Chapter 2
Background

An introduction to the research fields related to the trust management of services in cloud environments is given in this chapter to help readers gain a better understanding of the work described in this book. In particular, an overview of cloud services models and trust management techniques are presented in chapter. Furthermore, a generic framework is proposed to compare representative research prototypes and compare major cloud service providers [111].

This chapter is organized as follows. In Sects. 2.1 and 2.2, an overview of cloud services and their deployment models, and trust management techniques, are presented respectively. In Sect. 2.3, an analytical framework for trust management is proposed and a set of dimensions are identified for each layer in the framework to be used for comparing trust management solutions. In Sect. 2.4, 29 representative research prototypes are discussed and evaluated. In Sect. 2.5, several major cloud service providers are compared from a trust perspective. Finally, this chapter is summarized in Sect. 2.6.

2.1 Overview of Services in Cloud Environments

Cloud services are established based on five essential characteristics [96], 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. Based on the definition provided by the National Institute of Standards and Technology (NIST) [96], cloud computing can be defined as follows:

T. H. Noor et al., *Trust Management in Cloud Services,*
DOI 10.1007/978-3-319-12250-2_2

Definition 2.1 (Cloud Computing) *Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. This cloud model is composed of five essential characteristics, three service models, and four deployment models.* □

2.1.1 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 consumer [26, 33, 96]. Figure 2.1 depicts the structured layers of cloud services:

- *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) [6] and Simple Storage Service (S3) [7]). 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 [62]). 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 [63], Windows Live Mesh [100]). 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.

2.1.2 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

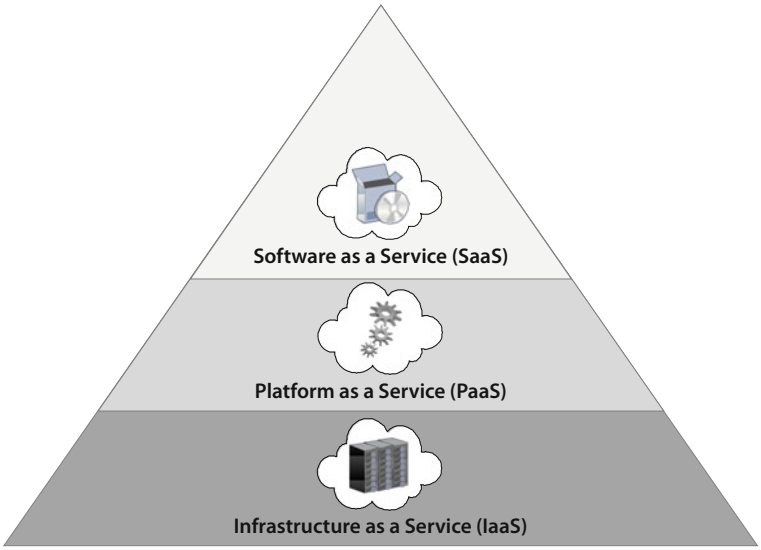

Fig. 2.1 Cloud service models

models, namely *Private*, *Community*, *Public*, and *Hybrid* [96, 138] depending on the cloud service consumer's needs. Figure 2.2 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 Fig. 2.2a), 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 Fig. 2.2b, 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 Fig. 2.2c). 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.

