chain of trusted relations is built from a single root of trust. Similarly, RecT is applicable for all three cloud service models.

4.1.3 *Reputation as a Trust Management Technique (RepT).* Reputation as a trust management technique (RepT) is important because the feedback of the various cloud service consumers can dramatically influence the reputation of a particular cloud service either positively or negatively. RepT has been used in the cloud environment [Habib et al. 2011; Noor and Sheng 2011b; 2011a; Krautheim et al. 2010; Manuel et al. 2009], the grid [Azzedin and Maheswaran 2002b; 2002a; 2004; Lin et al. 2004], P2P [Xiong and Liu 2004; Srivatsa et al. 2005; Srivatsa and Liu 2006; Aringhieri et al. 2005; Aringhieri et al. 2006; Zhou and Hwang 2006; 2007; Kamvar et al. 2003; Damiani et al. 2003; Damiani et al. 2002], as well as the service-oriented environment [Park et al. 2005; Conner et al. 2009; Malik and Bouguettaya 2009c; 2009b; 2009a]. Reputation can have direct or indirect influence on the trustworthiness of a particular entity (e.g., cloud service) as pointed in [Al-Sharawneh and Williams 2010]. Unlike RecT, in RepT, cloud service consumers do not know the source of the trust feedback, i.e., there is no trusted relations in RepT, see Figure 4(c) and 4(b). There are several online reputation-based systems such as the auction systems (e.g., eBay [eBay 2011] and Amazon [Amazon 2011]) where new and used goods are found, and the review systems [Epinions.com 2011] where

the consumers opinions and reviews on specific products or services are expressed.

Figure 4(c) depicts how RepT supports trust management. The cloud service consumer x has a certain minimum trust threshold $\mathcal{T}_x$ and the cloud service provider y has a set of trusted relations $\mathcal{T}r(y) = \{r_1, r_2, ..., r_i\}$ (i.e., other cloud service consumers), which give trust feedbacks on the cloud service provider $\mathcal{T}f(y) = \{f_1, f_2, ..., f_n\}$. These feedbacks are used to calculate the reputation of y, denoted as $\mathcal{R}ep(y)$, as shown in Equation 3. The cloud service consumer x determines whether to proceed with the transaction based on the reputation result of y. The more positive feedbacks that y receives, the more likely x will trust the cloud service provider y.

$$\mathcal{R}ep(y) = \frac{\sum_{x=1}^{|\mathcal{T}f(y)|} \mathcal{T}f(x,y)}{|\mathcal{T}f(y)|} \tag{3}$$

$$\mathcal{T}r\,(x,\,y) = \begin{cases} 1 & if \quad \mathcal{R}ep(y) \geq \mathcal{T}_x \\ 0 & otherwise \end{cases} \tag{4}$$

Similarly, there exist several efforts that use RepT in trust management of cloud computing. Habib et al. [Habib et al. 2011] focus on aggregating the reputation of a particular cloud service based on feedback using QoS and other attributes (e.g., elasticity, geographical location). The approach is applicable for different cloud service models. In [Krautheim et al. 2010], a reputation-based trust model is proposed that focuses on Infrastructure as a Service (IaaS) cloud services. Noor and Sheng [Noor and Sheng 2011b; 2011a] propose a reputation-based trust management framework that distinguishes the credible feedbacks from the misleading ones.

4.1.4 *Prediction as a Trust Management Technique (PrdT).* Prediction as a trust management technique (PrdT) is very useful especially when there is no prior information regarding the cloud service's interactions (e.g., previous interactions, history records) [Skopik et al. 2009]. PrdT has been proposed in the cloud environment [Habib et al. 2011; Noor and Sheng 2011b; 2011a] and the service-oriented environment [Skopik et al. 2009; 2010]. The basic idea behind PrdT is that *similar minded entities* (e.g., cloud service consumers) are more likely to trust each other [Matsuo and Yamamoto 2009; Ziegler and Golbeck 2007].

Figure 4(d) depicts how PrdT works to support trust management. The cloud service consumer x has some capabilities and interests (denoted i_x) represented in a vector space model by binary data, $i_x = (i_1, i_2, ..., i_j)$, and a certain minimum trust threshold $\mathcal{T}_x$ are used to determine whether to trust the other cloud service consumers. Similarly, the cloud service consumer y also has some capabilities and interests (denoted as i_y) represented in a vector space model by binary data, $i_y = (i_1, i_2, ..., i_k)$, and a certain minimum trust threshold $\mathcal{T}_y$ is also used to determine whether trust the other cloud service consumers. The similarity between those two vectors (i.e., i_x and i_y) can be calculated using a similarity measurement such as the Cosine Similarity [Skopik et al. 2009], as shown in Equation 5. The more similar these capabilities and interests are, the more likely that the cloud service consumer x will trust y.

$$sim\,(i_x,\, i_y) = \frac{i_x \cdot i_y}{\|i_x\| \cdot \|i_x\|} \tag{5}$$

$$\mathcal{T}r\,(x,\, y) = \begin{cases} 1 & if \quad sim\,(i_x,\, i_y) \geq \mathcal{T}_x \Leftrightarrow sim\,(i_x,\, i_y) \geq \mathcal{T}_y \\ 0 & otherwise \end{cases} \tag{6}$$

Noor and Sheng [Noor and Sheng 2011b; 2011a] propose a similarity technique (i.e., distinguishing similar minded cloud service consumers) to determine credible feedbacks from the misleading ones. Habib et al. [Habib et al. 2011] uses PrdT to increase the quality of feedback where the trustworthiness of cloud service consumers is derived from the consensus of feedbacks (i.e., where feedbacks on a cloud service are similar to trust or distrust). PrdT can be used to refine the trust results and to increase the credibility of trust feedbacks.

5. AN ANALYTICAL FRAMEWORK FOR TRUST MANAGEMENT

In this section, we propose a generic analytical framework for trust management in cloud environments (see Figure 5). In the framework, interactions in cloud applications occur at three layers. For each layer, a set of dimensions is identified that will be used as a benchmark to evaluate and analyze existing trust management research prototypes in Section 6.

5.1 Layers of the Trust Management Analytical Framework

The three layers of the trust management framework include: the *trust feedback sharing* layer, the *trust assessment* layer, and the *trust result distribution* layer (Figure 5).

— *Trust Feedback Sharing Layer (TFSL).* TFSL consists of different parties including cloud service consumers and providers, which give trust feedbacks to each other. These feedbacks are maintained via a module called the Trust Feedback Collector. The feedbacks storage relies on the trust management systems, in the form of centralized, decentralized or even in the cloud environment through a trusted cloud service provider.

— *Trust Assessment Layer (TAL).* This layer represents the *core* of any trust management system: trust assessment. The assessment might contain more than one metrics. TAL handles a huge amount of trust assessment queries from several parties through a module called the Trust Result Distributor. This typically involves checking the trust results database and performing the assessment based on different trust management techniques (more details on trust management techniques can be found in Section 4.1). TAL delivers the trust results to a database in the trust results distribution layer through the module of the trust result distributor. This procedure is taken to avoid redundancy issues in trust assessment.

— *Trust Result Distribution Layer (TRDL).* Similar to TFSL, this layer consists of different parties including cloud service consumers and providers, which issue trust assessment inquiries about other parties (e.g., a cloud service consumer

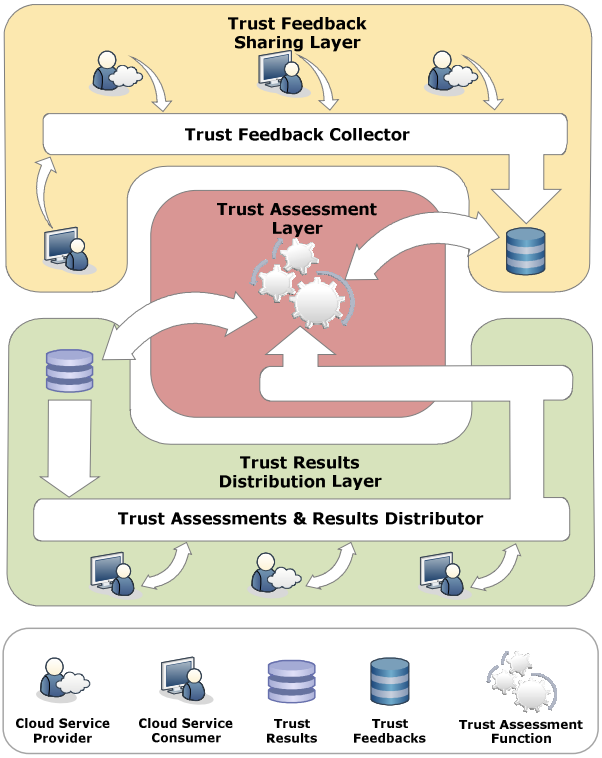


Fig. 5. Architecture of the Trust Management Analytical Framework

inquires about a specific cloud service). All trust assessment inquiries are transmitted to the trust assessment function through the module of trust assessment and results distributor. The final results are maintained in a database where cloud service consumers and providers can retrieve.

5.2 Dimensions for Evaluating Trust Management Frameworks

We identify a set of dimensions to study trust management issues where each layer of the framework has several dimensions. These dimensions are identified by considering the highly dynamic, distributed, and non-transparent nature of cloud environments.

