Enabling Trust Assessment In Clouds-of-Clouds: A Similarity-Based Approach

Reda Yaich, Nora Cuppens, Frédéric Cuppens
IMT Atlantique
Lab-STICC, UMR CNRS 6285
Institut Mines-Télécom
{first.last}@imt-atlantique.fr

ABSTRACT

In multi-cloud paradigm, cloud providers collaborate to form ad-hoc and ephemeral groups to fulfill the request of a single customer. In such settings, malevolent cloud providers may be tempted to provide cloud services that are below the expected quality. This temptation is further exacerbated by the inability of customers to effectively identify the responsible of service outage or degradation.

Furthermore, the highly competitive nature of cloud marketplaces leads each provider to propose regularly innovative new services, making the system open and highly dynamic. The introduction of new cloud services into the system challenges the established trust order as customers and providers must accept the risk of taking decisions under uncertainty. This problem, known as the cold-start problem, have been studied in the literature from the perspective of the individuals (providers/customers) but to the best of our knowledge, no prior work tried to address it from the perspective of the exchanged services and resources.

To that aim, we propose in this paper a similarity-based trust model that tackles both multi-cloud (i.e., group-reputation) and services high turnover (i.e., cold-start). In our model, past similar experiences are transferred to the providers proposing new services to enable and boost decision making and collaboration. We propose also a schema to derive multi-cloud trust using both customers and providers feedback experiences. We present also evaluations results to show the benefit of using our proposal and their impact on the simulated cloud-marketplace.

CCS CONCEPTS

•**Security and privacy** → **Social aspects of security and privacy;** *Web application security;* •**Applied computing** → *E-commerce infrastructure;*

KEYWORDS

Trust, Multi-Clouds, Group-Reputation, Cold-Start, Similarity

ARES '17, Reggio Calabria, Italy

978-1-4503-5257-4/17/08…$15.00
DOI: 10.1145/3098954.3098970

ACM Reference format:
Reda Yaich, Nora Cuppens, Frédéric Cuppens. 2017. Enabling Trust Assessment In Clouds-of-Clouds:
A Similarity-Based Approach. In *Proceedings of ARES '17, Reggio Calabria, Italy, August 29-September 01, 2017,* 9 pages.
DOI: 10.1145/3098954.3098970

1 INTRODUCTION

Multi-Cloud paradigm, or Clouds-of-Clouds [15], is the concomitant use of multiple cloud infrastructures to mainly get benefit from at least one of the following advantages: (a) minimize the "vendor lockin" risk that arise in tradition single cloud approaches, (b) improve the availability and fault-tolerance of cloud services using load balancing workloads between providers, and (c) comply with geographical proximity constraints imposed by business strategies or legal requirements[1] reducing services' vulnerability to denial of service attacks, and services outage. Consequently, companies are actively trying to avail themselves of the best from each offer.

In this paper, we focus on Cloud Market Places (CMP) wherein cloud providers (CSPs) and cloud customers (CSCs) interact around cloud services. The services desired by a customer are expressed using requests and those offered by the provider are described using offers. The interactions between Cloud Providers and Cloud Consumers are formalized as agreements (Service Level Agreements). An agreement is a legal contract that binds a cloud customer with the provider that is responsible of fulfilling his request. Unlike traditional approaches, in this work we are particularly interested in multi-clouds situations in which the request of a customer can only be fulfilled by multiple-providers. In sum, the Cloud Market Place (CMP) System Model can be defined at a time t by:

$$CMP = \langle C, Q, \mathcal{P}, O, \mathcal{S}, \mathcal{M}, \mathcal{A}, \mathcal{E} \rangle^t \quad (1)$$

Where $C = \{c1, c_2, \dots\}$ is the set of customers, $Q = \{q_1, q_2, \dots\}$ is the set of queries, $\mathcal{P} = \{p_1, p_2, \dots, p_l\}$ is the set of providers, $O = \{o_1, o_2, \dots\}$ is the set of offers, $\mathcal{S} = \{s_1, s_2, \dots\}$ is the set of cloud services, $\mathcal{M} = \{m_1, m_2, \dots\}$ is the set of multi-clouds, $\mathcal{A} = \{a_1, a_2, \dots\}$ is the set of agreements, $\mathcal{E} = \{e_1, e_2, \dots\}$ is the set of customers experiences. In what follows, we make use of c, r, p, o, s, m, a, e to refer to, respectively, an arbitrary customer, query, provider, offer, service, multi-cloud, agreement and experience.

Moving to multi-clouds brings to discussion serious security and privacy risks with a high potential harm to customers' and users' data and services. Indeed, the infrastructure offered by an

[1] European Level Directive 95/46/EC and the recent General Data Protection Regulation (GDPR) released in 2016 with with all European companies must comply with.

untrusted provider can be considered as a hostile environment wherein security objectives cannot be guaranteed. Thus, despite deploying appropriate security mechanisms, cloud provider must provide sufficient guarantees to gain customers' trustworthiness.

1.1 Problem Statement

In the cloud ecosystem, Security Service Level Agreements (SSLAs) are considered as a good trust enabler as they represent a legal document that certifies the providers' willingness to meet customers's expected Quality-of-Service (QoS) and Quality-of-Protection (QoP) [15]. From the cloud provider perspective, Qo(S&P) metrics provide a good indicator of the infrastructure capacities, while the same metrics from the customer perspective testify about the performances experienced by the cloud customer. The processing of Qo(S&P) metrics usually includes a customers × providers experiences matrix, as shown hereafter. Each matrix represent the experiences that all customers of the system ($\forall c \in C$) had with cloud providers (i.e., $\forall p \in \mathcal{P}$) for a particular cloud service $s \in \mathcal{S}$. Rows represent experiences issued by a certain customer, while columns reflect the experiences expressed with respect to a particular provider[2].

$$\mathcal{E}^s_{C:\mathcal{P}} = \begin{array}{c} \\ c_1 \\ c_2 \\ \vdots \\ c_y \end{array} \begin{array}{c} \begin{array}{cccc} p_1 & p_2 & \cdots & p_n \end{array} \\ \begin{pmatrix} \mathcal{E}^s_{c_1:p_1} & \mathcal{E}^s_{c_1:p_2} & \cdots & \mathcal{E}^s_{c_1:p_y} \\ \mathcal{E}^s_{c_2:p_1} & \mathcal{E}^s_{c_2:p_2} & \cdots & \mathcal{E}^s_{c_2:p_y} \\ \vdots & \vdots & \ddots & \vdots \\ \mathcal{E}^s_{c_x:p_1} & \mathcal{E}^s_{c_x:p_2} & \cdots & \mathcal{E}^s_{c_x:p_y} \end{pmatrix} \end{array}$$

The approach we advocate in this paper is to make use of theses experiences when assessing the trust that a customer can put in the candidate multi-cloud. As illustrated in Figure 1, before making a decision about the provider to engage with, for a specific service s, the requesting customer c will make use of a trust model to derive a trust value based on past-experiences. Then during the transaction (i.e., Cloud Service Delivery), the CSC c will make use of monitoring mechanisms to observe the behavior of the provider. We make the reasonable assumption that all service level objectives (SLOs) conveyed in an SSLA agreement (i.e., $a \in \mathcal{A}$) can be monitored and that monitoring information are reliable and could not be tempered.

