Journal of Theoretical and Applied Electronic Commerce Research
ISSN 0718–1876 Electronic Version
VOL 5 / ISSUE 2 / AUGUST 2010 / 77-96

This paper is available online at
www.jtaer.com
DOI: 10.4067/S0718-18762010000200006

A Flexible Architecture for Privacy-Aware Trust Management

Klemens Böhm[1], Sandro Etalle[2], Jerry den Hartog[3], Christian Hütter[4], Slim Trabelsi[5], Daniel Trivellato[6], and Nicola Zannone[7]

Karlsruhe Institute of Technology, Germany, [1] klemens.boehm@kit.edu, [4] christian.huetter@kit.edu
[2] University of Twente, Enschede, The Netherlands, sandro.etalle@utwente.nl
Eindhoven University of Technology, The Netherlands, [2] s.etalle@tue.nl, [3] j.d.hartog@tue.nl,
[6] d.trivellato@tue.nl, [7] n.zannone@tue.nl
[5] SAP Labs France, Mougins Cedex, France, slim.trabelsi@sap.com

Received 15 February 2010; received in revised form 10 June 2010; accepted 12 June 2010

Abstract

In service-oriented systems a constellation of services cooperate, sharing potentially sensitive information and responsibilities. Cooperation is only possible if the different participants trust each other. As trust may depend on many different factors, in a flexible framework for Trust Management (TM) trust must be computed by combining different types of information. In this paper we describe the TAS^3 TM framework which integrates independent TM systems into a single trust decision point. The TM framework supports intricate combinations whilst still remaining easily extensible. It also provides a unified trust evaluation interface to the (authorization framework of the) services. We demonstrate the flexibility of the approach by integrating three distinct TM paradigms: reputation-based TM, credential-based TM, and Key Performance Indicator TM. Finally, we discuss privacy concerns in TM systems and the directions to be taken for the definition of a privacy-friendly TM architecture.

Key words: Trust Management, Security Framework, Reputation-based Trust Management, Credential-based Trust Management, Key Performance Indicator Trust Management

Journal of Theoretical and Applied Electronic Commerce Research
ISSN 0718–1876 Electronic Version
VOL 5 / ISSUE 2 / AUGUST 2010 / 77-96

This paper is available online at
www.jtaer.com
DOI: 10.4067/S0718-18762010000200006

1 Introduction

In service-oriented systems a constellation of services cooperate, sharing potentially sensitive information and responsibilities. For example, service-oriented systems in the employability setting rely heavily on personally identifiable information (PII); e.g. a job seeker's CV needs to be matched with open positions, a worker's career plans need to be linked with suitable trainings, etc. Traditional access control mechanisms are centralized and operate under the assumption that all principals are known by the system. This assumption, however, is not applicable to distributed systems where principals do not know each other a priori.

Sharing PII in distributed systems is only possible if the different participants trust each other: the end users need to trust the services providers, but also the service providers have to trust each other. As trust may depend on many different factors, in a flexible framework for Trust Management (TM) trust must be computed by combining different types of information. The aim of the Trusted Architecture for Securely Shared Services (TAS^3) project is to build an architecture which enables sharing PII among services in a secure and trustworthy manner. This paper describes the TM framework developed within the TAS^3 project.

In TM systems (e.g., [5], [4], [29], [19], [13], [2]), decisions are taken based on statements made by multiple principals. The decision about who can be trusted (e.g., to access a resource) is taken not just by a single principal but by taking into account information from other principals. In this way the decision is, at least partially, delegated to other principals. The form of delegation depends on the relationship with the other principals and the type of trust information that is made available to her.

Existing TM systems compute trust from specific types of trust information. Credential-based TM systems [5], [4], [29] is an approach to distributed access control in which access decisions are based on credentials issued by multiple principals and stored in distributed manner. The credential-based TM service automates the process of determining whether a principal has the necessary credentials to access the requested resources. Using credential-based TM, an employability provider may trust that the job seeker has a MSc degree based on credentials issued by a university. In reputation-based TM (see [13] for an overview), the decision whether to trust a service provider or not depends on the reputation of that provider. The RTM service aggregates feedback given by users into reputation values. Users can define trust policies which refer to the reputation values in order to identify trustworthy service providers. Using reputation-based TM, a job seeker may trust an employability provider based on good feedback from previous clients. Key Performance Indicators (KPI) are financial or non-financial measures or metrics used to measure progress, performance or goals of an organization and provide another source of trust feedback. KPI-based TM is an approach in which trust evaluation is based on the experience on KPIs shared by different users. Using KPI-based TM, an employer may trust that a candidate is suitable to perform certain duties on the basis of their performance in previous working experience.

The approaches mentioned above typically focus on a single type of trust information. Several approaches (e.g., [6], [12], [18]) combine multiple sources of trust by incorporating an additional source in an existing system, using a single policy language to process both types of trust information. In [6], a predicate expressing a reputation within a given range is added to a logic-based system. The hybrid trust model of [12] combines experience in past interactions, recommendations and the credentials of the parties involved using the LTL logic language of the reputation system of [16]. In [18], the credential-based TM system RT [19] is extended with a flexible operator to capture behavioral style trust requirements. In this way also arbitrary nesting of credentials and behavioral trust requirements is possible. In [35], the KeyNote [5] system is complemented with an independent reputation management system and a 'decision maker' to combine their results. Our approach also supports arbitrary nesting and uses an overall structure similar to [35], but applied to an arbitrary and extensible combination of TM systems.

The main contribution of this paper is the introduction of a TM framework that integrates independent TM systems into a single framework with a unified interface that fits well with the service-oriented systems. A novel aspect is that the framework is easily extensible yet supports intricate combinations of trust services such as nesting. We show the flexibility of the approach by a prototype implementation of the framework integrating three distinct TM paradigms: reputation-based TM, credential-based TM, and dynamic Key Performance Indicator TM. We demonstrate the effectiveness of our approach within an employability scenario. In the employability setting, sensitive data, such as an ePortfolio, needs to be shared, for example when searching for a new job. Our

Journal of Theoretical and Applied Electronic Commerce Research
ISSN 0718–1876 Electronic Version
VOL 5 / ISSUE 2 / AUGUST 2010 / 77-96

This paper is available online at
www.jtaer.com
DOI: 10.4067/S0718-18762010000200006

use case shows how trust can be established and how it changes dynamically based on updates to the trust information. Since the data that is used to establish trust may itself be sensitive, we also discuss the privacy requirements and implications of the trust framework itself.

The remainder of this paper is structured as follows. Section 2 introduces an application scenario in the employability domain, showing different trust needs. Section 3 discusses different TM systems and how they contribute to building trust from the different sources of information available. Section 4 presents how different TM systems can be integrated into a single TM framework. Section 5 discusses the trust and privacy implication of the TM framework itself. Finally, Section 6 provides conclusions and future work.

2 Running Example: An Employability Case Study

This section introduces a case study in the employability domain that has been studied in the TAS^3 project. More precisely, we look at a scenario for student placement being run by the University of Nottingham (UoN). Our focus is on the trust requirements imposed by this scenario. Based on these requirements the next sections will discuss how individual trust management systems can contribute and how they can be combined.

Student employability is becoming a critical issue for Higher Education in the UK and elsewhere: graduates want to see a return on their educational (and financial) investment and are seeking ways of improving their currency in the pressured market for jobs. In the UK this is leading to an increased number of funding schemes, some associated directly with courses. Many UK universities are employing placement providers, either internally or in outsourcing, to widen and facilitate access by their students to such funding schemes.