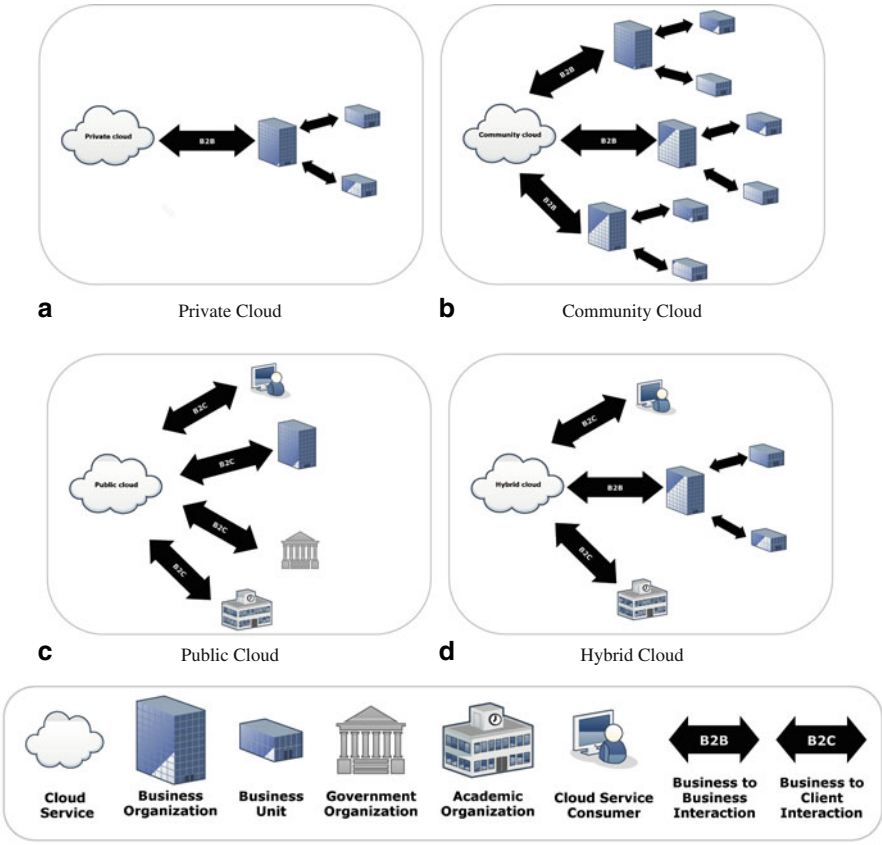

Fig. 2.2 Cloud service deployment models

- *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 Fig. 2.2b). 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.

2.2 Overview of Trust Management

Trust management is originally developed by Blaze et al. [22] 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 [21, 23–25]. 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, Peer-to-Peer (P2P) systems, Web services, wireless sensor networks, grid computing, and most recently cloud computing. There are several trust definition reported in the literature from different perspectives. However, we agree with the one provided by Jøsang et al. [76]. So for this work we use the following definition:

Definition 2.2 (Trust) *Trust is the extent to which a cloud service consumer is willing to depend on a cloud service provider, provisioning a cloud service and expects certain qualities that the cloud service provider promised to be met.* □

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 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 (Fig. 2.3a). On the other hand, in SRP, the service requester is the one who assesses the trustworthiness of the service provider (Fig. 2.3b).

2.2.1 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).

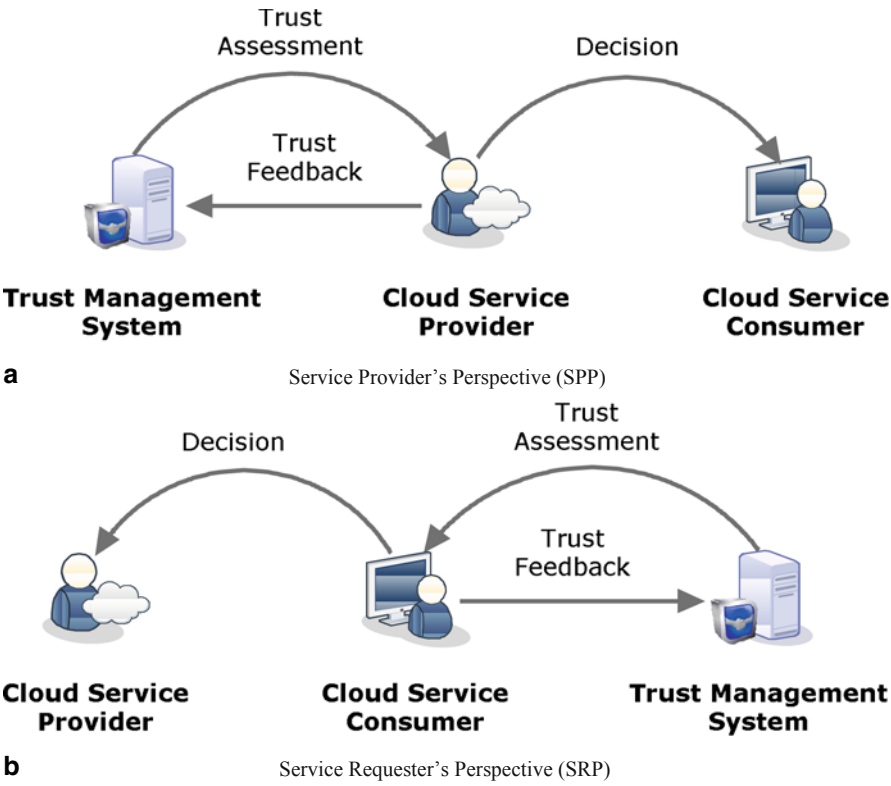

Fig. 2.3 Trust management perspectives

Figure 2.4 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.