5.2.1 *The Trust Feedbacks Sharing Layer.* There are four dimensions in this layer:

—*Credibility.* Credibility refers to the quality of the information or service that makes cloud service consumers or providers to trust the information or service. The credibility evaluation appears in several forms including the entity's credibility (e.g., a cloud service credibility) and the feedback credibility (more details are explained in Section 4.1.1). Since there is a strong relation between credibility and identification as emphasized in [David and Jaquet 2009], the parallel data

(i.e., feedback) processing require a proper identity scheme [Wei et al. 2009] for cloud service consumers and providers. For example, if no proper identity scheme is deployed, the trust management system can easily suffer from attacks such as *Sybil attacks* [Friedman et al. 2007], which leads to low accuracy in trust results.

—*Privacy.* This dimension refers to the degree of sensitive information disclosure that the cloud service consumers might face during the interactions with the trust management system. There are several cases of privacy breaches that may occur such as leaks of the cloud service consumers' sensitive information (e.g., user names, passwords, date of birth, address) or behavioral information (e.g., with whom the cloud service consumer interacted, the kind of cloud services the consumer showed interest, etc.). Indeed, cryptographic encryption techniques will decrease the data utilization [Ren et al. 2012] and traditional anonymization techniques (e.g., de-identification by removing personal identification information[Fung et al. 2010]) are inadequate in cloud environments [Roy et al. 2010] due to its highly dynamic and distributed nature.

—*Personalization.* Personalization refers to the degree of autonomy that the cloud service consumers and providers adhere to the trust management rules. Both can have proper personalization in their feedback designs and executions. This means that cloud service consumers and providers can select the feedback process (e.g., automated or manually driven) and the techniques they prefer. Personalization is applicable if the trust management system has fully autonomous collaboration, where each participant needs to interact via well-defined interfaces that allow participants to have control over their feedback and the flexibility to change their feedback processes without affecting each other. It is difficult to have a fully autonomous collaboration because of the complex translation features it requires [Medjahed et al. 2003].

—*Integration.* Integration refers to the ability to integrate different trust management perspectives and techniques. Participants can give their feedback from different perspectives (i.e., the cloud service provider and the cloud service consumer) through different trust management techniques (i.e., reputation, policy, etc.). Combining several trust management techniques can generally increase the accuracy of the trust results.

5.2.2 *The Trust Assessment Layer.* There are six dimensions in this layer:

—*Perspective.* Some trust management approaches focus on the cloud service provider's perspective while others focus on the cloud service consumer's perspective. It is therefore crucial to determine the perspective supported by a trust assessment function. The more perspectives the trust management system support, the more comprehensive the trust management system becomes.

—*Technique.* This dimension refers to the degree a technique can be adopted by the trust management system to manage and assess trust feedbacks. It is important to differentiate between the trust assessment functions that adopt a certain technique for trust management from the ones that adopt *several* trust management techniques together. Adopting several trust management techniques together can increase the accuracy of the trust results.

—*Adaptability.* Adaptability refers to how quickly the trust assessment function can adapt to changes of the inquisitive parties (i.e., cloud service providers or cloud service consumers). Some trust assessment inquiries can follow certain customized criteria from the inquisitive parties (e.g., weighing the feedback based on the size of the transaction), while others may follow the general trust assessment metric. In addition, updating feedbacks and trust results may be used as another indicator of adaptability because of the highly dynamic nature of cloud environments where new cloud service providers and consumers can join while others might leave at any time.

—*Security.* This dimension refers to the degree of robustness of the trust assessment function against malicious behaviors and attacks. There are two different security levels where attacks can occur: the *assessment function security level* and the *communication security level*. In the assessment function security level, there are several potential attacks against the trust assessment function including *whitewashing* [Lai et al. 2003], *self-promoting* [Douceur 2002], and *slandering* [Ba and Pavlou 2002]. Self-promoting and slandering attacks can either occur in a *Non-collusive Malicious Behavior* (e.g., an attacker gives numerous misleading feedbacks in a short period of time to increase or decrease the trust results of a cloud service) or *Collusive Malicious Behavior* (e.g., several attackers collaborate to give numerous misleading feedbacks). At the communication security level, there are several attacks such as *Man-in-the-Middle* (MITM) attack [Aziz and Hamilton 2009] and *Denial-of-Service* (DoS) attack or distributed *Denial-of-Service* (DDoS) attack [Hussain et al. 2003].

—*Scalability.* Given the highly dynamic and distributed nature of cloud environments, it is important that the trust management system be scalable. The scalability dimension refers to the ability of the trust management system to grow in one or more aspects (e.g., the volume of accessible trust results, the number of trust assessment inquiries that can be handled in a given period of time, and the number of trust relationships that can be supported). Trust models that follow a centralized architecture are more prone to several problems including scalability, availability and security (e.g., Denial-of-Service (DoS) attack) [Hoffman et al. 2009].

—*Applicability.* This dimension refers to the degree that the trust assessment function can be adopted to support trust management systems deployed for cloud services. It is important to differentiate the type of cloud services where the trust assessment functions are suitable. The more types of cloud services the trust assessment function can support, the more comprehensive the trust assessment function is.

5.2.3 *The Trust Results Distribution Layer.* There are four dimensions in this layer:

—*Response Time.* This is the time that the trust management system requires to handle trust assessment inquiries, to access feedbacks and to distribute trust results, especially when there is a significant number of trust relationships that are supported. If the trust management system needs a long response time, the number of inquiries that the trust management system will be able to handle will

be low.

—*Redundancy.* This dimension refers to the degree of redundancy support that the trust management system maintains in order to manage and assess the trust feedbacks. There are two redundancy approaches: i) the *assessment redundancy* (i.e., the unnecessary process of duplication that the trust assessment function performs) which occur when multiple trust assessment inquiries are issued sequentially for the same cloud service, and ii) the *trust data redundancy* (i.e., the replication of the trust data including feedbacks and trust results) used to avoid scalability and monitoring issues. Redundancy causes resource waste and eventually affects the performance of the trust management system.

—*Accuracy.* Accuracy refers to the degree of correctness of the distributed trust results that can be determined through one or more accuracy characteristics such as the unique identification of feedbacks and using the proper assessment function security level. Poor identification of feedbacks can lead to inaccurate trust results while the lack of proper assessment security function makes the trust management system penetrable and the distributed trust results are more likely to be manipulated by attackers.

—*Security.* The security dimension refers to the degree of protection that the trust assessments and results distributor have against malicious behaviors and attacks. The *access control level* determines whether the trust management system uses any access control technique for the trust results distribution while security at the *communication level* is similar to that in the trust assessment layer. Ultimately, if the trust assessments and results distributor have higher protection against security threats, the trust management system becomes more reliable.

6. RESEARCH PROTOTYPES

In this section, we present an overview of a set of representative research prototypes on trust management. These research prototypes are then analyzed and compared using the assessment dimensions identified in Section 5.2.

6.1 Overview of Major Research Prototypes

We present an overview of several representative trust management research prototypes on cloud computing and the most relevant areas such as the grid, Peer-to-Peer (P2P), and service-oriented computing.

—**Security-Aware Cloud Architecture:** In [Hwang et al. 2009; Hwang and Li 2010], Hwang et al. propose a security-aware cloud architecture that uses Virtual Private Network (VPN) and Secure Socket Layer (SSL) for secure communication. The research focuses on different trust management perspectives such as the cloud service provider's and consumer's perspectives. From the service provider's perspective, the proposed architecture uses the trust negotiation and the data coloring (integration) approach based on the fuzzy logic technique and the Public-Key Infrastructure (PKI) for cloud service consumer authentication. From the service consumer's perspective, the proposed architecture uses the Distributed-Hash-Table (DHT)-based trust-overlay networks among several data centers to deploy a reputation-based trust management technique. Although it is

mentioned that the architecture is reputation-based, it is actually based on predefined policies that evaluate the credibility of cloud services. In the other words, the security aware cloud architecture is a policy-based trust management system because reputation is actually based on other trusted participants opinions (i.e., cloud service consumers feedbacks) on a specific cloud service (as described in Section 4.1).

—**Compliant Cloud Computing Architecture (C3):** Brandic et al. [Brandic et al. 2010] propose a novel approach for compliance management in cloud environments to establish trust among different parties. The architecture focuses on cloud service consumer's perspective to protect cloud resources and preserve the privacy for all parties. This architecture is centralized and uses a certification mechanism for authentication, compliance management to help the cloud service consumers have proper choices in selecting cloud services. However, the architecture does not make use of other trust techniques such as *reputation*, *recommendation*, etc. which represent the participants' opinions. The authors further propose a novel language for specifying compliance requirements based on a model-driven technique using Unified Modeling Language (UML) for security, privacy and trust. The C3 middleware is responsible for the deployment of certifiable and auditable applications. This approach is considered to be a policy-based trust management system in the sense that it depends on policy compliance to enhance privacy, security and establish trust among cloud service providers and consumers.

—**TrustCloud: A Framework for Accountability and Trust in Cloud Computing:** Ko et al. [Ko et al. 2011] propose the TrustCloud framework for accountability and trust in cloud computing. The framework focuses on cloud service consumer's perspective to enforce cloud accountability and auditability. The framework exploits a centralized architecture, detective controls, and monitoring techniques for achieving trusted cloud services. In particular, TrustCloud consists of five layers, including *workflow*, *data*, *system*, *policies*, and *laws and regulations*, to address accountability in cloud environments. All these layers maintain the cloud accountability life cycle that consists of seven phases including *policy planning*, *sense and trace*, *logging*, *safe-keeping of logs*, *reporting and replaying*, *auditing*, and *optimizing and rectifying*.