To support the expanding demands of student placements, the process should be automated as much as possible. Central to this development is the exchange of sensitive and private data between all parties: universities, students, placement providers and companies offering employment. As a consequence, managing trust and privacy settings between all users of the system will increase in complexity.

Efficient flow and exchange of information about the students themselves, the programmes they are eligible for and the vacancies available is needed to support the matching of students with appropriate placements. There are elements of choice at both ends of the process: students want to be able to choose from a selection of suitable placements, while employers wish to choose from a selection of suitable candidates. Preservation of anonymity and gradual release of data, both from the students about themselves and the placement provider (employer), help to ensure fairness and impartiality throughout the process.

Below we present the steps in the scenario relevant to TM defined by UoN [24].

- A student attending a university course wants to submit a job application. This can be either to fulfill a particular requirement which is mandatory for the course, or a voluntary decision to improve the student's employability prospects once the course is completed.
- The student identifies a placement provider authorized by the university and registers with this provider. At this point a basic contractual agreement is established.
- The placement provider verifies the students's identity, and checks that she is eligible to participate in one or more of the funding schemes administered by the provider.
- The placement provider tells the student which funding schemes she is eligible for.
- The student chooses which funding scheme(s) she is interested in and completes a specific application form, including access to personal information in the form of a CV or ePortfolio.
- The student's data is used to match her profile with the vacancies included in the funding scheme. This matching process can be performed in house by the placement provider or by an external matching service.
- The matches are collated by the placement provider and presented to the student, which chooses those she is interested in. Further 'soft' matching then takes place, including a face-to-face interview.

Journal of Theoretical and Applied Electronic Commerce Research
ISSN 0718–1876 Electronic Version
VOL 5 / ISSUE 2 / AUGUST 2010 / 77-96

This paper is available online at
www.jtaer.com
DOI: 10.4067/S0718-18762010000200006

Alice	A_1	I let someone access my data if the average feedback about him is positive.
Bob	B_1	I want to request service X from someone who has got no negative feedback within the last 24h.
Carol	C_1	I will only interact with someone if the k most reputable entities recommend him.
Dave	D_1	I only request services from others if their performance regarding complex tasks was satisfactory.
Eve	E_1	I only use services with above average feedback from fellow MSc in Business Administration holders.
UoN	U_1	Placement provider shall present appropriate certificates before being allowed to access data about the student (e.g. to check eligibility of the student).
Placement provider	PP_1	Only applicants with a MSc in Business Administration issued by an accredited university can apply to managerial positions.

Table 1: Samples of Trust Policies

- If the student is accepted for a specific placement, she signs contracts with the placement provider agreeing to the terms for that placement.
- If no suitable placement is found, the student can repeat the process.
- After the process is completed, the student has the opportunity to give feedback on the service by rating different aspects of the service, such as speed or quality.

PII plays a key role in this scenario, as students' personal data needs to be exchanged among different services; e.g. to match a student to a position profile a matching service will need access to (a large portion) of the ePortfolio. In Table 1, we present an excerpt of the security policies governing our scenario. These policies show that there are significant differences in the way how the trustworthiness of stakeholders can be derived. For instance, the placement provider has to present a number of credentials to prove its trustworthiness to the university, whereas students can base their decision on past performance of the service (e.g. experiences of other students).

Different TM paradigms exist to address these different types of trust requirements. Below we discuss three TM approaches and show how they apply to this scenario. Then, we discuss how to integrate them into a single framework.

3 Trust Services

To meet the different trust requirements identified in the scenario of Section 2, we have selected, enhanced and implemented three TM paradigms: reputation-based TM, credential-based TM, and Key Performance Indicator TM. We treat the respective implementations as services, referring to them as *trust services*.

3.1 Reputation-based Trust Management

In reputation-based trust management (RTM), the decision whether to trust a service provider or not depends on the reputation of that provider. The RTM service aggregates feedback given by users into reputation values. Users can define trust policies which refer to the reputation values in order to identify trustworthy service providers.

3.1.1 Reputation-based Trust Model

A basic concept of RTM is that users rate service providers. When a user interacts with a service provider, he gives feedback on the interaction afterwards. The RTM service computes the reputation of a service provider based on the feedback. Users can then refer to the resulting reputation value to determine the privileges of

Journal of Theoretical and Applied Electronic Commerce Research
ISSN 0718–1876 Electronic Version
VOL 5 / ISSUE 2 / AUGUST 2010 / 77-96

This paper is available online at
www.jtaer.com
DOI: 10.4067/S0718-18762010000200006

that service provider. In the following, we call the instructions on how to derive the trustworthiness of service providers a *trust policy*.

We call a subject giving feedback the *rater* and the object on which feedback is given the *ratee*. Feedback data has several aspects, most importantly the feedback value $v \in [-1,1]$. The context c allows to distinguish between different types of services, while the facet f_c captures a specific perspective of context c. The timestamp t allows to emphasize the impact of current feedback. The certainty $\sigma \in [0,1]$ captures the rater's confidence in the rating, while the effort $\varepsilon \in [0,1]$ quantifies the perceived complexity of the service.

A relational representation of feedback data allows for a straightforward implementation of the RTM service based on standard database technology. We use the following relations to store the feedback data:

- `Feedback(rater,ratee,value,context,facet,time,certainty,effort)` contains feedback data as described above.
- `Entity(id,name,address,...)` contains the unique identifiers of the users and some additional data.
- `Situation(context,facet)` captures the facets possible in the different contexts.

Example 1 (Relational representation of feedback data) *The feedback of Alice "I am quite sure (.75) that the quality of service S by Bob was good (.95). It was a complex (.8) problem." is represented by the following feedback tuple:*

Rater	Ratee	Value	Context	Facet	Time	Certainty	Effort
Alice	Bob	0.95	S	Quality	12:09:45	0.75	0.8

Feedback data can be interpreted as a weighted, directed graph $G(V,E)$ with vertices in set V representing users, edges in set E representing feedback, and edge weights $w(e)$ giving the feedback value for edge $e \in E$.

3.1.2 Reputation-based Trust Policy Language

Since trust is a highly subjective topic, each user has individual policies on how to derive trust from feedback. Therefore, a flexible language is required to specify when a particular service provider is trusted. Notice that trust policies may require complex operations as well as a complete history of feedback data which is not available to isolated users.

We propose a language for the formulation of reputation-based trust policies based on *Relational Algebra* (RA). RA defines a set of operators to be applied on relations, which in turn are closed under these operators. In addition, the language employs a *centrality operator* that is described below. A trust policy is thus an algebra expression over the relational representation of feedback data.

The usage of centrality measures has been proposed to determine the reputation of users based on feedback data [14], [39]. Centrality measures are a special kind of graph algorithms which quantify the relative importance of nodes according to the graph structure. They compute a numerical value for each vertex, the *centrality score*, which allows for a ranking of the vertices. The intuition is that a service provider with a high centrality score is considered as trustworthy.

In the literature, a large number of different centrality measures has been proposed: HITS [15], Positional Power [11], PageRank [28], Proximity Prestige [22], and Integration & Radiality [34]. Most of these measures consider different parameters of the feedback graph, e.g., in/out degree of nodes, shortest paths, etc. This means that different measures are likely to yield different rankings for the same graph. In order to support a broad range of centrality measures and graph transformation techniques, the envisioned trust policy language must be extensible.