2.2.1.1 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 [4, 127, 154], the grid [136], P2P systems [137], Web applications [44] and the service-oriented environment [132, 133]. 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.

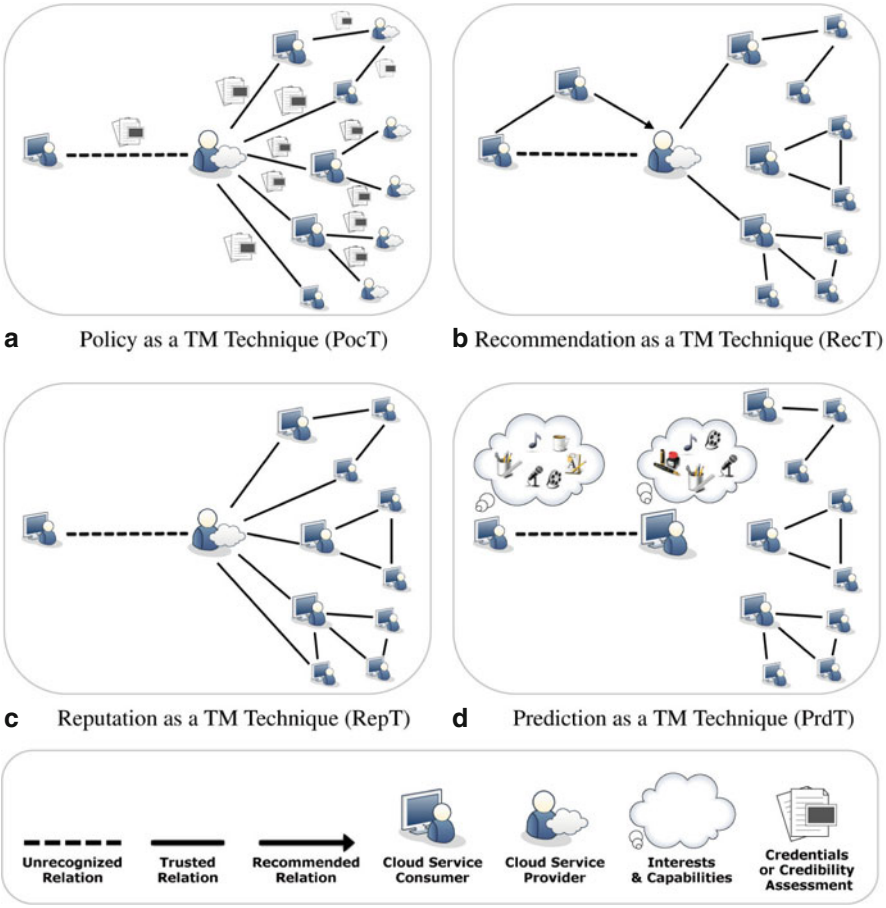

Fig. 2.4 Trust Management (TM) techniques

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 [72] 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 [67].

The feedback credibility [152] can be measured using several factors such as cloud service consumers' experience (i.e., the quality of feedbacks differs from one person to another [102]). Many researchers identify two features of credibility including trustworthiness and expertise [3, 89, 102, 140, 153].

For credential-based threshold, PocT follows either the Single-Sign-On (SSO) approach [114] 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 [143] 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 [36], the Simple Public Key Infrastructure (SPKI) [53], or the Security Assertion Markup Language (SAML) [29]. Many researchers use the digital certificates perspective to define the credential term [18, 28, 128] 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 [54] where the certificate authority is not required.