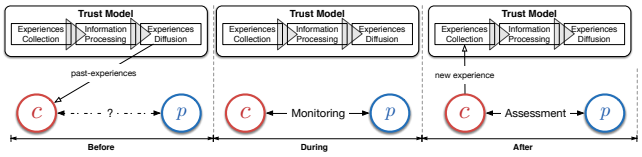

Figure 1: Classical Feedback-Based Trust Assessement

Nevertheless, the problem we address in this paper is different as providers are grouped into multi-clouds, as illustrated in Figure 2, wherein they engage to satisfy *"collectively "* the query of a unique customer. In such settings, dishonest cloud providers may be tempted to deliver services and protection mechanisms which quality level do not meet the expected/agreed standards. This situation is particularly interesting as from the customer perspective, its the whole multi-clouds that is responsible of any failure, and calls for appropriate mechanisms to assist clou d customers in selecting appropriate candidates.

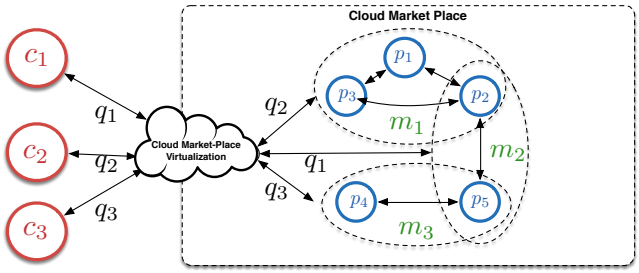

Figure 2: An example of a Cloud Market Place composed of three multi-Clouds

Here, the virtualization infrastructure (e.g., SUPERCLOUD Framework [15, 21]) that coordinates and orchestrates the uniform delivery of cloud-services, prevents the customers from having access to details about the concrete provider delivering the services under use. Consequently, in case of any service disruption, the customer can only identify the multi-cloud, or at least the set of providers that delivering that type of service, as responsible of that failure.

Therefore, we assume that, similarly to what is done by customers, the providers' processing of internal Qo(P&S) metrics implies the existence of a Providers × Providers experiences matrix as illustrated hereafter.

$$\mathcal{E}^s_{\mathcal{P}:\mathcal{P}} = \begin{array}{c} \\ p_1 \\ p_2 \\ \vdots \\ p_y \end{array} \begin{array}{c} \begin{array}{cccc} p_1 & p_2 & \cdots & p_x \end{array} \\ \begin{pmatrix} \emptyset & \mathcal{E}^s_{p_1:p_2} & \cdots & \mathcal{E}^s_{p_1:p_y} \\ \mathcal{E}^s_{p_2:p_1} & \emptyset & \cdots & \mathcal{E}^s_{p_2:p_y} \\ \vdots & \vdots & \ddots & \vdots \\ \mathcal{E}^s_{p_x:p_1} & \mathcal{E}^s_{p_x:p_2} & \cdots & \emptyset \end{pmatrix} \end{array}$$

The above matrix is traditionally used by cloud providers to select the best candidate provider to collaborate with within a multi-cloud (cf., [20]). In our approach, we advocate the extension of it's use to customers as it contains accurate and valuable information about the effective level of fulfillment of each provider in their previous interactions [19].

1.2 Contributions

In order to address the aforementioned issues, we first propose to extend the group reputation theory [3] to include group-level reflexive reputation. In other words, customers will make use of provider's experiences with each other during trust assessment.

Furthermore, the highly competitive nature of cloud marketplaces leads each provider to propose regularly innovative new services, making the system open and highly dynamic. The introduction of new cloud services into the system challenges the established trust order as customers and providers have to make decision under uncertainty. The uncertainty is due to the absence/lack of quantitative and qualitative monitoring data about the behavior of cloud providers in delivering these services.

[2]The exact semantic of each experience $\mathcal{E}^s_{x:y}$ is presented in Section 2.1

To that aim, we propose a similarity-based trust model in which the lack of experiences of new services is compensated by transferring similar experiences. The intuition behind our proposal is that providers showing good fulfillment level of services of a certain type are more likely (and competent) to exhibit similar behavior with newly introduced services.

1.3 Paper structure

The rest of the paper is organized as follows. In Section 2 we present the details of the similarity-based trust model we propose to tackle the problems introduced previously. Then in Section 3 we describe the experimental results we obtained in order to evaluate the benefits of our approach. In Section 4, we review the related works. Finally, Section 5 concludes this paper.

2 TRUST MANAGEMENT ISSUE WITHIN CLOUDS-OF-CLOUDS

Our trust model takes advantage of the assessments that customers and providers establish after each transaction to express their degree of satisfaction towards the quality of service provided by the collaborating partner. We advocate the use of cloud monitoring services to detect any deviation from the expected quality of service agreed upon by cloud customers and providers. Individual experiences are thus aggregated to compute a reputation that reflects the trustworthiness of a provider. Unlike traditional approaches, our model is used by CSCs to assess the trustworthiness of candidates multi-clouds as a set of cloud providers and not only a single provider. To the best of our knowledge, this is the first work that attempts to address such a composite assessment of trust in Cloud environments.

To proceed, we specify hereafter the schema we propose to make this assessment.

$$\mathcal{T}^{q}_{c:m} : (\alpha \times \mathcal{T}^{q}_{C:m}) + (|1-\alpha| \times \mathcal{T}^{q}_{\mathcal{P}:m}) \tag{2}$$

As stated in Equation 2, the trust of a cloud customer towards a set of providers organized into a multi-cloud is function of the *CSCs Trust* $\mathcal{T}^{q}_{C:m}$ and the *CSPs Trust* $\mathcal{T}^{q}_{\mathcal{P}:m}$. *CSCs Trust* refers to the trust that the customers (i.e., $\forall c \in C$) are willing to put into the providers that constitute the multi-cloud m. Analogously, the *CSPs Trust* reflects the trust that providers (i.e., $\forall p \in \mathcal{P}$) are willing to put in each other. We make use of $\alpha \in [0,1]$ to balance the importance of each type of trust in the final trust value.

In what follows, we first show how experiences are aggregated to compute a reputation value. Then we describe how this value is used to compute both customers and providers trust.