Until now no database management systems directly support the computation of centrality measures. In order to compute centrality measures, the feedback graph has to be constructed outside the database such that the algorithms can be computed there. However, this approach is inflexible because of the pre- and post-processing

Journal of Theoretical and Applied Electronic Commerce Research
ISSN 0718–1876 Electronic Version
VOL 5 / ISSUE 2 / AUGUST 2010 / 77-96

This paper is available online at
www.jtaer.com
DOI: 10.4067/S0718-18762010000200006

of feedback data and reputation values. For example, a user might decide to consider only feedback from certain users (i.e., pre-process the feedback data) or to trust only the k most reputable service providers (i.e., post-process the reputation values).

Our approach is to define a new operator, the *centrality operator*, which allows the computation of centrality measures inside the trust database. This allows for a seamless integration in existing query processing as well as flexible pre/post processing of the data. The centrality operator has two design targets:

1. *Support of various centrality measures.* The desired measure is passed as parameter to the centrality operator such that the operator can easily be extended with other measures.
2. *Flexible specification of the graph structure.* Make the representation of the graph explicit by passing a list of attributes which specify the source, destination, and weight of the edges.

Definition 1 (Centrality operator) *The centrality operator is defined as follows:*

$$\mathit{CENTRALITY}[\mathtt{name},\mathtt{A_v},\mathtt{A_s},\mathtt{A_d},\mathtt{A_w},\ \mathtt{measure}](\mathtt{R_{vertices}},\mathtt{R_{edges}})$$

The attributes of the centrality operator have the following meaning: `name` *specifies how the result column in the output relation is called.* $\mathtt{A_v}$ *specifies the column of the relation* $\mathtt{R_{vertices}}$ *that contains the IDs of the vertices.* $\mathtt{A_s}$, $\mathtt{A_t}$, *and* $\mathtt{A_w}$ *specify the columns of the relation* $\mathtt{R_{edges}}$ *that contain the source and destination vertices of the edges as well as the edge weights. Finally,* `measure` *specifies the centrality measure to be used.*

Note that $\mathtt{R_{vertices}}$ and $\mathtt{R_{edges}}$ can be arbitrary relations, e.g., they can be results from other operations. This flexible definition of the graph structure allows for a powerful pre-processing of the input data. Because of the closure property of the relational algebra, the result of the centrality operator is a relation as well. The resulting relation can be used as input into the next expression, allowing for a post-processing of the output data. The output of the centrality operator is a relation consisting of two attributes Vertex and Name. That is, each tuple of the output relation consists of a vertex identifier (e.g., name of the user) and the score obtained for the vertex according to the centrality measure specified.

Example 2 (Reputation-based trust policy) *A policy of the form "I trust service providers if their average feedback value in context c and facet f_c from the 10 most reputable users exceeds threshold t. Use the* `PageRank` *centrality measure to rank the users." combines the policies of Carol C_1 and Dave D_1 (Table 1). It can be expressed as follows:*

$$\begin{array}{l}
\textbf{PROJECTION}[ratee](\\
\quad \textbf{SELECTION}[avg_value > t](\\
\quad\quad \textbf{GROUP}[avg_value, AVG(value), \{ratee\}](\\
\quad\quad\quad \textbf{JOIN}[rater = id](\\
\quad\quad\quad\quad \textbf{TOP}[10, pagerank](\\
\quad\quad\quad\quad\quad \textbf{CENTRALITY}[pagerank, id, rater, ratee, value, PageRank](\\
\quad\quad\quad\quad\quad\quad Entity, \\
\quad\quad\quad\quad\quad\quad \textbf{SELECTION}[context = c, facet = f_c](Feedback))), \\
\quad\quad\quad\quad \textbf{SELECTION}[context = c, facet = f_c](Feedback) \\
))));
\end{array}$$

The centrality computation is very time-consuming and resource-intensive because of the algorithmic complexity of centrality measures. Thus, the centrality computation is the most costly part of the evaluation of a trust policy. Various optimizations have been proposed to improve the computation of centrality measures (see [17] for an overview).

Journal of Theoretical and Applied Electronic Commerce Research
ISSN 0718–1876 Electronic Version
VOL 5 / ISSUE 2 / AUGUST 2010 / 77-96

This paper is available online at
www.jtaer.com
DOI: 10.4067/S0718-18762010000200006

3.2 Credential-based Trust Management

Credential-based trust management (CTM) is an approach to distributed access control in which access decisions are based on credentials issued by multiple principals and stored in distributed manner. The CTM service automates the process of determining whether a principal has the necessary credentials to access the requested resources.

3.2.1 Credential-based Trust Model

In CTM [3], [5], [9], [19], [25], [32], access decisions are based on credentials. Credentials are digital certificates attesting that a certain subject has a certain attribute, and are digitally signed by the certificate issuer to ensure their authenticity and integrity. Every principal has the authority to define credentials; a statement defining a credential (or the conditions in which it is issued) is called an *assertion*. The set of assertions made by a principal is the *trust policy* of that principal. The key aspect in CTM is delegation of authority: a principal may transfer authority over some attribute to other principals by referring to their credentials in its trust policy.

A main challenge in distributed systems is represented by the heterogeneity of its components: reaching a semantic agreement on the vocabulary of parties is necessary to enable mutual understanding, and hence interoperability.

Example 3 *According to policy PP_1 in Table 1, the placement provider, ePlace, requires the applicants for managerial positions to have a* MSc in Business Administration*. Suppose that a student, Alice, submits an application for a managerial position and attaches to it a credential issued by the UoN attesting that she has a* MSc in Business Management*. Despite the fact that the degree of the applicant clearly fits the profile, ePlace would not accept Alice's application. Indeed, the certificate presented by Alice is different from the one required by ePlace.*

Most existing CTM systems implicitly assume a common vocabulary shared by all principals for the specification of credentials [3], [19], [25]. This assumption is however too restrictive for short-term or dynamic collaborations, where reaching a complete and precise semantic agreement on the vocabulary would be too costly or take too much time.

We address the problem of semantic agreement in two steps: (1) we combine CTM with ontologies to give a precise semantics to trust policies, and (2) we employ the notion of similarity for semantic alignment. Ontologies are increasingly seen as a means for enabling interoperability in heterogeneous systems [8]: augmenting CTM with ontologies ensures mutual understanding among principals [32]. Similarity expresses the degree of semantic resemblance between two ontology concepts [10]. Combining ontologies with similarity allows principals to specify trust policies in terms of their local vocabulary, increasing their autonomy, while guaranteeing interoperability [33].

3.2.2 Credential-based Trust Policy Language

We present a policy specification language based on constraint Datalog, which has been proposed as a foundation for CTM languages [20]. A *trust policy* consists of a set of Horn clauses of the form $H \leftarrow B_1, \ldots, B_n, c$, where H is an atom called *head*, and $B_1, \ldots, B_n, c$ (with $n \geq 0$) is called *body*, where $B_1, \ldots, B_n$ are atoms, and c is a constraint. Trust policies are specified using three types of constructs:

- *Ontology atoms*: are used to query the knowledge base represented by ontologies; they have the form $conceptURI(a)$ or $relationshipURI(a_1, a_2)$, where $conceptURI$ and $relationshipURI$ identify a concept and a relationship in the ontology, respectively. $conceptURI(a)$ holds if a is an instance of $conceptURI$. $relationshipURI(a_1, a_2)$ holds if instance a_1 is related to instance a_2 via $relationshipURI$. Following the XML convention, we use a prefix to denote namespaces (i.e., URIs).
- *Credential atoms*: have the form $\mathsf{cred}(issuer, attribute, subject)$ and represent digitally signed certificates released by an $issuer$ attesting an $attribute$ of a $subject$. $issuer$ and $subject$ are identified by a unique name (e.g., a public key) and $attribute$ is a $conceptURI$.