—**Multi-faceted Trust Management System Architecture for Cloud Computing:** Habib et al. [Habib et al. 2011] propose a multi-faceted Trust Management (TM) system for cloud computing to help consumers identify trustworthy cloud service providers. The system focuses on the service consumer's perspective to establish trust relations between cloud service providers and consumers. It uses a centralized approach to collect trust-relevant information from multiple sources. In particular, the architecture models uncertainty of trust information using a set of Quality of Service (QoS) attributes such as security, latency, availability, and customer support. Finally, the architecture combines two different trust management techniques, namely reputation and recommendation.

—**CLOUD-ARMOR: A Trust Management Framework for Services in Cloud Environments:** Noor and Sheng [Noor and Sheng 2011b; 2011a] propose a trust management framework to deliver Trust as a Service (TaaS). The

framework focuses on the cloud consumer's perspective to establish trust relations between cloud service providers and consumers. CLOUD-ARMOR relies on a decentralized architecture for trust management. It supports different models including a *credibility* model that distinguishes the credible feedbacks from the misleading ones and detects malicious feedbacks from attackers and a *replication determination* model that dynamically decides the optimal replica number of the trust management service so that the trust management service can be always maintained at a desired availability level.

—**Dynamic Policy Management Framework (DPMF):** Yu and Ng [Yu and Ng 2006; 2009] develop a dynamic policy management framework that allows authorization decisions for resource sharing among multiple virtual organizations to take place without requiring complete policy information. The framework focuses on the perspectives of both service consumers and providers to protect virtual organizations resources and to preserve privacy for all trust entities. Similar to CLOUD-ARMOR, this framework has a decentralized architecture. The framework uses a *Conflict Analysis with Partial Information* (CAPI) mechanism to deploy a policy-based trust management system that measures similarities among policies to minimize policy disclosures.

—**Sabotage-Tolerance and Trust Management in Desktop Grid Computing:** In [Domingues et al. 2007], Domingues et al. propose an approach for sabotage detection and a protocol for trust management that focuses on the service provider's perspective to protect grid resources and preserve privacy. This protocol has a centralized architecture that uses trust management based on a referral relationship technique (i.e., recommendation) for access control. Domingues et al. propose a *Volunteer Invitation-based System* (VIS) to deploy a recommendation-based trust management system that relies on the notion of responsibility clustering where each volunteer invitation holder has ultimate responsibility for referral relationships. These kinds of relationships are represented in a *trust tree* through multiple referral relationships where each level of the tree is responsible for the lower level's behavior.

—**Grid Secure Electronic Transaction (gSET):** Weishaupl et al. [Weishaupl et al. 2006] develop a dynamic trust management framework for virtual organizations to minimize the credentials disclosure between different parties. The framework focuses on both the service provider's and the service requester's perspectives to protect virtual organizations' resources and privacy. This framework has a centralized architecture that uses PKI for authentication and trust management for access control. The authors adapt the *Secure Electronic Transaction* (SET) concept which is originally developed by MasterCard, Visa and others to suit the grid environment. The deployed framework is a policy-based trust management system that depends on PKI to enhance privacy, security and establish trust between service providers and requesters.

—**Role-Based Trust Chains:** In [Chen et al. 2008], Chen et al. present a heuristic-weighting approach to discover a specific set of credentials which is referred to as *credential chains* that satisfies several roles at control authorization levels. Instead of disclosing and authenticating credentials for each state of services such as state machines [Thomas and Hun 2002], the heuristic edge

weighting approach allows the peer to choose the most likely path in credentials (i.e., credential chains) to minimize credential disclosures and establish role-based trust between peers in P2P networks. This approach has a decentralized architecture that uses a private key for authentication and credential chaining for role-based trust delegation. Credentials are signed by private keys to avoid their forgery. As a result, the deployed approach is considered as a policy-based trust management system that allows the service requesters to choose the most likely chain of credentials to establish trust delegation to access the resources that they select.

—**Bootstrapping and Prediction of Trust:** In [Skopik et al. 2009], Skopik et al. propose a bootstrapping and prediction approach for trust management in large-scale systems. The proposed techniques work when there is no prior information regarding a certain entity (e.g., no previous interactions, no history records, no external influence such as reputation, recommendations). The approach follows a centralized architecture and focuses on the service requester's perspective, helping them to choose the appropriate service. Skopik, et al. introduce the concepts of *mirroring* and *teleportation* of trust to deploy a trust management system that combines several trust management techniques such as *prediction* and *recommendation*. Both concepts depend on similarities among measures of interests and capabilities to establish trust between service requesters and providers. Although Skopik et al. claim that there is no prior information required regarding a certain entity, both concepts (i.e., mirroring and teleportation of trust) depend on previous, well-established and trustworthy relationships in order to measure the similarities in interests or capabilities. In the other words, it still presents a transitive trust flavor, representing an informal recommendation.

—**A Negotiation Scheme for Access Rights Establishment:** Koshutanski and Massacci [Koshutanski and Massacci 2007] present a negotiation scheme that allows access rights establishment based on prior knowledge about the kind of credentials and privacy requirements that are needed to take the appropriate access decisions. The scheme focuses on the service provider's perspective, has a centralized architecture, and uses certificates for authentication. Koshutanski and Massacci develop a negotiation mechanism to deploy a policy-based trust management system that gives all parties prior notification about credentials and privacy requirements to minimize the credentials disclosure among parties. The framework does not have any particular mechanism or assumptions for secure communications.

—**A Trust Management Framework for Service-Oriented Environments (TMS):** Conner, et al. [Conner et al. 2009] propose a trust management framework for service-oriented architecture (SOA), which focuses on the service provider's perspective to protect resources from unauthorized access. This framework has a decentralized architecture that uses trust management for access control and it assumes secure communication. However, the framework does not have any particular mechanism for uniquely authenticating service requesters, which eventually leads to poor identification of trust feedbacks. The framework offers multiple trust evaluation metrics to allow trust participants to have their own customized evaluation. To reduce communication overheads, Conner et al. introduce a trust

evaluation caching mechanism. This mechanism represents a good example for *assessment redundancy* (as described in Section 5.2.3) where the trust assessment function evaluates feedbacks only when necessary. The framework relies on a customized evaluation mechanism to deploy a reputation-based trust management system that allows service providers to assess their clients (i.e., service requesters) to establish trust between service providers and requesters. Although the framework allows customized trust evaluation, service providers need to develop their own reputation scoring functions.

—**Reputation Assessment for Trust Establishment Among Web Services (RATEWeb):** Malik and Bouguettaya [Malik and Bouguettaya 2009c; 2009b; 2009a] propose reputation assessment techniques based on QoS parameters. The techniques focus on the service requesters' perspective and the proposed system has a decentralized architecture where each service requester records her own perceptions of the reputation of a service provider. The proposed framework supports different models for feedback sharing including the *publish-subscribe collection* model, the *community broadcast collection* model, and the *credibility-based collection* model. Malik and Bouguettaya present several assessment metrics (e.g., rater credibility, majority rating, and temporal sensitivity), which enable the trust management system to combine several trust management techniques, such as *policy* and *reputation*, to improve the accuracy of trust results.

6.2 Evaluation of Trust Management Research Prototypes

The evaluation of trust management prototypes covers 30 representative research prototypes where 69% of these research prototypes have been published in the last 6 years and the rest represents some classical research prototypes that we cannot resist taking notice of them, due to their fundamental contribution and influence in the field of trust management. As shown in Figure 6, the evaluation is organized to assess research prototypes using three different layers (i.e., the *trust feedback sharing* layer, the *trust assessment* layer and the *trust result distribution* layer) based on a set of dimensions, proposed in Section 5.

6.2.1 *The Trust Feedback Sharing Layer (TFSL).* Figure 7 (a) shows some statistical information of research prototypes on the TFSL layer. For the credibility dimension, we note that the majority of research prototypes (63.5%) do not use any mechanisms to identify credible feedbacks in their trust models. For the privacy dimension, 50% of research prototypes do not have any particular mechanism for preserving the privacy of parties; 47% of research prototypes only focus on the service requesters' privacy and the rest 3% focus on the privacy of both (i.e., service requesters and service providers). For the personalization dimension, a high proportion of research prototypes (73%) does not consider the personalization aspect in their trust models and the rest research prototypes only use partial personalization in their trust models. Finally, for the integration dimension, the majority of research prototypes (73%) do not make strong use of feedbacks combination.