2.1 Experiences aggregation

In state-of-the-art trust and reputation approaches [10, 11], the trust that a customer c is willing to put into a provider is function of the experience derived from prior transactions. The experiences constitute customers' and providers' feedback and reflect their level of satisfaction with respect to the expected quality of service and protection. We denote $\mathcal{E}^{s}_{c:p} \subseteq \mathcal{E}$ the chronologically ordered set of experiences issued by the customer c towards the provider p for a specific delivered service s. Each experience $e_i^{\langle c,p,s\rangle} \in \mathcal{E}^{s}_{c:p}$ is stored in the system as a quintuplet [3]:

$$e_i^{\langle c,p,s\rangle} = \langle c, p, s, v, t\rangle \tag{3}$$

$e_i^{\langle c,p,s\rangle}$ maps the services provided by p towards a customer c at a time t to a fulfillment level v. For simplicity, we make use of a normalized rating scale of $[0,1]$. For instance, in the following experience example $\langle c, p, availability, 0.9995, t\rangle$, the monitored value $99,95\%$ (corresponds to the QoS Level Objective that represents availability of cloud services) is mapped to the normalized 0.9995 value. We assume that categorical values are mapped to $true$ if the service level is met and $false$ if not. These values are then converted to, respectively, 0 and 1.

The aggregation of individual and collective experiences constitute a reputation value. Computing the reputation of a set of providers (i.e., multi-cloud) is known in the literature as group reputation [3]. Few works tried to address it and no real consensus exists about how to obtain it. In this paper, we get inspired from the Simple Additive Weighting approach [12] and propose to proceed in two steps:

(1) First, we compute individual reputation of each provider. The schema used is identical for both customers and providers and will be presented in 2.2.
(2) Then the computed individual reputation values are combined to compute a collective reputation for the candidate multi-cloud. This step is described in 2.3.

2.2 Computation of Customers' Trust

The general formula used to aggregate experiences into reputation values is defined as follows

$$R^{s}_{c:p} = \frac{\sum_{i=1}^{\lambda} \left[(e_i^{\langle c,p,s\rangle}).v \right]}{\lambda} \tag{4}$$

Where :

$$\forall e_i, e_j \in \mathcal{E}^{s}_{c:p} \mid ij \implies e_i.t \geq e_j.t$$

In Equation 4, the reputation built based on the experiences of a customer c towards a provider p for a service s is the weighted sum of the fulfillment levels v. We make use of the constant λ to express how fast the reputation value of the provider changes after each experience. The larger the value of λ, the longer the memory of the system is. In other words, the constant λ reflects the willingness of a customer to forgive past negative experiences [22]. It avoids that providers suffer too much from their initial poor behavior which may sentence all the system. Thus in the formula 4, only the λ last experiences witnessed are used to compute the reputation.

As each provider may propose different services, the aggregation of experiences need to be performed for each of the services offered by the provider. If one needs to compute the provider general reputation, a simple mean over all his *'service specific'* reputation values is sufficient to obtain that value.

[3] Experiences are issued by both providers and customers. The same format applies, indistinghusbaly, to both types.

While $R^s_{c:p}$ captures the reputation that c associates to the provider c, the open and dynamic nature of cloud marketplaces oblige use to consider situations in which c have very limited to no past experience with the candidate provider p. In such settings, the traditional way to proceed is to consider both direct (i.e., individual) experience and indirect (i.e., collective) experiences that have been relayed by other customers.

To address this issue, we build on the Formula 4 to obtain two customers reputation metrics $\vec{R}^s_{c:p}$ and $\tilde{R}^s_{c:p}$. $\vec{R}^s_{c:p}$ capture the direct reputation of a customer towards a provider as presented in Formula 4, while $\tilde{R}^s_{c:p}$ captures the provider overall reputation based on all customers experiences.

$$\tilde{R}^s_{C:p} = \sum_{i=1}^{|C|} \left[\vec{R}^s_{c_i:p}\right] \times \frac{1}{|C|} \tag{5}$$

As discussed previously, one of the particularities of the problem addressed in this paper lies in the heterogeneous nature of multi-clouds. Indeed, when a multi-cloud m candidates to fulfill a query q, the services required within the query q are planned to be fulfilled by different cloud providers. For instance, a provider p_i can be responsible of *storing* and *encrypting* data while another provider p_j will host only *unencrypt* data. Therefore, we make use of m^s to refer to the subset of a multi-cloud that is responsible of delivering the service s. Consequently, trust values are made purpose specific as they are computed based on the services the providers' are responsible of. The direct and indirect trust of customer towards a group of providers m^s delivering a service s within the multi-cloud m is specified are follows.

$$\vec{T}^s_{c:m^s} = \sum_{i=1}^{|m^s|} \left[\vec{R}^s_{c:p_i}\right] \times \frac{1}{|m^s|} \tag{6}$$

$$\tilde{T}^s_{C:m^s} = \sum_{i=1}^{|C|} \left[\vec{T}^s_{c_i:m^s}\right] \times \frac{1}{|C|} \tag{7}$$

In the above equation, the individual reputation $\vec{R}^s_{c:p_i}$ are aggregated to build individual trust assessments $\vec{T}^s_{c:m^s}$ that are later one used to build collective trust assessments $\tilde{T}^s_{C:m^s}$ Once we have computed 'service specific' trust values, we aggregate these values to derive multi-cloud level trust as follows.

$$\vec{T}^q_{c:m} = \sum_{i=1}^{|q|} \vec{T}^{s_i}_{c:m^s} \times \frac{1}{|q|} \quad \forall s_i \in q \tag{8}$$

$$\tilde{T}^q_{C:m} = \sum_{i=1}^{|q|} \vec{T}^{s_i}_{C:m^s} \times \frac{1}{|q|} \quad \forall s_i \in q \tag{9}$$

$$\tag{10}$$

At this stage, we can define how the customers's trust towards a provider $T^q_{c:p}$ is derived based on the direct (i.e., $\vec{T}^s_{c:m}$) and indirect (i.e., $\tilde{T}^s_{C:m}$) trust values.

$$\mathcal{T}^q_{C:m} = (\beta \times \vec{T}^q_{c:m}) + (|1-\beta| \times \tilde{T}^q_{C:m}) \tag{11}$$

In Formula 11, β and $|\beta - 1|$ are weights used to balance between the *direct* and *indirect* trust such as $\beta \in [0, 1]$. For instance, if $\beta = 0.5$ the customer will give equal importance to *direct* and *indirect* trust.

2.3 The Multi-Cloud Reflexive Trust

From a theoretical point of view, the trustworthiness of a group of providers organized into a multi-cloud from the provider's point of view can be modeled as an aggregation of the trust that each member of $\mathcal{P}$ is willing to put into the participants in the multi-cloud. In this section, we will describe how the experiences providers have with each other during their previous interactions as members of the same multi-cloud can be used to compute the trustworthiness of a multi-cloud.