Journal of Theoretical and Applied Electronic Commerce Research
ISSN 0718–1876 Electronic Version
VOL 5 / ISSUE 2 / AUGUST 2010 / 77-96

This paper is available online at
www.jtaer.com
DOI: 10.4067/S0718-18762010000200006

1	$\mathsf{cred}(ePlace, ep{:}validCandidate, X) \leftarrow ep{:}accredited(Y), \mathsf{cred}(Y, ep{:}MScBA, X)$
2	$\mathsf{cred}(UoN, uon{:}MScBM, Alice)$
3	$\mathsf{sim}(ePlace, ep{:}MScBA, uon{:}MScBM, 0.9)$
4	$\mathsf{cred}(ePlace, ep{:}validCandidate, X) \leftarrow ep{:}accredited(Y), \mathsf{cred}(Y, Z, X), \mathsf{similar}(Z, ep{:}MScBA) \geq 0.8$

Figure 1: Credential-Based Trust Policy

- *Constraints* are conjunctions of Boolean expressions of the form $X \theta Y$, where θ is a comparison operator (e.g., $=$, $>$, $<$, etc.) and X and Y are terms.

Example 4 *Clauses 1 and 2 in Figure 1 represent, respectively, policy PP_1 in Table 1 and Alice's credential in Example 3. The credential attributes in both clauses and the first atom in the body of clause 1 are ontology concepts. In particular,* valideCandidate*,* accredited*, and* MScBA *are concepts defined in ePlace's ontology (ep), and* MScBM *is defined in the ontology of the University of Nottingham (uon).*

Similarity statements are asserted in the form of credentials. In the language, they are represented by *similarity credential atoms* of the form $\mathsf{sim}(issuer, att_1, att_2, degree)$, where $issuer$ is the unique name of the principal issuing (and signing) the similarity credential, att_1 and att_2 are $conceptURI$s, and $degree$ is the degree of similarity (with value in the range $[0 \ldots 1]$) between att_1 and att_2 in the view of $issuer$. To increase the flexibility of her policy and enhance interoperability with other principals, a principal might accept credentials about an attribute that is similar to a given known attribute for at least a certain degree. This is expressed by means of the following *similarity constraint*:

$$\mathsf{similar}(att_1, att_2) \geq threshold$$

where $threshold$ is the minimum degree of similarity between attributes att_1 and att_2 required by the principal. This constraint can be used as any other constraint in clauses.

Example 5 *Clause 4 in Figure 1 shows how clause 1 can be modified to increase the flexibility of ePlace's policy while preserving its semantics. Now, upon receiving credential* $\mathsf{cred}(UoN, uon{:}MScBM, Alice)$ *from Alice, ePlace checks whether according to the available similarity statements the similarity degree between $uon{:}MScBM$ and $ep{:}MScBA$ is higher than the threshold. Given the similarity credential in clause 3, Alice's application is accepted by ePlace.*

3.2.3 CTM Service Architecture

Every principal runs an instance of the CTM service and communicates with other principals by exchanging queries and responses. A high level overview of the architecture of the CTM service is presented in Figure 2. Queries sent to a principal are intercepted by her Policy Enforcement Point (*PEP*). In particular, we allow three types of queries:

1. *ground queries*: request a credential issued by the recipient attesting that a given subject has a given attribute;
2. *attribute queries*: request all credentials issued by the recipient for a given subject (i.e. the set of attributes of that subject);
3. *subject queries*: request all credentials issued by the recipient for a given attribute (i.e. the set of subjects having that attribute).

Upon intercepting a query, the PEP forwards it to the Policy Decision Point (*PDP*). The PDP is the component responsible to make a decision based the policy defined by the query's recipient. When evaluating a query, the PDP can interface with (local and/or remote) ontologies to retrieve information from the knowledge base. The decision of the PDP is returned to the PEP which enforces it, possibly releasing the requested credential. Notice that, since clauses may refer to credentials issued by other principals, the decision of the PDP can in turn be

Journal of Theoretical and Applied Electronic Commerce Research
ISSN 0718–1876 Electronic Version
VOL 5 / ISSUE 2 / AUGUST 2010 / 77-96

This paper is available online at
www.jtaer.com
DOI: 10.4067/S0718-18762010000200006

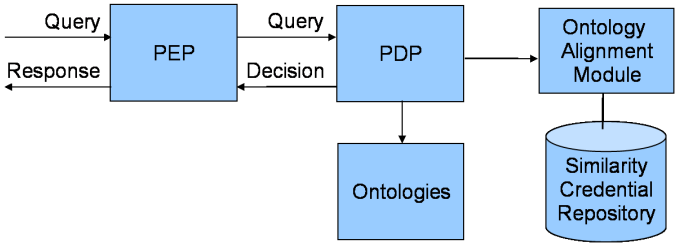

Figure 2: CTM Service Architecture

a query for such credentials. Several algorithms have been proposed to deal with the evaluation of distributed policies [1], [3], [9], [21]. The proposed architecture is flexible enough to implement any of them. Similarity constraints in the policy are resolved by contacting the *Ontology Alignment Module*. This module computes the degree of similarity between the two attributes in the constraint based on the similarity credentials available in the *Similarity Credential Repository*.

3.3 KPI-based Trust Management

Key Performance Indicators (KPI) are financial or non-financial measures or metrics used to measure progress, performance or goals of an organization and provide another source of trust feedback. KPI trust management (KPITM) is an approach in which trust evaluation is based on the experience on KPIs shared by different users.

3.3.1 KPI-based Trust model

We propose a KPI-based trust model that takes into account trust metrics based on KPI. In our approach, a user can express his trust preferences via a policy that specifies the sources of the performance indicators factors and how the factors should be combined to obtain a reputation score. According to his business objectives, the user is able to prioritize some indicators by setting a strong weight affecting the result of the trust score. These indicator values are then aggregated and normalized in order to obtain a unified reputation value.

Compared to the traditional trustworthiness models deployed in famous Internet websites, such as Amazon or E-Bay, that rely on subjective ratings, we propose a trust model based on quantitative facts (measurable performance). In the KPI-based model, each item has a semantic meaning that explains the context of the measured value related to every performance parameter (e.g., delivery time, financial results, overall budget, company growth, etc.). Combining different KPI items offers the possibility to the trustee to customize his trust evaluation by expressing complex semantic queries.

This new model adds more dynamicity to the trust evaluation due to the freshness and the diversity of the KPI values that are evolving continuously in time. An entity that wants to connect to a KPITM service in order to evaluate the trust of another entity has to select three elements: the KPI items that are relevant for him; the URI or the pointer to the source of the performance value; and the ratios affected to every item in order to prioritize some values during the trust evaluation. All this information must be contained in the core query sent to the KPITM engine that will automatically connect to the different sources, get the values of each item and compute the trust value.