6.2.2 *Trust Assessment Layer (TAL).* Figure 7 (b) depicts statistical information of research prototypes on the TAL layer. For the perspective dimension, we note that there is a fair degree of variety in the listed research prototypes. More

Prototypes	TFSL				TAL						TRDL			
	Credibility	Privacy	Personalization	Integration	Perspective	Technique	Adaptability	Security	Scalability	Applicability	Response Time	Redundancy	Accuracy	Security
Ko et al.11	EC	SR	N	NFC	SRP	PocT$^{\mu}$	N	AFL\CL	C	IaaS	NAT	N	F	ACL\CL
Habib, et al. 11	EC	N	P	SFC	SRP	RecT\RepT\PrdT	P	AFL\CL	C	All	NAT	N	F	ACL\CL
Noor and Sheng 11	FC\EC	SR	P	NFC	SRP	RepT\PrdT	F	AFL\CL	D	All	NAT	TR	F	ACL\CL
Krautheim, et al. 10	EC	SR	N	SFC	SRP\SPP	RecT\RepT	N	CL	C	IaaS	NAT	N	P	ACL\CL
Brandic et al. 10	EC	SR	P	NFC	SRP	PocT$^{\mu}$	P	CL	C	IaaS\PaaS	NAT	N	P	ACL\CL
Yao, et al. 10	EC	N	N	NFC	SRP	PocT$^{\mu}$	P	CL	C	IaaS	SAT	N	P	ACL\CL
Hwang, et al. 09	EC	SR	N	NFC	SRP	PocT$^{\Omega}$	N	AFL\CL	C	All	SAT	N	F	ACL\CL
Santos, et al. 09	EC	SR	N	NFC	SRP	PocT	N	CL	D	IaaS	NAT	TR	P	ACL\CL
Manuel, et al. 09	FC\EC	SR	N	SFC	SRP	PocT\RepT	N	AFL\CL	C	All	SAT	N	F	ACL\CL
Alhamad, et al. 10	EC	SR	P	NFC	SRP	PocT$^{\mu}$\RepT	N	N	D	IaaS	SAT	N	P	N
Azzedin and Maheswaran 02	FC\EC	N	N	SFC	SRP	RepT	N	AFL	D	IaaS	SAT	AR\TR	P	ACL
Ching, et al. 04	FC\EC	N	N	SFC	SRP\SPP	PocT\RepT	N	AFL\CL	D	IaaS	SAT	AR\TR	F	ACL\CL
Yu, et al. 06	N	SR\SP	N	NFC	SRP\SPP	PocT	N	AFL	D	IaaS	NAT	AR\TR	P	ACL
Domingues, et al. 06	EC$^{\beta}$	N	N	NFC	SPP	RecT	N	N	C	All	NAT	N	N	ACL
Song, et al. 05a	EC	N	N	NFC	SRP	PocT$^{\Omega}$	F	AFL\CL	C	IaaS	SAT	N	F	ACL\CL
Song, et al 05b	EC	N	N	NFC	SRP\SPP	PocT$^{\Omega}$	F	AFL\CL	C	All	SAT	N	F	ACL\CL
Weishaupl, et al. 06	N	SR	N	NFC	SRP\SPP	PocT	N	CL	C	IaaS	NAT	N	P	ACL\CL
Chen, et al. 08	N	SR	P	NFC	SRP	PocT	N	AFL\CL	D	All	SAT	AR\TR	F	ACL\CL
Srivatsa, et al. 06	FC\EC	N	N	NFC	SRP\SPP	RepT	F	AFL	D	All	SAT	TR	F	ACL
Aringhieri, et al. 06	FC\EC	SR	N	NFC	SRP\SPP	RepT	P	AFL\CL	D	All	SAT	TR	F	ACL\CL
Zhou, et al. 07	FC\EC	N	N	NFC	SRP\SPP	RepT	P	AFL\CL	D	All	SAT	TR	F	ACL\CL
Kamvar, et al. 03	FC\EC	N	N	NFC	SRP\SPP	RepT	P	AFL	D	All	NAT	TR	N	ACL
Xiong, et al. 04	FC\EC	N	N	NFC	SRP	RepT	F	AFL	D	All	NAT	AR\TR	P	ACL
Skopik, et al. 09	N	N	N	SFC	SRP	RecT\PrdT	N	N	C	All	NAT	N	F	N
Skopik, et al. 10	N	N	N	SFC	SRP	PocT\PrdT	N	CL	C	IaaS	SAT	TR	P	ACL\CL
Koshutanski, et al. 07	N	SR	N	NFC	SPP	PocT	N	AFL	C	IaaS	NAT	N	F	ACL
Park, et al. 05	FC\EC	N	N	SFC	SRP	RecT\RepT	N	AFL\CL	D	All	SAT	TR	F	ACL\CL
Skogsrud, et al. 04	N	SR	P	NFC	SPP	PocT	F	AFL\CL	D	IaaS	SAT	TR	F	ACL\CL
Conner, et al. 09	N	N	P	NFC	SPP	RepT	P	AFL	D	All	SAT	AR\TR	N	ACL
Malik, et al. 09	FC\EC	SR	P	NFC	SRP	PocT\RepT	F	AFL	D	All	SAT	TR	F	ACL\CL

Notes:

$^{\Omega}$Although it is mentioned that the trust management system is based on *Reputation,* it is actually based on pre-defined policies that measure the credibility of the trust related parties (i.e., entities).

$^{\beta}$Entity's credibility is identified through referral relationships.

$^{\mu}$Service Level Agreement (SLA) is used to perform a *policy-based* trust management.

Trust Feedbacks Sharing Layer (TFSL)

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

Trust Assessment Layer (TAL)

Perspective		Technique		Adaptability		Security		Scalability		Applicability	
SPP	Service Provider Perspective	PocT	Policy Technique	F	Full	AFL	Support Assessment Function level	C	Centralized	IaaS	Infrastructure as a Service
SRP	Service Requester Perspective	RecT	Recommendation Technique	P	Partial	CL	Support Communication level	D	Decentralized	PaaS	Platform as a Service
		RepT	Reputation Technique	N	None					SaaS	Software as a Service
		PrdT	Prediction Technique			N	None			All	All three models

Trust Results Distribution Layer (TRDL)

Response Time		Redundancy		Accuracy		Security	
SAT	Strong Emphasis of Assessment Time	AR	Support Assessment Redundancy	F	Full	ACL	Support Access Control level
NAT	No Strong Emphasis of Assessment Time	TR	Support Trust Data Redundancy	P	Partial	CL	Support Communication level
		N	None	N	None	N	None

Fig. 6. Evaluation of Trust Management Research Prototypes

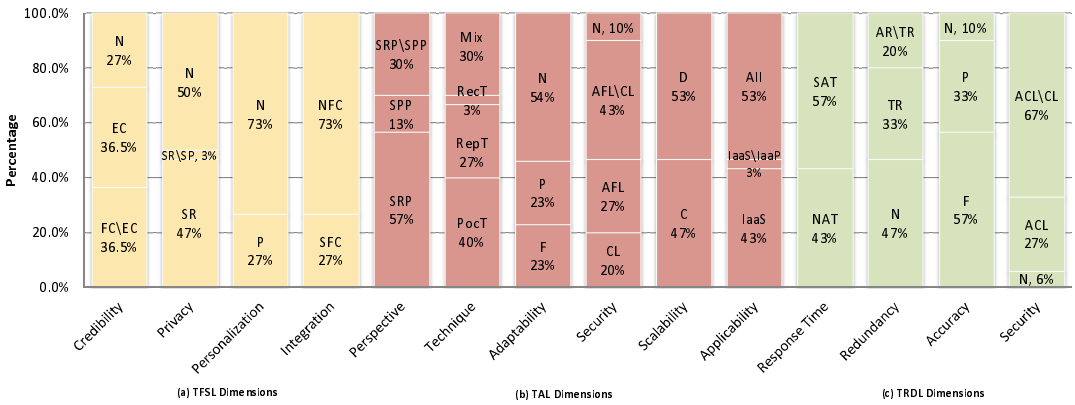


Fig. 7. Evaluation of Trust Management Research Prototypes Among All Dimensions

than half of the research prototypes (57%) focus on the service requester's perspective (SRP); 13% of the research prototypes focus on the service provider's perspective (SPP); and the rest 30% focus on both (i.e., SSP and SRP). For the technique dimension, 40% of research prototypes use policy as a trust management technique (PocT); 27% of research prototypes use reputation as a trust management technique (RepT); 30% of research prototypes use a combination of different trust management techniques (i.e., policy, recommendation, reputation, or prediction). Interestingly, only 3% of research prototypes use recommendation as a trust management technique (RecT).

For the adaptability dimension, more than half of the representative research prototypes (54%) do not support adaptability to changes that trusted parties require. 23% of research prototypes support partial adaptability in their trust models and the remaining research prototypes (23%) support full adaptability in their trust models. For the security dimension, 10% of research prototypes do not use any security mechanisms; 20% of research prototypes support secure communication; 27% of research prototypes support the assessment function level security (AFL) and the rest (43%) of research prototypes support both secure communication and AFL. For the scalability dimension, 53% of research prototypes have a decentralized architecture for their trust management system. Finally, for the applicability dimension, 53% of research prototypes can be adapted to support trust management system deployed for all types of cloud services (i.e., IaaS, PaaS, SaaS); 43% of research prototypes use approaches suitable for IaaS cloud services. Only 3% of research prototypes use approaches suitable for both models of IaaS and PaaS.

6.2.3 *Trust Result Distribution Layer (TRDL).* Figure 7 (c) shows the statistical information of the prototypes on the TRDL. For the response time dimension, we note that the majority of research prototypes (57%) have a strong emphasis on the assessment time. For the redundancy dimension, 47% of the research prototypes do not focus on redundancy techniques at all. 33% of the research prototypes support trust results redundancy (TR) and the remaining prototypes (20%) sup-

port both, i.e., TR and the trust assessment redundancy (AR). For the accuracy dimension, more than half of the representative research prototypes (57%) are accurate in meeting the inquisitive parties expectations. 33% of research prototypes have partial accuracy and 10% have no accuracy in meeting the inquisitive parties expectations. Finally, for the security dimension, 6% of research prototypes do not use any security mechanisms to mitigate potential attacks that target trust results. 27% of research prototypes support the Access Control Level (ACL) security and the remaining prototypes (67%) support the both (i.e, secure communication and ACL).