To proceed, we first adapt the Formula 4 to compute the reputation of a provider from another provider perspective.

$$\vec{R}^s_{p_i:p_j} = \begin{cases} \vec{R}^s_{p_i:p_j} & \forall p_i, p_j \in \mathcal{P} / i \neq j \\ 1 \; if \;\; i = j \end{cases} \tag{12}$$

We assume in Formula 12 that trust is reflexive so that the reputation of a provider towards itself is optimal (i.e., $\forall p \in \mathcal{P}, R^s_{p:p} = 1$). Once the aggregation of providers' experience is done, we adapt the schema we used with cusomers trust in Section 2.2 to derive the direct and indirect trust towards the multi-cloud. The obtained formulas are reported hereafter.

$$\vec{R}^s_{m:p} = \sum_{i=1}^{|m|} \left[\vec{R}^s_{p_i:p}\right] \times \frac{1}{|\mathcal{P}|} \tag{13}$$

$$\tilde{R}^s_{\mathcal{P}:p} = \sum_{i=1}^{|\mathcal{P}|} \left[\vec{R}^s_{p_i:p}\right] \times \frac{1}{|\mathcal{P}|} \tag{14}$$

$$\vec{T}^s_{m:m^s} = \sum_{i=1}^{|m^s|} \left[\vec{R}^s_{m:p_i}\right] \times \frac{1}{|m^s|} \tag{15}$$

$$\vec{T}^s_{\mathcal{P}:m^s} = \sum_{i=1}^{|m^s|} \left[\vec{R}^s_{\mathcal{P}:p_i}\right] \times \frac{1}{|m^s|} \tag{16}$$

$$\vec{T}^q_{m:m} = \sum_{i=1}^{|q|} \vec{T}^{s_i}_{m:m^s} \quad s_i \in q \tag{17}$$

$$\tilde{T}^q_{\mathcal{P}:m} = \sum_{i=1}^{|q|} \vec{T}^{s_i}_{\mathcal{P}:m^s} \quad s_i \in q \tag{18}$$

At this stage, we can define how the providers' trust towards a multi-cloud $\mathcal{T}^q_{\mathcal{P}:m}$ is derived based on the direct (i.e., $\vec{T}^q_{m:m}$) and indirect (i.e., $\tilde{T}^q_{\mathcal{P}:m}$) multi-clouds trust values.

$$\mathcal{T}^q_{\mathcal{P}:m} = (\gamma \times \vec{T}^q_{m:m}) + (|1-\gamma| \times \tilde{T}^q_{\mathcal{P}:m}) \tag{19}$$

Here again, we assume that the trust of a multi-cloud is function of the trust of all providers towards the members of the multi-cloud and the reflexive trust of these members towards the constituted multi-cloud. Similarly to Formula 11, we make use of the weighting factor γ to balance the importance of each type of trust withing Formula 19.

2.4 Similarity-based Trust Bootstraping

We explained in Section 1.1 the inherent difficulty to obtain critical amount of experiences to build a pertinent reputation in open an dynamic environments such as Cloud Marketplaces. The settings justifying this difficulty have been characterized in the literature as the *coldstart* and *newcomers* problems [5].

We present in this Section a way to extend the use of the trust model we proposed previously to process similarity-based experiences. The objective is to allow a cloud customer to compute the trust he could put in providers proposing new services based on their past behavior delivering similar services. When requesting a newly introduced service s, the customer c will perform a similarity-based trust assessment about the multi-cloud m when the following conditions hold: (a) c is unable to acquire direct and (service s) specific trust evaluation for m and (b) no indirect experiences about m providing s can be obtained after processing $\mathcal{E}$.

To proceed, we assume the existence of a function Sim that computes the similarity between any two arbitrarily selected services among $\mathcal{S}$ [4]. We define the similarity function Sim as follows:

$$Sim : \mathcal{S} \times \mathcal{S} \rightarrow [0, 1] \tag{20}$$

The result of $Sim(s_i, s_j)$ reflects the proximity between the services s_i and s_j. We assume that the function Sim is **reflexive** (i.e., $\forall s \in \mathcal{S}, Sim(s, s) = 1$) and **symmetric** (i.e., $\forall s_i, s\ j \in \mathcal{S}, Sim(s_i, s_j) = Sim(s_j, s_i)$). We also make the assumption that if we compute $Sim(s_i, s_j) = x$ and $Sim(s_j, s_k) = y$, nothing meaningful could be inferred from x and y about the value of $Sim(s_i, s_k)$ as we believe that transitivity do not hold, unless a problem specific settings.

Based on the above similarity function, we define the set of similar services of level n with respect to the service s_i as follows:

$$\mathcal{S}^n_{s_i} = \{s \in \mathcal{S} | Sim(s_i, s) \geq \frac{n}{10}\} \mid n \in [0, 10] \tag{21}$$

In Formula 21, n reflects the minimal degree of similarity that we admit in each set. Similarity degree of 10 means that the services are identical, while a degree of 0 means that any service is admitted in the set $\mathcal{S}^0_{s_i}$ (i.e., $\forall s \in \mathcal{S}, \mathcal{S}^0_s = \mathcal{S}$). In this last setting, we fall in the classical generic trust mechanisms in which providers are associated to a generic reputation metric.

Now we build on Formula 20 and Formula 21 to introduce the extended experience set $\mathcal{E}^{s^n}_{x:y}$ of rank n. $\mathcal{E}^{s^n}_{x:y}$ is defined as follows:

$$\mathcal{E}^{s^n}_{x:y} = \{e \in \mathcal{E}^s_{x:y} | s \in \mathcal{S}^n_{s_i}\} \tag{22}$$

Informally, the extended experiences set contains the experiences that customer/provider x possesses about the provider y delivering the service s, as well as experiences about y delivering any service s_i which similarity is of rank n (i.e., $Sim(s, s_i) \geq \frac{n}{10}$). Furthermore, we assume also that experiences are ranked in the set based on their proximity then their chronology. Once we have defined the extend experience set, the aggregation of theses experiences to compute the direct reputation ($\vec{R}^{s^n}_{C:p}$) of rank n and the indirect reputation ($\tilde{R}^{s^n}_{C:p}$) of rank n as defined in formula 4 and formula 8 is relatively straightforward. For the sake of concision, we will define only the direct reputation as the indirect one is derived in an analogous way.

$$R^{s^n}_{c:p} = \frac{\sum_{i=1}^{\lambda} \left[(e_i^{\langle c,p,s_i \rangle}).v \right]}{\lambda} \mid Sim(s, s_i) >= \frac{n}{10} \tag{23}$$

Where : $\forall e_i, e_j \in \mathcal{E}^{s^n}_{c:p}$
$i < j \implies (Sim(e_i.s, s) \geq Sim(e_j.s, s)) \wedge (e_i.t \geq e_j.t)$