Example 6 *Figure 3 describes the trust establishment process that must take place between the employability provider and the branch office representing the employee. During this process, one of the trust requirements given by the job provider could be related to some performance aspects of the employee as provided by his employer. Let us assume that a research lab is looking for a new researcher that fulfills some research requirements. One of the performance requirements could be the number of patents submitted during the last 12 months. According to job provider's business objectives, the number of filed patents must be between* $K_{min} = 1$ *and* $K_{max} = 5$ *files. Using this scale we normalize the patent number in order to fit a [0,1] scale. For example, if*

Journal of Theoretical and Applied Electronic Commerce Research
ISSN 0718–1876 Electronic Version
VOL 5 / ISSUE 2 / AUGUST 2010 / 77-96

This paper is available online at www.jtaer.com
DOI: 10.4067/S0718-18762010000200006

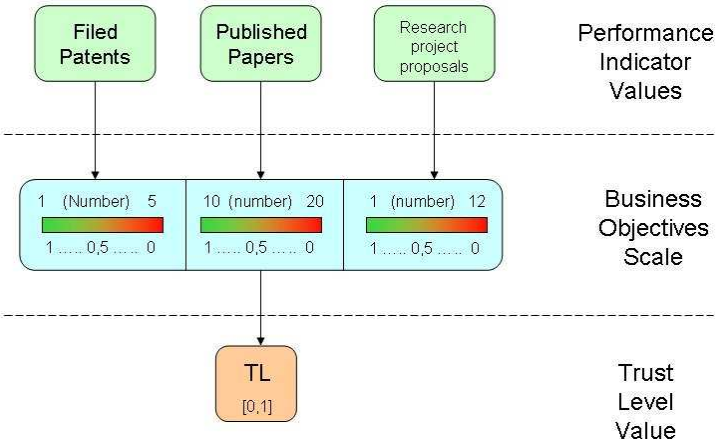

Figure 3: KPI-Based Trust Model

the number is $K_i = 3$ files, the trust value will be 0.5.

The KPI-based trust model offers the possibility to quantify the trustworthiness values according to business objectives and SLAs and permits to any business process component to determine which business partner is more trustworthy according to an objective estimation. In particular, our PKI-based trust model allows a trustee to evaluate the weight of a recommendation by applying the business objective scale of the recommender.

More formally, the KPI-based trust model that we present in this paper is composed of three complementary layers:

- *Performance Indicator Values*: are collected from the different sources providing the values related to the performance items.
- *Business Objectives Scale*: are fixed by the trustee according to the performance indicators related to their business objectives. An interval of values (min and max) must be chosen for every performance indicator in order to normalize the measured value with a [0,1] scale. The [0,1] normalization rule is written as follows:

$$\begin{cases} 1 & if \;\; K_i > K_{max} \\ 0 & if \;\; K_i < K_{min} \\ \frac{K_i - K_{min}}{K_{max} - K_{min}} & otherwise \end{cases}$$

in case of higher is better KPI normalization,

$$\begin{cases} 1 & if \;\; K_i > K_{max} \\ 0 & if \;\; K_i < K_{min} \\ \frac{K_{max} - K_i}{K_{max} - K_{min}} & otherwise \end{cases}$$

in case of lower is better KPI normalization.

Where K_i is the measured performance indicator value, K_{min} and K_{max} are the maximum and minimum values declared in the business objectives scale. If lower values are better (e.g., the delivery time example) the value is reversed by subtracting it from 1.

- *Trust Level Value*: is the aggregation of all the normalized performance indicators plus eventually some external values like the recommendation from other trusted entities.

3.3.2 KPI Trust Policy Language

The KPI trust service is a modular component that can be invoked by means of KPI estimation requests, which are specified as SQL-like queries. The following structured elements are used to specify requests and their outcome:

Entity: the entity for which the KPI is requested/given.

Journal of Theoretical and Applied Electronic Commerce Research
ISSN 0718–1876 Electronic Version
VOL 5 / ISSUE 2 / AUGUST 2010 / 77-96

This paper is available online at
www.jtaer.com
DOI: 10.4067/S0718-18762010000200006

KpiFactor: the value of one trust indicator.
Name: the name of the performance indicator.
Type: the type of the performance indicator.
Url: a reference to the provider of an indicator (e.g., a point for a DB connector).
kpiTotal: the KPI-based trust value for an entity.
Scale: the range used to normalize the performance indicator value.
Weight: the relative importance of this factor in the kpiTotal.

Queries are structured as follows:

GET *Result*
FROM *KPI Elements*
WHERE *Parameters*

where

- *Result* is either *kpiTotal*, *KpiFactor* or both and indicates whether the structured element sent as response should describe the combined kpiTotal, each of the factors or both.
- *KPI elements* is a list of structured elements describing KpiFactors using *Entity*, *Name*, *Type* and/or *Url*. This list is represented by a conjunction or disjunction of the elements.
- *Parameters*: give the *Scale* and *Weight* of each KpiFactor.

Example 7 *A manager in a research team that is willing to hire internally a researcher can send this query to the local HR database:*

GET $kpiTotal$
FROM $URL.User_1.Filled\_Pattents.US$ **AND**
$URL.User_1.Published\_Papers$ **AND**
$URL.User_1.Research\_Projects$
WHERE $[1,5]||0.5$ **AND** $[10,20]||0.3$ **AND** $[1,12]||0.2$

In this query, the manager specifies three main performance indicators that fulfill his requirements. His request asks for the number of filled US patents with a priority of 50% and a scale of [1,5] the number of publications with a priority of 30% and a scale of [10,20], and finally the number of research projects with a priority of 20% and a scale of [1,12].

4 Combined Trust Management Framework

As we can see from the scenario in Section 2, the TM framework (TMF) needs to address several challenges to fulfill the requirements of a trusted service-oriented system. In particular:

- It needs to allow the specification of trust policies combining different sources of trust. A flexible combination where new types of sources can be added should be used. Trust at different points in the scenario is based on different types of information. For example, the policy of the UoN relies on credentials, while Alice's policy determines the trust level of a provider based on feedback from past experiences.
- It needs to effectively and efficiently evaluate queries based on different trust policies. Each trust service described in Section 3 makes use of a given type of trust information, e.g. feedback for RTM, credentials for CTM. The remaining challenge is to combine these services to support multiple types of trust information.
- It needs a modular integration of the trust services into the combined service-oriented framework, to ensure easy deployment on existing and new service-oriented networks.

Below we describe possible ways of expressing combined trust policies, how such combined policies are evaluated, and how the result of the evaluation is made available to the service-oriented system.

Journal of Theoretical and Applied Electronic Commerce Research
ISSN 0718–1876 Electronic Version
VOL 5 / ISSUE 2 / AUGUST 2010 / 77-96

This paper is available online at
www.jtaer.com
DOI: 10.4067/S0718-18762010000200006

4.1 Trust Policies

The TAS^3 TMF combines different sources of trust by integrating different trust services. Each trust service defines possible policies and corresponding queries. We refer to the specification of the queries along with any policy information needed to evaluate them (some information may already be inherent to the trust service) as *trust metrics*. In particular, a trust metric captures the information needed by a service to compute a level of trust for a given principal. The term (trust) policy, from here on, refers to the overall, combined trust policy language unless stated otherwise. We support three ways of creating combined trust policies from separate metrics:

- A policy can consist of the logical combination of *independent policy conditions* of the form ⟨*service, metric, level*⟩, where each policy condition refers to a single trust *service*, and the *metric* is a reference to a trust metric of this service. In this way, each trust service can compute the trust *level* independently, being completely unaware of the existence of the other trust services.
- A policy can consist of *nested* policy conditions of the form ⟨*Service, metric*(⟨*ServiceA, metricA, levelA*⟩, *. . .*, ⟨*ServiceZ, metricZ, levelZ*⟩*), level*⟩, where the trust level computed by one service depends on the trust levels provided by other trust services. Multiple levels of nesting are possible, i.e. the metrics *metricA, . . ., metricZ* may themselves have nested trust conditions. This adds more expressiveness (see e.g. Example 8) but also more complexity to the policy language.
- Finally, a policy could be a single policy condition of the form ⟨*service, metric, level*⟩, with *metric* referring to some integrated metric combining metrics of all the supported services. The specialized evaluation method needed for the interpretation of the metric offers the potential for any type of combination and optimization of computation but is restricted to this specific language. To maintain flexibility, we will treat such a combined language with evaluation method as just another trust service.