7. CLOUD SERVICE PROVIDERS

Major software vendors such as IBM, Microsoft, Amazon are offering different cloud services. The purpose of this section is to analyze these cloud services from the aspect of trust. It should be noted that there is a large number of cloud providers and we will not be able to cover all of them. Instead, we focus on some major players in this arena. In this section, we first discuss a set of trust characteristics for cloud services and then compare several major cloud providers.

7.1 Trust Characteristics in Cloud Services

Many researchers use a qualitative approach to compare existing cloud services for all three different service models (i.e., IaaS, PaaS and SaaS) among several cloud service providers from different perspectives such as the security features [Hwang and Li 2010; Hwang et al. 2009], virtual infrastructure management capabilities [Sotomayor et al. 2009], and services functionalities [Buyya et al. 2008]. On the other hand, others use a quantitative approach to compare the use of cloud services among several cloud service providers (i.e., in terms of the number of cloud service consumers). For example, Guy Rosen has conducted a survey of the market use of the cloud computing [Rosen 2011]. The survey compares the number of publicly accessible websites hosted on several cloud services (about 500,000 sites). According to the survey [Rosen 2011], the number of sites (i.e., cloud service consumers) reached 3,278 in August 2009 and this figure dramatically increased to nearly 9,000 in January 2011. Intuitively, this is an indicator that the cloud environment is becoming increasingly attractive.

In the following, we define a set of trust characteristics, including *authentication*, *security*, *privacy responsibility*, *virtualization* and *cloud service consumer accessibility*, which will be used to compare several major cloud service providers:

—*Authentication.* This characteristic refers to the techniques and mechanisms that are used for authentication in a particular cloud. Cloud consumers have to establish their identities every time they attempt to use a new cloud service by registering their credentials, which contain sensitive information. This can lead to privacy breaches if no proper identity scheme is applied for the cloud service consumers.

—*Security.* There are three security levels in a particular cloud: the *communication* security level (CSL), the *data* security level (DSL) and the *physical* security level (PSL). CSL refers to communication techniques such as Secure Socket Layer (SSL), etc. DSL refers to data replication techniques for data recovery. Finally,

PSL refers to physical security techniques such as hardware security.

—*Privacy Responsibility.* The privacy responsibility can be categorized into two different privacy responsibility categories: the *cloud service provider* privacy responsibility category and the *cloud service consumer* privacy responsibility category.

—*Virtualization.* This characteristic refers to techniques that are used for virtualization. There are two virtualization levels in a particular cloud: the *Operating System* (OS) level and the *application container* level. Virtualization techniques allow the cloud service provider to control and manage the underlying cloud environment, whereas the cloud service consumers have control on their virtual machines which include the storage, the process and even the selection of some network components for communication.

—*Cloud Consumer Accessibility.* This characteristic refers to techniques and mechanisms that are used for cloud service consumers to access cloud services such as Graphical User Interfaces (GUIs), Application Programming Interfaces (APIs), command-line tools, etc.

7.2 Comparison of Major Cloud Service Providers

We compare several representative cloud service providers including IBM, Microsoft, Google and Amazon and the result is shown in Table I. From the table we note that some of the cloud service providers (e.g., Amazon) focus on providing one cloud service model only while others (e.g., IBM and Microsoft) focus on providing all three service models (i.e., IaaS, PaaS and SaaS). It is worth mentioning that cloud service providers are targeting specific portions of cloud service consumers. For example, IBM is targeting only the service provider portion of the cloud service consumers. Consequently, most of the interactions are considered business-to-business interactions while other cloud service providers such as Microsoft, Google and Amazon are targeting both of the cloud service consumers portions (i.e., the service provider and service requesters). Thus, most of the interactions are business-to-business (B2B) and business-to-client (B2C).

Another interesting observation from Table I is that given the diverse number of available technologies, a cloud service consumer faces many configuration options when using cloud services. These options include the number of virtual machines, the type of virtual machines, time of tenancy, access control polices, etc. We argue that there is a need for intelligent techniques to make the cloud platform learn the patterns that cloud service consumers usually use to simplify the configuration process and make it more user-friendly. In addition, cloud service providers may deliver several cloud services that have similar features. It is very important for cloud service consumers to be able to choose a cloud service provider that provides *trustworthy* cloud services. The decision can be made on the basis of previous cloud service consumer's feedbacks where trust management is an effective approach to assess and establish trusted relationships.

8. TRUST MANAGEMENT OPEN ISSUES

For trust management of services in cloud environments, there is a need for efficient techniques to integrate all feedbacks from different parties such as cloud service

Table I. Comparison: Representative Cloud Service Providers Versus Service Trust Characteristics

Cloud Service Providers	IBM			Microsoft			Google		Amazon
Supported Service Models	IaaS	PaaS	SaaS	IaaS	PaaS	SaaS	PaaS	SaaS	IaaS
Service Types	Computation, Storage	Web apps	Web apps	Storage	Web apps	Web apps	Web apps	Web apps	Computation, Storage
Service Names	Ensembles	BlueCloud, Websphere CloudBurst Appliance, Research Compute Cloud (RC2)	Lotus Live	Microsoft Live Mesh	Windows Azure	.NET service, dynamic customer relationship management (CRM)	Google App Engine	Gmail, Google Docs	Elastic Compute Cloud (EC2), Simple Storage Service (S3), Simple Queue Service, SimpleDB
Authentication	Public-Key Infrastructure (PKI) and access management services			Rule-based access control, and password-based protection			Secure Shell (SSH) and Rule-based access control		Password-based protection or Secure Shell (SSH)*
Communication Security Level	WebSphere2 (Secure Socket Layer (SSL)) or Virtual Private Network (VPN)*			Secure Socket Layer (SSL) for data transfers			Secure Socket Layer (SSL) for data transfers		Secure Socket Layer (SSL) for data transfers
Data Security Level	Data de-duplication practices			Replicated data for data recovery			Grid-based redundancy		Elastic Block Store for failure recovery
Physical Security Level	Hardware security in data centers			Hardware security in data centers			Local and central monitoring techniques for Hardware security		Hardware security in data centers
Privacy	Cloud consumer's responsibility			Cloud Provider's responsibility			Cloud provider's responsibility		Cloud consumer's responsibility
Virtualization	Operating system level running on IBM PowerVM			Operating system level			Application container level		Operating system level running on a Xen hypervisor
Cloud Service Consumer Accessibility	Browser-based accessible GUI using Dojo Toolkit			Web-based Live Desktop			Web-based Administration		Amazon EC2 Command-line Tools or API*

Note:

* The cloud service consumer has the choice of provisioned technologies

providers and requesters. This still remains a significant challenge due to highly dynamic, distributed, and non-transparent nature of cloud services. Although current trust management techniques provide the foundation for establishing trust management of services in cloud environments, several research issues still need to be addressed. In particular, we identify the following directions for future research, namely *identification*, *privacy*, *personalization*, *integration*, *security*, and *scalability*.

—*Identification*: Since there is a strong relationship between credibility and identification as emphasized in [David and Jaquet 2009], it is crucial that trust management systems effectively identify cloud service consumers and providers in order to i) evaluate the credibility of entities (e.g., a cloud service's credibility) and trust feedbacks (more details are explained in Section 4.1.1) and ii) protect the integrity of the trust management system's parallel data (i.e., feedback) processing. However, based on the statistical information of the representative research prototypes in Section 6.2, we note that many of the research prototypes (63.5%) do not use any mechanisms to identify credible feedbacks in their trust models. In the cloud environment, credible feedbacks identification is becoming a significant challenge because of the overlapping interactions between the cloud service providers and consumers. The need to determine credible feedbacks will require appropriate strategies such as the one used in *SecureMR* [Wei et al. 2009] where a novel decentralized replication-based integrity verification scheme for running MapReduce is proposed.

—*Privacy*: Privacy is a fundamental concern in cloud computing. In particular, managing trust in cloud environments requires trust management systems to deal with the cloud service consumers' personal information. Cloud service consumers face several privacy threats such as i) leaking information pertaining to personal property (e.g., user names, passwords, date of birth, address) and ii) tracking consumers' behaviors (e.g., with whom they interacted, which cloud services they used, etc). According to the statistical information in Section 6.2, 50% of the research prototypes do not have any particular mechanism for preserving the privacy of participants. There is therefore a strong need for efficient techniques in preserving privacy of participants but with full consideration of the trust management system availability. One way to preserve privacy is to use cryptographic encryption techniques but there is no efficient way to process encrypted data [Pearson and Benameur 2010]. Another way is to adopt privacy techniques such as the ones used for *Airavat* [Roy et al. 2010] where a new approach integrating the mandatory access control and differential privacy is proposed for running MapReduce on Amazon's IaaS (EC2). The differential privacy technique could be used to ensure that the trust result of a cloud service does not violate the privacy of a cloud service consumer who gives the feedback. Fung et al. [Fung et al. 2010] overview several approaches for preserving privacy in data publishing and we believe that extensive work is needed for developing effective and efficient solutions for privacy protection in the cloud.

—*Personalization*: Cloud services provision several technologies for the same context (e.g., security) and the choice is up to the cloud service consumers (e.g., the use of SSL or VPN for IaaS, PaaS and SaaS such as in IBM, Password-based protection or Secure Shell (SSH) for IaaS such as in Amazon) regardless if the

cloud service consumer is a service provider or a service requester (i.e., these technologies are not suitable for all cloud service consumers). We therefore argue that there is a need for flexible techniques to help cloud service consumers in personalizing the provisioned technologies according to their specific needs. In addition, the number of technologies provisioned by cloud services might be large, which means that a cloud service consumer may face configuration difficulties when using cloud services (e.g., the number of virtual machines, the type of virtual machines, time of tenancy, and access control polices). As a result, there is a strong need for intelligent techniques to make the cloud platform learn the patterns that cloud service consumers usually use. In Section 6.2, we note that a high proportion of research prototypes (73%) does not consider the personalization aspect in their trust models and only 27% of research prototypes use partial personalization in their trust models. Consequently, trust personalization is becoming increasingly important. Trust management systems that support personalization should ensure that participants i) have the control over their trust feedbacks, ii) have their own personalized assessment criteria, iii) have the control over their trust results, and iv) have the flexibility to change their feedback processes.