In this new reputation metric, only the λ most similar and most recent experiences are used in computing the reputation of y towards x for providing the service s. We make the assumption that this approach of proceeding will guarantee that sufficient experiences are available to make an accurate reputation trust assessment of a candidate provider, while giving priority to the most similar and fresher experiences. Afterwards, the generalization of the formulas used to compute the direct and indirect trust of rank n is a straightforward. As, both processes rely on the newly computed reputation, the changes are automatically propagated to derive then new rank n values $\vec{R}^{s^n}_{m:p}$, $\tilde{R}^{s^n}_{\mathcal{P}:p}$, $\vec{T}^{s^n}_{m:m^s}$,$\vec{T}^{q^n}_{c:m}$, $\tilde{T}^{q^n}_{C:m}$, $\mathcal{T}^{q^n}_{C:m}$, $\ldots$, $\mathcal{T}^{q^n}_{\mathcal{P}:m}$. Consequently, we will refrain from rewriting each formula and we only present how the final trust is specified.

$$\mathcal{T}^{q^n}_{c:m} : (\alpha \times \mathcal{T}^{q^n}_{C:m}) + (|1 - \alpha| \times \mathcal{T}^{q^n}_{\mathcal{P}:m}) \tag{24}$$

Obviously, when specifying a rank $n = 10$, the trust computed is no more similarity-based and we get the same value as we would have obtained if we used the schema $\mathcal{T}^q_{c:m}$ defined in Formula 2.

In the next section, we describe the algorithm used by each customer to select the best candidates using the trust model we sketched in this section.

2.5 Trustworthy Multi-Clouds Selection

Algorithm 1: Selects the most trustworthy multi-cloud *MTMC* candidate

FindMTMC $(c, q, \mathcal{M}^q, \mathcal{E}, \aleph)$
- **inputs** : -A customer c from C
  -An active query q broadcasted by c
  -A set of candidate multi-clouds for query q
  -The set of all experiences shared *CMP*
  -Services similarity degree $\aleph \in [0, 10]$
- **output** : The most trustworthy multi-cloud denoted $\mathcal{M}^*$
- $\mathcal{M}^* \leftarrow \emptyset$;
- $HighestTrust \leftarrow 0$;
- $\mathcal{T} \leftarrow 0$;
- **foreach** *multi-cloud* $m_i \in \mathcal{M}^q$ **do**
  - $\mathcal{T} \leftarrow \mathcal{T}^{q^{\aleph}}_{c:m_i}$;
  - **if** $\mathcal{T} > HighestTrust$ **then**
    - $HighestTrust \leftarrow \mathcal{T}$;
    - $\mathcal{M}^* \leftarrow m_i$;
- **return** $\mathcal{M}^*$;

At this stage, we have introduced the necessary schema to compute the trust level of each multi-cloud. In this section, we present

[4] The description of the approach used to compute the similarity of two services is out of the scope of this paper. We refer the reader to models using Ontologies [13] or those using TF-IDF on SLA describing the cloud service [14]

briefly the Algorithm we use to find the most trustworthy multi-cloud candidate for a specific cloud query q. Once the query is issued, we assume that providers are grouped into multi-cloud and replies to the query q with an offer. When the offer satisfies the query requirements, the multi-cloud is added to the set $\mathcal{M}^q$ representing eligible candidates for query q. The objective of Algorithm **??** presented hereafter is to process each candidate and calculate it's trustworthiness degree with $\mathcal{T}^{q^{\aleph}}_{c:m_i}$. The similarity degree used to compute this trust is user-specific input parameter. With $\aleph$, the user decides the extent to which he wants that the experience used to compute trust matches the services requested in the query q (i.e., Similarity-Level). The impact of the parameter $\aleph$ on the efficiency of the computed trust degree will be evaluated in Section 3, as well as the overall approach.

3 EVALUATION

To evaluate the relevance and benefits of our trust model, we setup a simulation experiments using the Multi-Agent Simulation Platform Repast Symphony [17]. In this section, we fist present the simulation model. Then we provide details about the settings we used for this model. Finally, we describe the results we obtained with different settings.

3.1 Simulation Model

The simulation model consists of two separate groups of (software) agents representing Cloud customers (i.e., elements from C) and Cloud providers (.i.e., elements from $\mathcal{P}$). Parties share a common market place wherein providers advertise cloud service offers (i.e., elements from O) and customer propagate cloud service queries (i.e., elements from Q).

Simulation time is continuous and each interval $\{t, t + \delta t\}$ represent a round of simulation. Within each round, each provider reviews his active SSLA agreements (i.e., elements from $\mathcal{A}$) and decides what level of service he will deliver, for each service of each agreement. The compliance of a provider to an SSLA is function of a behavioral probability $\omega \in [0, 1]$.

Cloud customers can be in two states, (a) running services, (b) pending offers. When a customer is running his services in the system, he will make an assessment of the service level effectively delivered by the multi-cloud he contracted with. If no active contract is running, the customer will wait for candidates for each offer, then selects the most trustworthy multi-cloud after a fraction of time $\pi \cdot \delta t$. If the customer receives only one offer, a minimal trustworthiness threshold $\phi \in [0, 1]$ is used to exclude untrustworthy or not-sufficiently trustworthy candidates.

The propensity of interaction (i.e., issuing queries for customers and making offers for providers) is set using a probabilistic value $\chi \in [0, 1]$. We assume also that at each round, customers, (resp providers) can join/leave the cloud market place with a probability of ζ^C (resp. $\zeta^{\mathcal{P}}$), introducing a certain degree of dynamism in the simulation. When a customer/provider decides to leave the system, he waits until he finishes all his active contracts to make his departure effective.

Cloud services exchanged within the marketplace are represented with a numerical value τ to allow their comparison. Initially, the system contains 20 different types of services that are affected randomly to cloud providers. We also make use of a probability parameter $\zeta^{\mathcal{S}}$ to make the set of cloud services offered in the system evolve during the simulation.

3.2 Simulation Settings

In our simulation, we have created 1000 cloud providers, and 30 cloud customers with the particularity that none of the providers is able to fulfill any of the customers request alone, making the creation of multi-clouds a necessity for them.

The default ζ^C and $\zeta^{\mathcal{P}}$ parameters have been fixed to 0.05, which means that at each round of the simulation 5 customers and/or providers leave/join the system. Similarly, we set $\zeta^{\mathcal{S}}$ to 0.01 to make one new service appear at each round.

Moreover, the offers waiting duration was fixed to 10 rounds (i.e., $\pi = 10$), the interaction propensity χ was set to 0.8 and the trustworthiness minimal threshold ϕ to 0.5. All providers are also initialized with a default reputation value of 0.5 to avoid the initial providers' cold-start problem.

The compliance behavior parameters ω implies the existence of several types of providers' behavior. In our settings, we defined five types : *Very Good Behavior* (VGB), *Good Behavior* (GB), *Normal Behavior* (NB), *Bad Behavior* (BB) and Very Bad Behavior (VBB). VGB makes the provider perform as expected with almost a full compliance with the Quality of Service and Protection stated in the SLA. At the opposite, Providers showing a VBB are more prone to failure as most of their SLAs are not fulfilled. Each type of providers' behavior is defined in terms of mean and standard deviation parameters of a Gaussian distribution from which CSPs simulation behavior will be drawn (from the range [0,1]). These parameters are summarized the Table hereafter.