Notice that the first approach can be seen as a special case of the second.

Example 8 *As an example of a nested policy consider requirement E_1 in Table 1, which required above average feedback coming from fellow MSc graduates. This can be expressed as*

$$\langle RTM, \texttt{feedbackAmongst}(\langle CTM, \texttt{MscCredential}\rangle), 0.75\rangle$$

Here `MscCredential` *refers to the metric (a credential query) given in Example 5. We omit the trust level for CTM conditions which is always 'true'. The metric* `feedbackAmongst` *(a database query) is similar to that of Example 2 except that the MSc list replaces the 'top 10' selection.*

Having established the possible formats of the policies, we now show how we can evaluate these policies and provide the result to the service-oriented system.

4.2 Trust Policy Decision Point

For independent policy conditions, the trust services can compute trust levels separately and we only need to combine their results. For nested policies we also need to enable interaction between the trust services. To achieve this, we introduce a *Trust Policy Decision Point (Trust PDP)*, which is responsible for coordinating and combining the trust services. The Trust PDP employs the PDP interface defined by XACML [26], allowing seamless integration within authorization mechanisms based on this popular standard.

The Trust PDP accepts two types of queries through the following interface:

evaluate(XACML-request-context): XACML-response-context;
callBack(TrustMetric,TrustLevel,entity[]): (entity,TrustLevel)[]

A trust evaluation request (see Figure 4 for an example) is sent to the Trust PDP in the form of a XACML request context describing a query of the form "is this requester trusted to perform this action on this resource?". The

Journal of Theoretical and Applied Electronic Commerce Research
ISSN 0718–1876 Electronic Version
VOL 5 / ISSUE 2 / AUGUST 2010 / 77-96

This paper is available online at
www.jtaer.com
DOI: 10.4067/S0718-18762010000200006

```
<?xml version="1.0" encoding="UTF-8"?>
<Request xmlns="urn:oasis:names:tc:xacml:1.0:context"
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Subject>
    <Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-id"
               DataType="http://www.w3.org/2001/XMLSchema#string">
      <AttributeValue>ServiceX</AttributeValue>
    </Attribute>
  <Resource>
    <Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id"
               DataType="http://www.w3.org/2001/XMLSchema#string">
      <AttributeValue>CV</AttributeValue>
    </Attribute>
  </Resource>
  <Action>
    <Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"
              DataType="http://www.w3.org/2001/XMLSchema#string">
      <AttributeValue>read</AttributeValue>
    </Attribute>
  </Action>
</Request>
```

Figure 4: Example Trust Evaluation Request

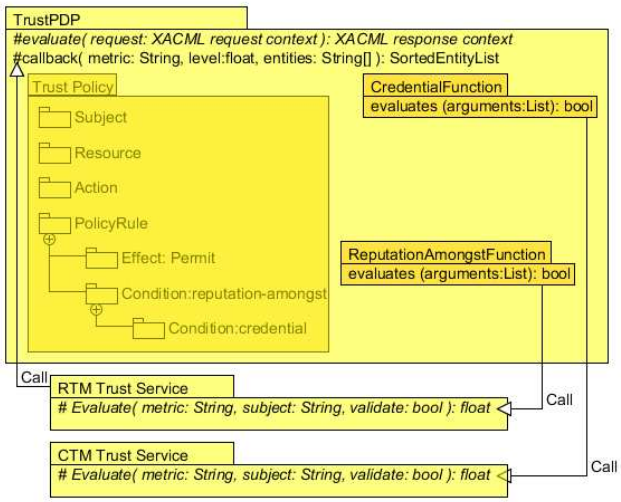

Figure 5: Trust PDP with Trust Policy and Trust Services

Trust PDP will evaluate the combined trust policies with the help of the trust services. The returned decision, encoded in a XACML response context, can be *permit* (trusted), *deny* (not trusted), *not applicable* (no applicable trust policies) or *indeterminate* (error occurred).

The Trust PDP can be configured to combine any number of trust services. Interaction between them is managed by the Trust PDP by means of a *callback* function. This function takes a trust metric, a required trust level on this metric and a list of principals to be assessed. It returns the list of principals sorted by their trust level. The list of principals to rank is optional; if it is not given, the Trust PDP returns a list of arbitrary principals with the required trust level. The required level is also optional; if not given, the Trust PDP returns the trust level of all the (listed) principals.

Example 9 *Figure 5 shows the Trust PDP with the policy of Example 8, which uses nested policy conditions. The Trust PDP uses the callback function to retrieve from the CTM service the list of principals that have a MSc credential. Then, it sends the list to the RTM service to compute the trust level of the provider based on the feedback provided by the principals in the list.*

Figure 5 shows a minimum interface that needs to be supplied by the trust services. The actual implementation offers a richer interface allowing some optimization of the queries as described below.

Journal of Theoretical and Applied Electronic Commerce Research
ISSN 0718–1876 Electronic Version
VOL 5 / ISSUE 2 / AUGUST 2010 / 77-96

This paper is available online at
www.jtaer.com
DOI: 10.4067/S0718-18762010000200006

4.3 Adding Trust Services

We have shown the three trust services included in our implementation. Additional trust services can be easily added provided that they meet the following requirements:

- *A trust service must supply a trust metric language.* A trust metric describes how the trust service computes its level of trust. Notice that this language does not need to capture the way in which trust is computed; it needs only to capture the information needed to build a trust level query.
- *A trust service must supply a trust level evaluation interface.* This interface describes the basic functionality of the trust service; the trust service needs to provide functions to be called by the Trust PDP for the evaluation of a trust metric. The returned trust level must be a standard numeric value (including Boolean) or the trust service must provide a method to compare trust levels.

The following optional features should be supported by the trust service to support the full functionality of the trust management framework.

- *A trust service should supply a parametrized metric language which allows the nesting of (other) trust metrics.* Without this feature, the trust service is limited to independent policy conditions only (see Section 4.1).
- *A trust service should offer an interface for discovering principals with a given minimum trust level.* This feature is needed to support discovery of trusted principals by the Trust PDP.

4.4 Implementation

To support the evaluation of trust policies, we have extended XACML using standard extension mechanisms provided by XACML itself. XACML allows the introduction of user-defined functions in addition to the default ones [26]. In particular, we define functions which, given a set of arguments, generate a request in a format appropriate for a certain trust service.

Figure 6 shows an example policy encoding that the average feedback score amongst certified students from Nottingham must be at least 0.8. For readability a shorthand notation is used for the data types, e.g. *String* rather than `"http://www.w3.org/2001/XMLSchema#string"`. Notice the use of the locally defined trust function to indicate the trust services to be called.