—*Integration*: In the cloud environment, trusted parties can give their feedback from different perspectives (e.g., cloud service provider, cloud service consumer) using different techniques (e.g., reputation, policy, etc). Thus, it is important that trust management systems can make use of feedbacks by combining several techniques (e.g., the combination of the reputation technique and the recommendation technique can increase the accuracy of trust results). Combining trust management perspectives can lead to better trust results by matching appropriate service requesters to the trustworthy service providers. Unfortunately, we observe in Section 6.2 that the majority of the research prototypes (73%) do not make use of feedbacks integration. As a result, we believe that novel approaches that combine different trust management techniques and make use of feedbacks integration are needed to improve trust results.

—*Security*: Security is a critical issue for cloud computing to be adopted and must be enforced to give businesses the confidence that their data are safely handled. However, it is not unusual that a cloud service experiences malicious behaviors from its users. Due to the dynamic interactions and the distributed nature of cloud environments, it is difficult to know from whom the attack (e.g., whitewashing, self-promoting, and slandering attacks) is expected. Therefore, it is crucial that the trust management systems reliably identify malicious behaviors and mitigate such attacks. Similarly, from Section 6.2, we notice that 37% of research prototypes do not support or at least assume secure communication while 30% of research prototypes do not support the Assessment Function Level security (AFL) in the TAL Dimensions; 33% of research prototypes also do not support or assume secure communication in the TRDL Dimensions. Proper defense techniques are needed to reliably identify malicious behaviors and mitigates such attacks in cloud environments. Some recent proposals include the header analysis approach for Denial-of-Service (DoS) attacks detection proposed in [Hussain et al. 2003], the precise timing approach for identifying Man-in-the-Middle

(MITM) attacks proposed in [Aziz and Hamilton 2009], and the credibility-based trust evaluation approaches proposed in [Xiong and Liu 2004; Srivatsa and Liu 2006; Malik and Bouguettaya 2009a; Noor and Sheng 2011a]).

—*Scalability*: In cloud environments, the number of cloud services and their consumers is large and usually highly dynamic where new cloud services, as well as consumers, can join while others might leave the cloud environment at any time. This highly dynamic and distributed nature of cloud services requires that trust management systems be highly scalable in order to efficiently collect feedbacks and update trust results. According to the evaluation provided in Section 6.2, 47% of research prototypes rely on a centralized architecture for their trust management, which is not scalable and more prone to problems such as availability and security (e.g., Denial-of-Service (DoS) attack) [Hoffman et al. 2009]. Therefore, we believe that proper scalability and availability techniques are needed for trust management systems. Some recent work includes a decentralized approach proposed in [Noor and Sheng 2011a] where a replication model is proposed and in [Conner et al. 2009] where load balancing techniques are used to increase the availability of the trust management system.

9. CONCLUSION

In recent years, cloud computing has become a vibrant and rapidly expanding area of research and development. Trust is widely regarded as one of the top obstacles for the adoption and the growth of cloud computing. In this article, we have presented a comprehensive survey that is, to the best of our knowledge, the first to focus on the trust management of services in cloud environments. We distinguish the trust management perspectives and classify trust management techniques into four different categories. We further propose a generic analytical framework that can be used to compare different trust management research prototypes based on a set of assessment criteria. We overview and compare 30 representative research prototypes on trust management in cloud computing and the relevant research areas. 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.

ACKNOWLEDGMENTS

Talal H. Noor's work has been supported by King Abdullah's Postgraduate Scholarships, the Ministry of Higher Education: Kingdom of Saudi Arabia. Quan Z. Sheng's work has been supported by the Australian Research Council Discovery Grant DP0878917. We express our gratitude to the anonymous reviewers for their comments and suggestions which have greatly helped us to improve the content, quality, organization, and presentation of this work.

REFERENCES

Al-Sharawneh, J. and Williams, M. 2010. Credibility-based Social Network Recommendation: Follow the Leader. In *Proc. of the 21st Australasian Conf. on Information Systems (ACIS'10)*. Brisbane, Australia.

Alhamad, M., Dillon, T., and Chang, E. 2010. SLA-based Trust Model for Cloud Computing.

In *Proc. of the 13th Int. Conf. on Network-Based Information Systems (NBiS'10)*. Takayama, Gifu, Japan.

Amazon. 2011. Amazon.com: Online Shopping for Electronics, Apparel, Computers, Books, DVDs & More. Accessed 01/3/2011, Available at: http://www.amazon.com/.

Amazon-EC2. 2011. Elastic Compute Cloud (Amazon EC2). Accessed 01/04/2011, Available at: http://aws.amazon.com/ec2.

Amazon-S3. 2011. Amazon Simple Storage Service (Amazon - S3). Accessed 29/03/2011, Available at: http://aws.amazon.com/s3.

Aringhieri, R., Damiani, E., De Capitani di Vimercati, S., Paraboschi, S., and Samarati, P. 2006. Fuzzy Techniques for Trust and Reputation Management in Anonymous Peer-to-Peer Systems. *Journal of the American Society for Information Science and Technology 57,* 4, 528–537.

Aringhieri, R., Damiani, E., De Capitani di Vimercati, S., and Samarati, P. 2005. Assessing Efficiency of Trust Management in Peer-to-Peer Systems. In *Proc. of IEEE 14th Int. Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprise (WETICE'05)*. Linkoping, Sweden.

Armbrust, M., Fox, A., Griffith, R., Joseph, A., Katz, R., Konwinski, A., Lee, G., Patterson, D., Rabkin, A., Stoica, I., and Zaharia, M. 2010. A View of Cloud Computing. *Communications of the ACM 53,* 4, 50–58.

Artz, D. and Gil, Y. 2007. A Survey of Trust in Computer Science and the Semantic Web. *Web Semantics: Science, Services and Agents on the World Wide Web 5,* 2, 58–71.

Aziz, B. and Hamilton, G. 2009. Detecting Man-in-the-Middle Attacks by Precise Timing. In *Proc. of the 3rd Int. Conf. on Emerging Security Information, Systems and Technologies (SECURWARE'09)*. Athens/Glyfada, Greece.

Azzedin, F. and Maheswaran, M. 2002a. Integrating Trust Into Grid Resource Management Systems. In *Proc. of the Int. Conf. on Parallel Processing (ICPP'02)*. Vancouver, BC, Canada.

Azzedin, F. and Maheswaran, M. 2002b. Towards Trust-aware Resource Management in Grid Computing Systems. In *Proc. of the 2nd IEEE/ACM Int. Symp. on Cluster Computing and the Grid (CCGrid'02)*. Berlin, Germany.

Azzedin, F. and Maheswaran, M. 2004. A Trust Brokering System and its Application to Resource Management in Public-resource Grids. In *Proc. of the 18th Int. Parallel and Distributed Processing Symp. (IPDPS'04)*. Santa Fe, New Mexico.

Ba, S. and Pavlou, P. 2002. Evidence of the Effect of Trust Building Technology in Electronic Markets: Price Premiums and Buyer Behavior. *MIS Quarterly 26,* 3, 243–268.

Bertino, E., Ferrari, E., and Squicciarini, A. 2004. Trust Negotiations: Concepts, Systems, and Languages. *Computing in Science and Engineering 6,* 4, 27–34.

Bertino, E., Paci, F., Ferrini, R., and Shang, N. 2009. Privacy-preserving Digital Identity Management for Cloud Computing. *IEEE Data Eng. Bull 32,* 1, 21–27.

Blaze, M., Feigenbaum, J., Ioannidis, J., and Keromytis, A. D. 1999. *Secure Internet Programming.* Springer-Verlag, London, UK, Chapter The Role of Trust Management in Distributed Systems Security, 185–210.

Blaze, M., Feigenbaum, J., and Keromytis, A. 1998. KeyNote: Trust Management for Public-key Infrastructures. In *Proc. of the 6th Int. Workshop on Security Protocols*. Cambridge, UK.

Blaze, M., Feigenbaum, J., and Lacy, J. 1996. Decentralized Trust Management. In *Proc. of IEEE 17th Symp. on Security and Privacy (SP'96)*. Oakland, CA, USA.

Blaze, M., Feigenbaum, J., and Strauss, M. 1998. Compliance Checking in the PolicyMaker Trust Management System. In *Proc. of the 2nd Int. Conf. on Financial Cryptography (FC'98)*. Anguilla, BWI.

Blaze, M., Ioannidis, J., and Keromytis, A. 2000. Trust Management and Network Layer Security Protocols. In *Proc. of the 7th Int. Workshop on Security Protocols*. London, UK.

Brandic, I., Dustdar, S., Anstett, T., Schumm, D., Leymann, F., and Konrad, R. 2010. Compliant Cloud Computing (C3): Architecture and Language Support for User-Driven Compliance Management in Clouds. In *Proc. of IEEE 3rd Int. Conf. on Cloud Computing (CLOUD'10)*. Miami, Florida, USA.