CSP profile	mean	StDev
VGP	0.95	0.05
GP	0.75	0.2
NP	0.50	0.05
BP	0.35	0.15
VBP	0.1	0.1

Table 1: Providers' profile settings

In this paper, we have made the hypothesis that the same provider can perform well for some specific services, while having very poor results for some others. Consequently, we make use of heterogeneous profiles such as the same provider is affected a random behavior type for each of the services he provides.

With respect to customers, the different parameters of our model implies the existence of several types of customers. As our objective is to compare the different features proposed by our approach, we retained eight different profiles that we summarizes in the following Table.

Cloud customers of type CDT (Customer Direct Trust) will serve as control population, they rely only on their own experience when making trust decisions. At the opposite, CIT (Customers Indirect Trust) relies on their experiences and the experience of other customers as well. The CRT (Multi-Clouds Reflexive Trust) refers to

	α	β	γ	$\aleph$	ζ^S
CDT	1	1	-	10	0.01
CIT	1	0.5	-	10	0.01
CRT	0	0.5	1	10	0.01
2DT	0.5	0.5	0.5	10	0.01
S2DT6	0.5	0.5	0.5	6	0.1
S2DT3	0.5	0.5	0.5	3	0.1
S2DT0	0.5	0.5	0.5	0	0.1
S2DT10	0.5	0.5	0.5	10	0.1

Table 2: Customers' profile settings

customers that consider using the multi-cloud providers experience with each other in their trust assessment. Two-Dimensions Trust (2DT) refers to customers that make use of all experiences available in the system. These customers believe that using the experience of other customers, as well as other providers (not necessarily those involved in the multi-cloud) will strength their trust decisions. S2DT refers to the category of customers that make use of services similarity based experiences in their trust assessment. By default, 2DT customers make use of similarity degree of 10 meaning that only experiences that concern the requested service enter in consideration when computing trust. To assess the benefit of our similarity based trust model, we make use of three additional customers type S2DT0, S2DT3 and S2DT6 that use, respectively, 0, 3 and 6 as similarity degrees. We also make use of S2DT10 as control population for our similarity-based model.

3.3 Results

The results we present in this section describe our experiments with the settings aforementioned. Each simulation consisted of 5 000 rounds, which we assume is enough to observe stable behaviors. As stated before, we make use of 1000 cloud providers, and 30 cloud customers. Also, the results of each customer are directly impacted by the different probabilities we use in our model. Thus, in order to minimize their effect on our results, and in order to explore the space of all possible outcomes as well, all results presented plot the mean of 20 executions of the same simulation settings.

The main metrics we refer to in our results analysis are the number of active contracts within the system (i.e., #Active Contracts) and the mean of the experiences reported after these contracts (i.e., Average Experiences Value). We believe that the first metric reflects the interaction enabling feature of our model and the second its satisfaction efficiency.

3.3.1 ***Direct Trust / Indirect Trust.*** We make use of this first series of experiments to build our ground truth. For that aim, we compare customers using only their direct experiences (i.e., CDT) with those using other customers experience (i.e., CIT) when making their decisions. As noted in Figure 3, the population using indirect trust reached and stabilized at a higher number of contracts more rapidly that those relying only on their individual observations. We notice however, that the excess of contracts seams to be slightly less qualitative.

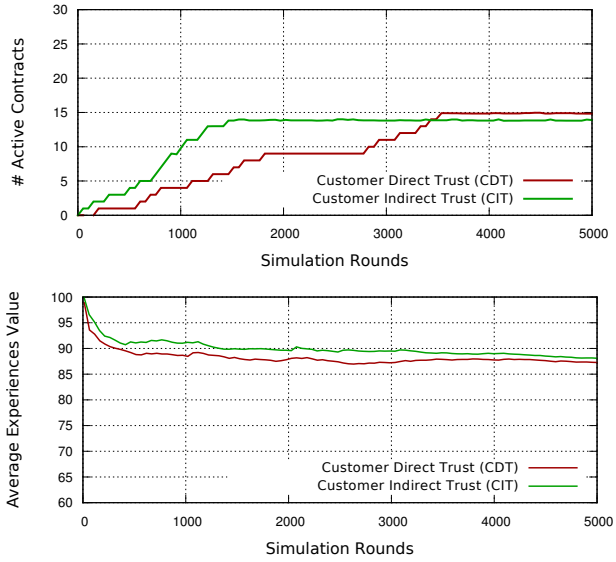

Figure 3: Direct Trust Vs Indirect Trust Results

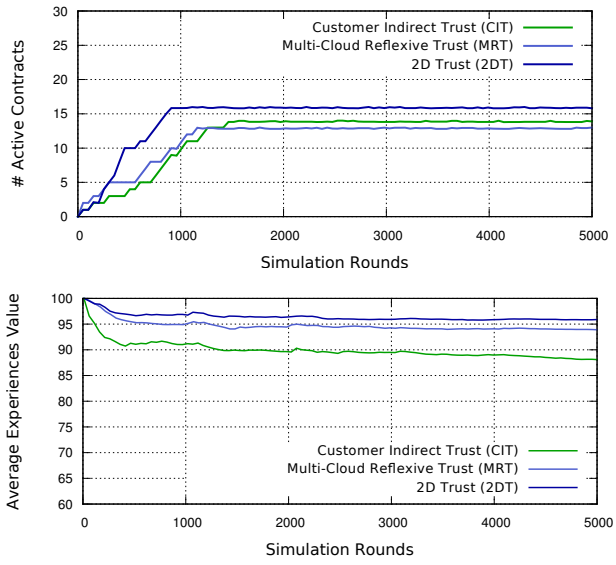

Figure 4: Multi-Clouds Reflexive Trust

3.3.2 ***Customers Trust with/without Providers Trust.*** Figure 4 shows the number of active contracts when using customers's profiles CIT, MRT and 2DT. The results show that introducing the experiences that providers have had with each other reduces the number of contracts in the system. As provider's experiences with each other are more accurate, the resulting reputation is more discriminant, which explains the reduced number of contracts. However, this loss of quantity in terms of contracts is compensated in terms of quality gain as the running contracts tend to reach a much higher level of satisfaction.

3.3.3 ***Effectiveness of Similarity-Based Trust.*** In order to observe the performance of our trust model when new cloud services are introduced, we changed cloud services' dynamics probability ζ^S from 0.01 to 0.1 so that 10 services will be added and/or removed at each round. Once the new service have been selected, providers can make offers with them.