Figure 2.4a 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 Eq. 2.1).

$$\mathcal{T}r\,(x,\,y) = \begin{cases} 1 & if \quad \mathcal{C}_x \geq \mathcal{T}_y \Leftrightarrow \mathcal{C}_y \geq \mathcal{T}_x \;\; or \;\; \mathcal{R}_y \geq \mathcal{T}_x \Leftrightarrow \mathcal{R}_x \geq \mathcal{T}_y \\ 0 & otherwise \end{cases} \tag{2.1}$$

The literature reports some efforts of PocT in cloud computing. For example, Brandic et al. [26] propose a novel language for specifying compliance requirements based on a model-driven technique and Ko et al. [80] present a TrustCloud framework that uses SLA detective controls and monitoring techniques for achieving trusted cloud services. Hwang et al. [73, 74] propose a security aware cloud architecture that uses pre-defined policies to evaluate the credibility of cloud services and Habib et al. [67] 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 [102, 103] propose a credibility model that distinguishes credible feedbacks from the misleading ones. PocT is applicable for all three cloud service models.

2.2.1.2 Recommendation as a Trust Management Technique (RecT)

Recommendation as a trust management technique (RecT) has been widely used in the cloud environment [67, 82], the grid [49], and the service-oriented environment [113, 134]. 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 [86]. 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 his/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 his/her trusted relations trust the service.

Figure 2.4b 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 Eq. 2.2.

$$\mathcal{T}r\,(x,\ y \mid \mathcal{T}r\,(z,\ y)) = \begin{cases} 1 & if \quad \mathcal{T}r\,(z,\ y) = 1 \\ 0 & otherwise \end{cases} \tag{2.2}$$

One of the recent efforts using RecT in cloud computing is reported in [67]. 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 [82], 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.

2.2.1.3 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 [67, 82, 92, 102, 103], the grid [13–15, 85], P2P [8, 9, 41, 42, 77, 139, 140, 153, 158, 159], as well as the service-oriented environment [35, 89–91, 113]. Reputation can have direct or indirect influence on the trustworthiness of a particular entity (e.g., cloud service) as pointed in [3]. 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 Fig. 2.4b and 2.4c. There are several online reputation-based systems such as the auction systems (e.g., eBay [52] and Amazon [5]) where new and used goods are found, and the review systems [55] where the consumers opinions and reviews on specific products or services are expressed.

Figure 2.4c 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 $\mathcal{T}r(y) = \{r_1, r_2, \ldots, r_i\}$ (i.e., with other cloud service consumers), which give trust feedbacks on the cloud service provider $\mathcal{T}f(y) = \{f_1, f_2, \ldots, f_n\}$. These feedbacks are used to calculate the reputation of y, denoted as $\mathcal{R}ep(y)$, as shown in Eq. 2.3. 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.

$$\mathcal{R}ep(y) = \frac{\sum_{x=1}^{|\mathcal{T}f(y)|} \mathcal{T}f(x, y)}{|\mathcal{T}f(y)|} \tag{2.3}$$

$$\mathcal{T}r(x, y) = \begin{cases} 1 & if \quad \mathcal{R}ep(y) \geq \mathcal{T}_x \\ 0 & otherwise \end{cases} \tag{2.4}$$

Similarly, there exist several efforts that use RepT in trust management of cloud computing. Habib et al. [67] 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 [82], a reputation-based trust model is proposed that focuses on Infrastructure as a Service (IaaS) cloud services. Noor and Sheng [102, 103] propose a reputation-based trust management framework that distinguishes the credible feedbacks from the misleading ones.

2.2.1.4 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) [134]. PrdT has been proposed in the cloud environment [67, 102, 103] and the service-oriented environment [134, 135]. The basic idea behind PrdT is that *similar minded entities* (e.g., cloud service consumers) are more likely to trust each other [94, 160].

Figure 2.4d 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, $i_x = (i_1, i_2, \ldots, i_j)$, 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, $i_y = (i_1, i_2, \ldots, i_k)$, and a certain minimum trust threshold $\mathcal{T}_y$ is also used to determine

http://www.springer.com/978-3-319-12249-6

Trust Management in Cloud Services

Noor, T.H.; Sheng, Q.Z.; Bouguettaya, A.
2014, XIX, 119 p. 39 illus., Hardcover
ISBN: 978-3-319-12249-6