Buyya, R., Yeo, C., and Venugopal, S. 2008. Market-oriented Cloud Computing: Vision, Hype, and Reality for Delivering it Services as Computing Utilities. In *Proc. of IEEE 10th Int. Conf. on High Performance Computing and Communications (HPCC'08)*. Dalian, China.

Camenisch, J. and Van Herreweghen, E. 2002. Design and Implementation of the Idemix Anonymous Credential System. In *Proc. of the 9th ACM Conf. on Computer and Communications Security (CCS'02)*. Washingtion, DC, USA.

Cantor, S., Kemp, J., Philpott, R., and Maler, E. 2005. Assertions and Protocols for the OASIS Security Assertion Markup Language (saml) v2. 0. Accessed 7/3/2010, Available at: http://docs.oasis-open.org/security/saml/v2.0/saml-core-2.0-os.pdf.

Cavoukian, A. 2008. Privacy in the clouds. *Identity in the Information Society 1,* 1, 89–108.

Chen, K., Hwang, K., and Chen, G. 2008. Heuristic Discovery of Role-Based Trust Chains in Peer-to-Peer Networks. *IEEE Transactions on Parallel and Distributed Systems 20,* 1, 83–96.

Clark, K., Warnier, M., Brazier, F., and Quillinan, T. 2010. Secure Monitoring of Service Level Agreements. In *Proc. of the 5th Int. Conf. on Availability, Reliability, and Security (ARES'10)*. Krakow, Poland.

Conner, W., Iyengar, A., Mikalsen, T., Rouvellou, I., and Nahrstedt, K. 2009. A Trust Management Framework for Service-Oriented Environments. In *Proc. of the 18th Int. Conf. on World Wide Web (WWW'09)*. Madrid, Spain.

Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., and Polk, W. 2008. RFC 5280 - Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile. Accessed: 19/04/2010, Available at: http://tools.ietf.org/html/rfc5280.

Damiani, E., De Capitani di Vimercati, S., Paraboschi, S., and Samarati, P. 2003. Managing and sharing servents' reputations in p2p systems. *IEEE Transactions on Knowledge and Data Engineering 15,* 4, 840–854.

Damiani, E., De Capitani di Vimercati, S., Paraboschi, S., Samarati, P., and Violante, F. 2002. A Reputation-Based Approach for Choosing Reliable Resources in Peer-to-Peer Networks. In *Proc. of the 9th ACM Conf. on Computer and Communications Security (CCS'02)*. Washington, DC, USA.

David, O. and Jaquet, C. 2009. Trust and Identification in the Light of Virtual Persons. Accessed 10/3/2011, Available at: http://www.fidis.net/resources/deliverables/identity-of-identity/.

De Capitani di Vimercati, S., Foresti, S., Jajodia, S., Paraboschi, S., Psaila, G., and Samarati, P. 2012. Integrating Trust Management and Access Control in Data-Intensive Web Applications. *ACM Transactions on the Web (TWEB) 6,* 2, 1–44.

Domingues, P., Sousa, B., and Moura Silva, L. 2007. Sabotage-tolerance and Trust Management in Desktop Grid Computing. *Future Generation Computer Systems: The Int. Journal of Grid Computing and eScience 23,* 7, 904–912.

Douceur, J. R. 2002. The Sybil Attack. In *Proc. of the 1st Int. Workshop on Peer-to-Peer Systems (IPTPS'02)*. Cambridge, MA, USA.

eBay. 2011. ebay - new & used electronics, cars, apparel, collectibles, sporting goods & more at low prices. Accessed 01/3/2011, Available at: http://www.ebay.com/.

Ellison, C. 1996. Establishing Identity Without Certification Authorities. In *Proc. of the 6th Conf. on USENIX Security Symposium (SSYM'96), Focusing on Applications of Cryptography-Volume 6*. San Jose, CA, USA.

Ellison, C., Frantz, B., Lampson, B., Rivest, R., Thomas, B., and Ylonen, T. 1999. SPKI Certificate Theory.

Epinions.com. 2011. Reviews from epinions. Accessed 01/3/2011, Available at: http://www1.epinions.com/.

Fernandez-Gago, M., Roman, R., and Lopez, J. 2007. A survey on the Applicability of Trust Management Systems for Wireless Sensor Networks. In *Proc. of the 3rd Int. Workshop on Security, Privacy and Trust in Pervasive and Ubiquitous Computing (SECPerU'07)*. Istanbul, Turkey.

Foster, I., Zhao, Y., Raicu, I., and Lu, S. 2008. Cloud Computing and Grid Computing 360-degree Compared. In *Proc. of Grid Computing Environments Workshop (GCE'08)*. Texas, USA.

Friedman, E., Resnick, P., and Sami, R. 2007. *Algorithmic Game Theory*. Cambridge University Press, New York, USA, Chapter Manipulation-Resistant Reputation Systems, 677–697.

Fung, B., Wang, K., Chen, R., and Yu, P. 2010. Privacy-preserving Data Publishing: A Survey of Recent Developments. *ACM Computing Surveys (CSUR) 42,* 4, 1–53.

Google-Apps. 2011. Google Apps. Accessed 03/04/2011, Available at: http://www.google.com/apps/intl/en-au/business/index.html#utm_campaign=en-au&utm_source=en-ha-apac-au-bk- google&utm_medium=ha&utm_term=google%20app.

Google-Docs. 2011. Google Docs - Online documents, spreadsheets, presentations, surveys, file storage and more. Accessed 11/04/2011, Available at: https://docs.google.com/.

Gottfrid, D. 2007. Self-Service, Prorated Supercomputing Fun. *The New York Times*. Accessed: 20/04/2011, Available at: http://open.blogs.nytimes.com/2007/11/01/self-service-prorated-super-computing-fun/.

Grandison, T. and Sloman, M. 2000. A Survey of Trust in Internet Applications. *IEEE Communications Surveys and Tutorials 3,* 4, 2–16.

Habib, S., Ries, S., and Muhlhauser, M. 2011. Towards a Trust Management System for Cloud Computing. In *Proc. of IEEE 10th Int. Conf. on Trust, Security and Privacy in Computing and Communications (TrustCom'11)*. Changsha, China.

Hoffman, K., Zage, D., and Nita-Rotaru, C. 2009. A Survey of Attack and Defense Techniques for Reputation Systems. *ACM Computing Surveys (CSUR) 42,* 1, 1–31.

Hussain, A., Heidemann, J., and Papadopoulos, C. 2003. A Framework for Classifying Denial of Service Attacks. In *Proc. of the 2003 conf. on Applications, technologies, architectures, and protocols for computer communications (SIGCOMM'03)*. Karlsruhe, Germany.

Huynh, T., Jennings, N., and Shadbolt, N. 2006. Certified Reputation: How an Agent Can Trust a Stranger. In *Proc. of the 5th Int. joint Conf. on Autonomous Agents and Multiagent Systems (AAMAS'06)*. Hakodate, Hokkaido, Japan.

Hwang, K., Kulkareni, S., and Hu, Y. 2009. 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)*. Chengdu, China.

Hwang, K. and Li, D. 2010. Trusted Cloud Computing with Secure Resources and Data Coloring. *IEEE Internet Computing 14,* 5, 14–22.

Jøsang, A., Ismail, R., and Boyd, C. 2007. A Survey of Trust and Reputation Systems for Online Service Provision. *Decision Support Systems 43,* 2, 618–644.

Kamvar, S., Schlosser, M., and Garcia-Molina, H. 2003. The Eigentrust Algorithm for Reputation Management in P2P Networks. In *Proc. of the 12th Int. Conf. on World Wide Web (WWW'03)*. Budapest, Hungary.

Ko, R., Jagadpramana, P., Mowbray, M., Pearson, S., Kirchberg, M., Liang, Q., and Lee, B. 2011. TrustCloud: A Framework for Accountability and Trust in Cloud Computing. In *Proc. of IEEE World Congress on Services (SERVICES'11)*. Washington, DC, USA.

Koshutanski, H. and Massacci, F. 2007. A Negotiation Scheme for Access Rights Establishment in Autonomic Communication. *Journal of Network and Systems Management 15,* 1, 117–136.

Krautheim, F., Phatak, D., and Sherman, A. 2010. Introducing the Trusted Virtual Environment Module: A New Mechanism for Rooting Trust in Cloud Computing. In *Proc. of the 3rd Int. Conf. on Trust and Trustworthy Computing (TRUST'10)*. Berlin, Germany.

Lai, K., Feldman, M., Stoica, I., and Chuang, J. 2003. Incentives for Cooperation in Peer-to-Peer Networks. In *Proc. of the 1st Workshop on Economics of Peer-to-Peer Systems*. Berkeley, CA, USA.

Lin, C., Varadharajan, V., Wang, Y., and Pruthi, V. 2004. Enhancing Grid Security with Trust Management. In *Proc. of the 2004 IEEE Int. Conf. on Services Computing (SCC'04)*. Shanghai, China.

Liu, G., Wang, Y., and Orgun, M. 2009. Trust Inference in Complex Trust-oriented Social Networks. In *Proc. of IEEE 12th Int. Conf. on Computational Science and Engineering (CSE'09)*. Vancouver, Canada.

Malik, Z. and Bouguettaya, A. 2009a. Rater Credibility Assessment in Web Services Interactions. *World Wide Web 12,* 1, 3–25.

Malik, Z. and Bouguettaya, A. 2009b. RATEWeb: Reputation Assessment for Trust Establishment Among Web services. *The VLDB Journal 18,* 4, 885–911.