The prototype implementation of the TAS^3 TMF incorporates a Trust PDP in combination with the three trust services described in Section 3. When evaluating a trust metric containing one or more credentials, the TAS^3 Trust-PDP contacts the PEP of the CTM service by sending a SAML [27] $<$`AttributeQuery`$>$. The PEP response is returned to the Trust-PDP as a SAML $<$`AttributeStatement`$>$. To support subject queries (see Section 3.2), we allow for queries that do not specify the subject (which is a mandatory field for a SAML $<$`AttributeQuery`$>$). When the Trust PDP needs to evaluate a RTM statement, it invokes the RTM service by generating an SQL query similar to the one given in Example 2. Similarly, the KPI trust service accepts requests in an SQL-like language (see Example 7).

5 Trust & Privacy

In this section we address the implicit trust relations and the privacy aspects of the TMF. While establishing trust is required for secure sharing of private data, the data that is used to establish trust may itself be sensitive. First, we discuss trust relationships between the different components of the TMF and implication for privacy, and then we analyze privacy requirements and solutions for the deployment of privacy-friendly trust services.

Journal of Theoretical and Applied Electronic Commerce Research
ISSN 0718–1876 Electronic Version
VOL 5 / ISSUE 2 / AUGUST 2010 / 77-96

This paper is available online at
www.jtaer.com
DOI: 10.4067/S0718-18762010000200006

```
<?xml version="1.0" encoding="UTF-8"?>
<Policy xmlns="urn:oasis:names:tc:xacml:1.0:policy"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        PolicyId="TSP:Rep:requester"
        RuleCombiningAlgId="urn:oasis:names:tc:xacml:1.0:rule-combining-
algorithm:permit-overrides">
 <Description>Allow read CV if good reputation amongst students</Description>
 <Target><Subjects><AnySubject/></Subjects> <Resources><AnyResource/><Resource>
        <Actions> <AnyAction/> </Actions> </Target>
 <Rule RuleId="trustcondition:read:cv" Effect="Permit">
  <Target><Subjects><AnySubject/></Subjects>
  <Resources><Resource>
   <ResourceMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
    <AttributeValue DataType=String>CV</AttributeValue>
    <ResourceAttributeDesignator DataType=String
     AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id"/>
    </ResourceMatch></Resources>
  <Actions><Action>
    <ActionMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
     <AttributeValue DataType=String>read</AttributeValue>
     <ActionAttributeDesignator DataType=String
      AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"/>
    </ActionMatch></Action> </Actions>
 </Target>
 <Condition
FunctionId="http://localhost:8080/trustpdp/names/function#avgfeedback-amongst">
  <!-- Trusted raters: Nothingham.Students -->
  <Apply FunctionId="http://localhost:8080/trustpdp/names/function#credential-
subject-list">
    <AttributeValue DataType=String>Nottingham</AttributeValue>
    <AttributeValue DataType=String>Student</AttributeValue>
  </Apply>
  <!-- The requester (subject id) -->
  <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-one-and-only">
    <SubjectAttributeDesignator DataType=String
     AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-id" />
  </Apply>
  <!-- The required score on this metric -->
   <AttributeValue DataType=Double>0.8</AttributeValue>
  </Condition>
 </Rule>
 <Rule RuleId="DenyAllOthers" Effect="Deny"/>
</Policy>
```

Figure 6: Example Trust Policy

5.1 Trust in the Trust Management Framework

The first aspect to notice is that the TMF is embedded in a larger trusted and secure architecture design. The TAS^3 project defines vetting of the federation of services, logging and auditing procedures and a legal framework working in unison with the technology. All these help achieve accountability of participants and build a setting in which a TMF can be successfully deployed.

A relying party, typically a service-oriented system, needs to be able to depend on the trust decisions made by the Trust PDP. A service can run its own Trust PDP on site to ensure the control needed to achieve the required trust in this key component. Trust services can be run locally or external.

Trust services need to be trusted with the trust information they process, both from a privacy standpoint and from a security/reliability point. For instance, providers of trust information need to be sure that their privacy is protected and their provided information is not abused. Consumers of trust scores need to be able to trust that the provided trust levels are accurate (at least as much as is possible based on the trust information available to the trust service). In TAS^3 the providers of trust services are subject to the same vetting and logging-auditing procedures just like any other service providers. Still as the trust services play a pivotal role in ensuring a trusted and secure network they especially need to be trustworthy. One could write trust policies to evaluate the trustworthiness of trust services themselves but this seems unlikely to be used in practice. A more likely scenario is that a fixed set of trust services provided by trusted parties are chosen.

For the evaluation of trust policies, trust services need to share information with the Trust PDP and, in case of nested policies, with other trust services. In this setting, the question is whether a trust services trust the Trust PDP and other trust services with this information. In the scenario of Section 2, we are likely to have a reputation

Journal of Theoretical and Applied Electronic Commerce Research
ISSN 0718–1876 Electronic Version
VOL 5 / ISSUE 2 / AUGUST 2010 / 77-96

This paper is available online at
www.jtaer.com
DOI: 10.4067/S0718-18762010000200006

server run by the university, which is trusted by students both with their feedback and their credentials. The Trust PDP controlling access to their PII is also run by the university and thus equally trusted. The Trust PDP run by a placement provider, controlling access to e.g. open managerial positions, would be less trusted with respect to feedback information, and privacy protection should be applied as described below.

Multiple reputation services might exist; the university might maintain a reputation service establishing reputation of placement providers, while a placement provider may assess itself the reputation of the companies at which it places students. Reputation may be shared, but neither party is trusted with all the information in the feedback databases on which these reputations are based. In this case the resulting reputation can still be combined using the Trust PDP. Complex centrality measure computations however are not possible as they typically require sharing too much information about the feedback between the services.

5.2 Privacy enhancement for RTM

Users of reputation systems are exposed to various privacy risks. These risks range from identity disclosure (i.e., the identity of the user giving feedback is revealed) to attribute disclosure (i.e., the type of service on which the user gives feedback is revealed). We will structure our discussion of privacy protection according to the two major tasks of RTM: the collection of feedback data and the computation of reputation values.

Regarding the collection of feedback data, we propose the following approaches to protect the privacy of the users: (i) *Anonymous:* Users do not reveal their identities, i.e., feedback is given anonymously. While this approach conceals the identity of the user, it offers only limited value of the feedback. Since more sophisticated trust metrics require persistent identities, only basic trust metrics such as average feedback can be applied. (ii) *Anonymous with linked attributes:* The identity of the users remains undisclosed, but they assign certain attributes (e.g., their profession) with the feedback. While the value of the feedback is still limited, this approach allows for nested trust metrics (e.g., “average feedback amongst certified students”). (iii) *Identity disclosure:* Users disclose their (trust server) identity with the feedback. This approach offers the highest value of the feedback, allowing for trust metrics based on centralities [14], [39]. To protect the privacy of users, the identities used to give feedback should not be linkable to identities used by other service providers.

Existing RTM systems rely on identities which are persistent over time and valid for multiple domains. That is, neither the identity of the subject giving feedback nor the identity of the object on which feedback is given may change over multiple sessions or vary for different service providers. The TAS^3 architecture provides a federated identity management (FIDM) which lets users use different, non-linkable identifiers at different services and provides a linking service [7] with which users can combine attributes provided by different identity providers. This allows users to increase the impact of their feedback by linking relevant attributes without revealing their identity.

The RTM service uses identity mapping to maintain persistent identities. Identity mapping converts an identity from one domain to an identity valid in another domain. The conversion is made by an identity provider (IdP) that trusts the starting domain and is trusted by the ending domain. If a user chooses to disclose its identity when submitting feedback for an interaction, the RTM service maps the provided (transient) identity of the user to the persistent identity issued by the RTM service itself. The RTM service only knows that this user has had an interaction with the given service provider, but does not get to know the other identifiers of the user. Vice versa, the service with which the user interacted will not get to know the persistent identity used by the RTM service.