The behavior of each provider with respect to this new service is computed as follows. We first derive the similarity between this service and all the services proposed by the same provider. We identify the most similar best service (i.e., service associated to a good behavior such as $\omega \geq 0.5$) and the most similar worst service (i.e., service associated to a bad behavior such as $\omega < 0.5$). Then the new service behavior probability ω is derived using the mean of ω from the two services :

$$\omega_{NewService} = \frac{\omega_{BestSimilarService} + \omega_{WorstSimilarService}}{2} \tag{25}$$

As a result (cf., Figure 5), customers with profile $S2DT10$ was unable to make trust assessement, making the number of active contract collapse compared nominal settings $2DT$. The other profiles S2DT6, S2DT3 and S2DT0 succeed maintaining a reasonable amount of contracts. However, the broad experience set used in S2DT3 and S2DT0 due to the weak similarity level used caused loss of SLA fulfillment level.

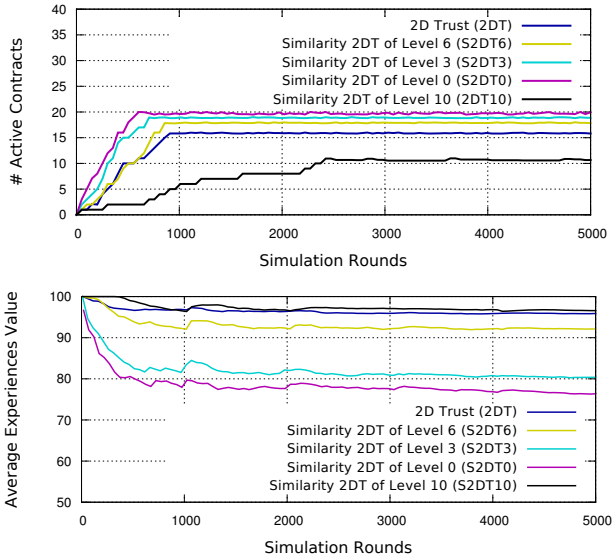

Figure 5: Similarity-Based Trust Results

As the trustworthiness of providers is specific to the services in which they show good behavior, this trustworthiness could not be necessarily transferred to services that are too dissimilar. From these results it appears that a similarity level of 60% reveals to be a good balance between safety and efficiency in the system.

4 RELATED WORKS

Trust is a well established mechanism that supports cooperation and collaboration in open and decentralized systems [8, 20]. Consequently, this concept has been extensively investigated in the last fifteen years resulting in the emergence of numerous trust models [10]. Each model tried to represent the information a participant should collect about the others and proposed an evaluation schema to derive trust from that information [20]. The objective of this section is not to review the complete literature on this subject, there exist several surveys that provided more comprehensive studies (c.f.[1], [11],[4], [2],[20]). Instead, we describe representative approaches and discuss their relevance to multi-cloud computing.

In [7], the authors proposed a trust management framework for multi-clouds in which they combined *objective* (i.e., SLA-Based) and *subjecive* (Reputation-Based) approaches. However, their solution relies on *Trust Service Providers*, which are independent third-party trust brokers that need to be trusted by both Cloud Providers, Cloud Service Providers and Cloud Services Users.

In [16], the authors proposed the "Trust as a Servics" (TaaS) framework in which the authors introduced and *adaptive credibility model* that distinguishes between credible trust feedbacks and malicious feedback by considering cloud service consumers' capability and majority consensus of their feedbacks. In this work, trust has been addressed only from users' perspective.

In [9], the authors proposed a multi-facets model to manage trust in Cloud Computing marketplaces. Like in our approach, their model collects several attributes to assess the trustworthiness of a Cloud Service Provider. These attributes correspond to Service Level Objectives defined within active SLAs. Feedback information is also collected from different sources and used alongside SLA metrics to derive a trust score for each CSP. The authors make use of the Consensus Assessments Initiative Questionnaire. CAIQ is maintained and shared by the Cloud Cloud Security Alliance and the authors use it as a way to extract SLA compliance information [9] .

In the Joint Risk and Trust Model (JRTM) developed in the context of the A4Cloud (Accountability for Cloud) [6]. In this approach, statistical data collected from third party services (i.e., a Trust as a Service Provider) are accumulated and computed to estimate the trust that a Cloud Customer puts on a specific Cloud Service Provider. The model relies on the assessment of the cloud service security and privacy risk to derive a trust metric. The information used include statistics on the number of security and privacy incidents that the CSP was subject to.

In [18] the authors defined a similarity based prediction model. Entities (i.e., cloud users and cloud providers) are represented using a vector of capabilities and interests. The more theses vectors are similar the more likely trust can be established between them.

In [9] the authors presented a behavior-based trust model in which the trust value depends on the expected behavior of the cloud provider. The behavior of the provider is assessed with respect to specific attributes such as security measures, compliance and customer support. Here again, the authors focus on the perspective of the user that tries to select the best provider.

Although the related work testifies to a steady progress in the management of trust, to the best of our knowledge, no approach

tried to address the issues of multi-clouds (group-reputation) and cold-start due to services offers evolution and its impact on the management of trust.

5 CONCLUSION

We presented, in this paper, a novel trust model for multi-cloud systems in which trust assessment is made more complex as limited to no prior experiences are found in the system. This situation is known in the literature as the cold-start problem. We were particularly interested in situations wherein the cold-start is due to a highly dynamic environment wherein new services are introduced at a very high pace. The complexity of these systems is also due to the virtualization infrastructure that prevents cloud customers from identifying the responsible of service outage.

The contributions reported in this article addressed these problems in three steps. First, we have made the hypothesis that cloud providers should have service specific trust assessment and not global reputations as it is done in the literature. Second, we propose to make the experiences the providers have with each other accessible to customers so they could make their their trust assessment more accurate. Finally, proposed a similarity-based trust assessment to enrich the experience base of customers (and providers). We assume that this last feature will enable collaboration, minimizing the cold-start impact on the system.

The results that have been obtained show that using similarity-based approach trust assessment allow cloud customers overcoming lack of feedback when new cloud services are proposed in the market-place. Our model stimulates interaction in absence of experience while classical approaches tend to demonstrate a strong dependence on the abundance of customers feedback.

The results show also that the risk of extending the experience base to similar service is beneficial up to 60 % similarity level. Below, we observe a clear degradation of the overall SLA fulfillment level. In future works, more complex settings need to be studied in order to tweak the similarity value to the best we can obtain.

The results we have obtained show that similarity-based trust mechanisms can clearly help cloud customers make accurate and safe trust decisions in situations where no direct or indirect prior experience exists. We therefore conclude that our Similarity-Based approach can assist positively cloud customers in their decision making process.

ACKNOWLEDGMENTS

This research was partially funded by the European Commission through the Horizon 2020 project SUPERCLOUD under grant agreement 643964. The views and conclusions contained in this document are those of the author(s) and should not be interpreted as representing the official policies, either expressed or implied, of the European Commission.

REFERENCES

[1] Alfarez Abdul-rahman and Stephen Hailes. 1997. A Distributed Trust Model. *Computer* (1997).