Malik, Z. and Bouguettaya, A. 2009c. Reputation Bootstrapping for Trust Establishment Among Web Services. *IEEE Internet Computing 13,* 1, 40–47.

Manuel, P., Thamarai Selvi, S., and Barr, M.-E. 2009. Trust Management System for Grid and Cloud Resources. In *Proc. of the 1st Int. Conf. on Advanced Computing (ICAC'09)*. Chennai, India.

Marti, S. and Garcia-Molina, H. 2006. Taxonomy of Trust: Categorizing P2P Reputation Systems. *Computer Networks 50,* 4, 472–484.

Matsuo, Y. and Yamamoto, H. 2009. Community Gravity: Measuring Bidirectional Effects by Trust and Rating on Online Social Networks. In *Proc. of the 18th Int. Conf. on World Wide Web (WWW'09)*. Madrid, Spain.

Medjahed, B., Benatallah, B., Bouguettaya, A., Ngu, A., and Elmagarmid, A. 2003. Business-to-Business Interactions: Issues and Enabling Technologies. *The VLDB Journal 12,* 1, 59–85.

Mell, P. and Grance, T. 2011. The NIST Definition of Cloud Computing. Accessed: 05/06/2012, Available at: http://csrc.nist.gov/publications/drafts/800-145/Draft-SP-800-145_cloud-definition. pdf.

Microsoft. 2011. Windows Live Mesh 2011. Accessed 09/05/2011, Available at:https://www.mesh.com/.

Noor, T. H. and Sheng, Q. Z. 2011a. Credibility-Based Trust Management for Services in Cloud Environments. In *Proc. of the 9th Int. Conf. on Service Oriented Computing (ICSOC'11)*. Paphos, Cyprus.

Noor, T. H. and Sheng, Q. Z. 2011b. Trust as a Service: A Framework for Trust Management in Cloud Environments. In *Proc. of the 12th Int. Conf. on Web and Information Systems (WISE'11)*. Sydney, Australia.

Park, S., Liu, L., Pu, C., Srivatsa, M., and Zhang, J. 2005. Resilient Trust Management for Web Service Integration. In *Proc. of IEEE Int. Conf. on Web Services (ICWS'05)*. Orlando, Florida.

Pashalidis, A. and Mitchell, C. 2003. A Taxonomy of Single Sign-on Systems. In *Proc. of the 8th Australasian Conf. on Information Security and Privacy (ACISP'03)*. Wollongong, Australia.

Pearson, S. and Benameur, A. 2010. Privacy, Security and Trust Issues Arising From Cloud Computing. In *Proc. IEEE 2nd Int. Conf. on Cloud Computing Technology and Science (CloudCom'10)*. Indianapolis, Indiana, USA.

Ren, K., Wang, C., and Wang, Q. 2012. Security Challenges for the Public Cloud. *IEEE Internet Computing 16,* 1, 69–73.

Rosen, G. 2011. Jack of All Clouds. Accessed 25/02/2011, Available at: http://www.jackofallclouds.com/2011/01/state-of-the-cloud-january-201/.

Roy, I., Setty, S., Kilzer, A., Shmatikov, V., and Witchel, E. 2010. Airavat: Security and Privacy for MapReduce. In *Proc. of the 7th USENIX Symp. on Networked Systems Design and Implementation (NSDI'10)*. San Jose, CA, USA.

Ruohomaa, S. and Kutvonen, L. 2005. Trust Management Survey. In *Proc. of the 3rd Int. Conf. on Trust Management (iTrust'05)*. Paris, France.

Sabater, J. and Sierra, C. 2005. Review on Computational Trust and Reputation Models. *Artificial Intelligence Review 24,* 1, 33–60.

Santos, N., Gummadi, K., and Rodrigues, R. 2009. Towards Trusted Cloud Computing. In *Proc. of 2009 Workshop on Hot Topics in Cloud Computing (HotCloud'09)*. San Diego, CA, USA.

Seamons, K., Winslett, M., and Yu, T. 2001. Limiting the Disclosure of Access Control Policies During Automated Trust Negotiation. In *Proc. of the Symp. on Network and Distributed System Security (NDSS'01)*. San Diego, CA, USA.

Silaghi, G., Arenas, A., and Silva, L. 2007. Reputation-based Trust Management Systems and Their Applicability to Grids. Tech. Rep. Core-GRID (TR-0064), Institute on Knowledge and Data Management Institute on System Architecture, Coimbra, Portugal.

Skogsrud, H., Benatallah, B., Casati, F., Toumani, F., and Australia, T. 2007. Managing Impacts of Security Protocol Changes in Service-oriented Applications. In *Proc. of the 29th Int. Conf. on Software Engineering, (ICSE'07)*. Minneapolis, MN, USA.

Skogsrud, H., Motahari-Nezhad, H., Benatallah, B., and Casati, F. 2009. Modeling Trust Negotiation for Web Services. *Computer 42,* 2, 54–61.

Skopik, F., Schall, D., and Dustdar, S. 2009. Start Trusting Strangers? Bootstrapping and Prediction of Trust. In *Proc. of the 10th Int. Conf. on Web Information Systems Engineering (WISE'09)*. Poznan, Poland.

Skopik, F., Schall, D., and Dustdar, S. 2010. Trustworthy Interaction Balancing in Mixed Service-Oriented Systems. In *Proc. of ACM 25th Symp. on Applied Computing (SAC'10)*. Sierre, Switzerland.

Song, S., Hwang, K., and Kwok, Y. 2005. Trusted Grid Computing with Security Binding and Trust Integration. *Journal of Grid computing 3,* 1, 53–73.

Song, S., Hwang, K., Zhou, R., and Kwok, Y. 2005. Trusted P2P Transactions with Fuzzy Reputation Aggregation. *IEEE Internet Computing 9,* 6, 24–34.

Sotomayor, B., Montero, R., Lorente, I., and Foster, I. 2009. Virtual Infrastructure Management in Private and Hybrid Clouds. *IEEE Internet Computing 13,* 5, 14–22.

Srivatsa, M. and Liu, L. 2006. Securing Decentralized Reputation Management Using TrustGuard. *Journal of Parallel and Distributed Computing 66,* 9, 1217–1232.

Srivatsa, M., Xiong, L., and Liu, L. 2005. TrustGuard: Countering Vulnerabilities in Reputation Management for Decentralized Overlay Networks. In *Proc. of the 14th Int. Conf. on World Wide Web (WWW'05)*. Chiba, Japan.

Suryanarayana, G. and Taylor, R. 2004. A Survey of Trust Management and Resource Discovery Technologies in Peer-to-Peer Applications. Tech. Rep. UCI-ISR-04-6, The University of California, Irvine, California, USA.

Thomas, D. and Hun, A. 2002. State Machines. *IEEE Software 19*, 10–12.

Viega, J. 2009. Cloud Computing and the Common Man. *Computer 42,* 8, 106–108.

Wang, Y. and Vassileva, J. 2007. Toward Trust and Reputation Based Web Service Selection: A Survey. *International Transactions on Systems Science and Applications 3,* 2, 118–132.

Wei, W., Du, J., Yu, T., and Gu, X. 2009. SecureMR: A Service Integrity Assurance Framework for MapReduce. In *Proc. of the Annual Computer Security Applications Conf. (ACSAC'09)*. Honolulu, Hawaii, USA.

Wei, Y. and Blake, M. B. 2010. Service-oriented Computing and Cloud Computing: Challenges and Opportunities. *Internet Computing, IEEE 14,* 6, 72–75.

Weishaupl, T., Witzany, C., and Schikuta, E. 2006. gSET: Trust Management and Secure Accounting for Business in the Grid. In *Proc. of the 6th IEEE Int. Symp. on Cluster Computing and the Grid (CCGrid'06)*. Singapore.

Xiong, L. and Liu, L. 2003. A Reputation-based Trust Model for Peer-to-Peer e-commerce Communities. In *Proc. of IEEE Int. Conf. on e-Commerce (CEC'03)*. Newport Beach, CA, USA.

Xiong, L. and Liu, L. 2004. Peertrust: Supporting Reputation-based Trust for Peer-to-Peer Electronic Communities. *IEEE Transactions on Knowledge and Data Engineering 16,* 7, 843–857.

Yao, J., Chen, S., Wang, C., Levy, D., and Zic, J. 2010. Accountability as a Service for the Cloud. In *Proc. of IEEE Int. Conf. on Services Computing (SCC'10)*. Miami, FL, USA.

Yu, C. and Ng, K. 2006. A Mechanism to Make Authorization Decisions in Open Distributed Environments without Complete Policy Information. In *Proc. of the Int. Conf. on Computational Science (ICCS'06)*. Reading, UK.

Yu, C. and Ng, K. 2009. DPMF: A Policy Management Framework for Heterogeneous Authorization Systems in Grid Environments. *Multiagent and Grid Systems 5,* 2, 235–263.

Zhou, R. and Hwang, K. 2006. Trust Overlay Networks for Global Reputation Aggregation in P2P Grid Computing. In *Proc. of the 20th Int. Symp. on Parallel and Distributed Processing (IPDPS'06)*. Rhodes Island, Greece.

Zhou, R. and Hwang, K. 2007. Powertrust: A Robust and Scalable Reputation System for Trusted Peer-to-Peer Computing. *IEEE Transactions on Parallel and Distributed Systems 18,* 5, 460–473.

Ziegler, C. and Golbeck, J. 2007. Investigating Interactions of Trust and Interest Similarity. *Decision Support Systems 43,* 2, 460–475.