[2] D. Artz and Y. Gil. 2010. A survey of trust in computer science and the Semantic Web. *Web Semantics: Science, Services and Agents on the World Wide Web* 5, 2 (June 2010), 58–71. DOI: http://dx.doi.org/10.1016/j.websem.2007.03.002

[3] B. Baranski, T. Bartz-Beielstein, R. Ehlers, T. Kajendran, B. Kosslers, J. Mehnen, T. Polaszek, R. Reimholz, J. Schmidt, K. Schmitt, D. Seis, R. Slodzinski, S. Steeg, N. Wiemann, and M. Zimmermann. 2006. The Impact of Group Reputation in Multiagent Environments. In *2006 IEEE International Conference on Evolutionary Computation*. 1224–1231. DOI: http://dx.doi.org/10.1109/CEC.2006.1688449

[4] Matt Blaze, Joan Feigenbaum, and AngelosD. Keromytis. 1999. KeyNote: Trust Management for Public-Key Infrastructures. In *Security Protocols*, Bruce Christianson, Bruno Crispo, WilliamS. Harbison, and Michael Roe (Eds.). Lecture Notes in Computer Science, Vol. 1550. Springer Berlin Heidelberg, 59–63. DOI: http://dx.doi.org/10.1007/3-540-49135-X_9

[5] Chris Burnett, Timothy J. Norman, and Katia Sycara. 2013. Stereotypical trust and bias in dynamic multiagent systems. *ACM Trans. Intell. Syst. Technol.* 4, 2, Article 26 (April 2013), 22 pages. DOI: http://dx.doi.org/10.1145/2438653.2438661

[6] E. Cayirci. 2013. A joint trust and risk model for MSaaS mashups. In *Simulation Conference (WSC), 2013 Winter*. 1347–1358. DOI: http://dx.doi.org/10.1109/WSC.2013.6721521

[7] Wenjuan Fan and Harry Perros. 2014. A novel trust management framework for multi-cloud environments based on trust service providers. *Knowledge-Based Systems* 70 (2014), 392–406. DOI: http://dx.doi.org/10.1016/j.knosys.2014.07.018

[8] S. Gupta and H.W. Kim. 2004. Virtual community: concepts, implications, and future research directions. In *Proceedings of the 10th American Conference on Information System*. 2679–2687.

[9] S.M. Habib, S. Ries, and M. Muhlhauser. 2011. Towards a Trust Management System for Cloud Computing. In *Trust, Security and Privacy in Computing and Communications (TrustCom), 2011 IEEE 10th International Conference on*. 933–939. DOI: http://dx.doi.org/10.1109/TrustCom.2011.129

[10] Audun Jøsang. 2007. Trust and reputation systems. In *Foundations of security analysis and design IV*, Alessandro Aldini and Roberto Gorrieri (Eds.). Springer-Verlag, Berlin, Heidelberg, 209–245. http://dl.acm.org/citation.cfm?id=1793914.1793923

[11] A. Jøsang, R. Ismail, and C. Boyd. 2007. A survey of trust and reputation systems for online service provision. *Decision Support Systems* 43, 2 (2007), 618–644. DOI: http://dx.doi.org/10.1016/j.dss.2005.05.019

[12] I. Kaliszewski and D. Podkopaev. 2016. Simple Additive weighting-A Metamodel for Multiple Criteria Decision Analysis Methods. *Expert Syst. Appl.* 54, C (July 2016), 155–161. DOI: http://dx.doi.org/10.1016/j.eswa.2016.01.042

[13] Jaeyong Kang and Kwang Mong Sim. 2011. Cloudle: An Ontology-Enhanced Cloud Service Search Engine. (2011), 416–427. DOI: http://dx.doi.org/10.1007/978-3-642-24396-7_33

[14] Natallia Kokash. 2006. A Comparison of Web Service Interface Similarity Measures. (2006), 220–231. http://dl.acm.org/citation.cfm?id=1565192.1565213

[15] M. Lacoste, M. Miettinen, N. Neves, F. M. V. Ramos, M. Vukolic, F. Charmet, R. Yaich, K. Oborzynski, G. Vernekar, and P. Sousa. 2016. User-Centric Security and Dependability in the Clouds-of-Clouds. *IEEE Cloud Computing* 3, 5 (Sept 2016), 64–75. DOI: http://dx.doi.org/10.1109/MCC.2016.110

[16] Talal H. Noor and Quan Z. Sheng. 2011. Trust As a Service: A Framework for Trust Management in Cloud Environments. In *Proceedings of the 12th International Conference on Web Information System Engineering (WISE'11)*. Springer-Verlag, Berlin, Heidelberg, 314–321. http://dl.acm.org/citation.cfm?id=2050963.2050992

[17] E. Tatara, M. J. North, T. R. Howe, N. T. Collier, and J. R. Vos. 2005. An Introduction to Repast Simphony Modeling using a Simple Predator-Prey Example. In *Proceedings of the Agent 2005 Conference on Generative Social Processes, Models and Mechanisms*, C. M. Macal, M. J. North, and D. Sallach (Eds.).

[18] F. Xie, Z. Chen, H. Xu, X. Feng, and Q. Hou. 2013. TST: Threshold based similarity transitivity method in collaborative filtering with cloud computing. *Tsinghua Science and Technology* 18, 3 (June 2013), 318–327. DOI: http://dx.doi.org/10.1109/TST.2013.6522590

[19] Reda Yaich, Olivier Boissier, Gauthier Picard, and Philippe Jaillon. 2012. An Adaptive and Socially-Compliant Trust Management System for Virtual Communities. In *The 27th ACM Symposium On Applied Computing (SAC 2012)*. ACM Press. http://www.acm.org/conferences/sac/sac2012/

[20] Reda Yaich, Olivier Boissier, Gauthier Picard, and Philippe Jaillon. 2013. Adaptiveness and Social-Compliance in Trust Management within Virtual Communities. *Web Intelligence and Agent Systems (WIAS), Special Issue: Web Intelligence and Communities* 11, 4 (2013).

[21] Reda Yaich, Sabir Idrees, Nora Cuppens, Frédéric Cuppens, Marc Lacoste, Nizar Kheir, Ruan He, Khalifa Toumi, Krzysztof Oborzynski, Meilof Veeningen, and Paulo Sousa. 2015. D1.2 - SUPERCLOUD Self-Management of Security Specification. *SUPERCLOUD* (2015). https://supercloud-project.eu/downloads/SC-D1.2-Self-Management_Security_Specification-PU-M09.pdf

[22] Giorgos Zacharia, Alexandros Moukas, and Pattie Maes. 2000. Collaborative reputation mechanisms for electronic marketplaces. *Decision Support Systems* 29, 4 (2000), 371–388. DOI: http://dx.doi.org/10.1016/S0167-9236(00)00084-1