The second major task of RTM is to compute reputation values based on the feedback data. One approach to privacy protection is to restrict the access to feedback data by publishing aggregated reputation values only. In our opinion, such an approach is infeasible because it forces all users to agree on one aggregation scheme. In the following, we present several protection mechanisms which preserve the privacy of raters while granting access to unaggregated feedback data. Users can then apply individual metrics to derive reputation values from feedback data.

If identities are anonymized naively, e.g., the identifiers are simply replaced by random numbers, subjects can still be identified if some additional attributes such as place of residence are available. This issue has been addressed by a data protection model named k-anonymity [30]. A dataset satisfies k-anonymity if each record in

Journal of Theoretical and Applied Electronic Commerce Research
ISSN 0718–1876 Electronic Version
VOL 5 / ISSUE 2 / AUGUST 2010 / 77-96

This paper is available online at
www.jtaer.com
DOI: 10.4067/S0718-18762010000200006

this dataset is indistinguishable from at least $k-1$ other records with respect to some identifying attributes.

Further privacy problems can arise if the service provision (e.g., some medical treatment) is sensitive information. Since k-anonymity cannot prevent attribute disclosure in datasets which homogeneous records (such as feedback datasets), the notion of l-diversity [23] has been proposed. This model requires that each set of records has at least l well-represented values for each sensitive attribute.

To protect the privacy of users in published feedback datasets, we demand that these datasets satisfy l-diversity. This requirement is hard to meet in practice because new feedback is continuously added by the users. To grant access to the most current feedback items, feedback data must be anonymized on the fly. A linear-time algorithm for computing anonymized tables has been proposed [38] which obeys the l-diversity requirement. Using this algorithm allows reputation systems to continuously publish feedback datasets while preventing identity disclosure entirely and avoiding attribute disclosure in most cases.

5.3 Privacy enhancement for CTM

In CTM, credentials may be confidential and this has an impact both on what may be revealed to another principal as well as on the credential release rule evaluation process. There are thus two aspects that are worth clarifying.

The first aspect is related to Trust Negotiation [36], [37]. CTM systems are typically open systems in which anyone is by default allowed to query any principal. However, the result of a query may reveal sensitive information. The idea behind Trust Negotiation is that a principal, before revealing information about herself to the requester, may ask him additional information (in the form of credentials). In turn, the requester itself may protect its credentials with a policy, initiating a communication which may eventually lead to the evaluation of the initial request. Hence the name trust *negotiation*. This process could in principle go on indefinitely long, and could also deadlock (the canonical example being a secret agent willing to show that he is a secret agent only to someone who has already proved to be a secret agent).

Example 10 *Suppose that Alice applies for a vacancy in the hospital of Nottingham. For this kind of position, the on-line placement provider requires Alice to submit her personal health record (PHR). However, the PHR clearly contains sensitive information, and Alice is willing to disclose it only to "trustworthy" institutions certified by the Better Bureau of Privacy. Therefore, Alice asks the provider to show her such credential, initiating a negotiation process.*

A second aspect in which credential sensitivity plays an important role is in the evaluation of credential release policies itself. Recall that one of the distinctive features of CTM is that principals are allowed to refer to each other's policies.

Example 11 *The Better Bureau of Privacy (BBP) releases a "trustworthy" certificate according to the following rule: "An institution is* trustworthy *if it is* reliable *according to the Office of Fair Trading (OFT)", i.e., if no fraud has been reported against it. In the language presented in Section 3.2.2, this statement is represented as follows:*

$$\mathrm{cred}(BBP, bbp : trustworthy, X) \leftarrow \mathrm{cred}(OFT, oft : reliable, X).$$

Here, BBP's policy depends on the definition of 'reliable' in OFT's policy, which in turn may depend on policies specified by other principals.

In the example above, it is clear that the BBP needs the OFT to disclose (part of) its *extensional database*, i.e., the requested credential. What might be less clear is that in most of the existing CTM systems (e.g., [19], [9]), the OFT would eventually have to reveal to the BBP (or to some other external authority) also part of her *intensional database*, that is, the actual policy statements used to determine whether some institution is *reliable*. The reason is that in presence of mutually recursive policy statements it is very hard to recognize when the computation is finished in a distributed way. Novel distributed algorithms for policy evaluation are needed. In particular, they should support decentralized loop detection and termination and as such allow to evaluate distributed policies while "minimizing" the information that principals reveal to each other.

Journal of Theoretical and Applied Electronic Commerce Research
ISSN 0718–1876 Electronic Version
VOL 5 / ISSUE 2 / AUGUST 2010 / 77-96

This paper is available online at
www.jtaer.com
DOI: 10.4067/S0718-18762010000200006

5.4 Privacy enhancement for KPITM

Factors of KPIs are already precomputed scores which are typically available to any one who is allowed to see the resulting KPI score. These factors can be collected from different sources more or less public. Although these sources can publish public performance values, some details related to the estimation parameters can be private. We can categorize the KPI sources in three privacy levels:

- *Public*: The performance indicator needed to calculate the trust values are publicly available and all the parameters related to these indicators are public. For example the financial quotations of the companies that are publicly accessible by any user without restrictions.
- *Confidential*: some performance indicators are computed from internal confidential sources with a restricted access. For example, the HR information of a company can be considered as confidential data, and can be accessed only by some authorized people from the company (HR, Managers, CTO, etc.). In this case the KPITM service needs to be allowed access by the access control system used to protect the data. Also, the resulting trust scores should be treated as being equally confidential as the input they are made from; the available data protection mechanisms should also be applied to this data.
- *Sensitive parameters*: some performance indicators are issued from private or sensitive parameters. This means that the value of the performance indicator is public but the parameters used to get these values should remain confidential. For instance, if the performance indicator is based on some statistics related to personal information collected from a company's employee or client database, anonymization techniques (compliant with k-anonymity, l-diversity, or t-closeness) can be used to protect sensitive information that can be used to re-identify private and personal data.

As unexpected side effect public KPIs may be exploited using correlation techniques exposing private information. This threat can be reduced by the usage of some privacy risk evaluation tools [31] that are able to estimate the re-identification probability of private data when exposing public ones.

6 Conclusions & Future Work

The TAS^3 TMF incorporates trust decisions in service-oriented systems in a practical non-disruptive way. It offers a flexible and extensible system which can be adjusted to the needs of different scenarios and settings as well as changing user requirements.

We are testing the TMF, using a basic prototype implementation of the framework, with scenarios from the employability and e-health settings. Although the use case presented in Section 2 offers a clear and simple federation led by the university, which provides a foundation for building trust, such a clear trust anchor is not available in all the settings that we considered. The trusted federation building mechanisms (vetting, legal framework, logging and auditing, etc.) of TAS^3 methods will be needed to create a setting where the TMF can be successfully deployed. It will also be necessary to implement the privacy enhancements discussed in Section 5 and adjust security, trust and privacy mechanisms dependent on the situation.

Further planned work addresses limits of effective policies; what kind of policies can be effectively computed in what settings. Also performance enhancements such as minimizing overhead imposed by the trust evaluation and ensuring reliable behavior under/in high load situations will be addressed. Finally, we also plan to investigate the use of the TMF on low resource devices (already supported in part thanks to remote trust evaluation offered by the trust services).

To ensure that we are actually supporting the user rather than building another hurdle that the user will try to work around, we plan to look at the user perception of trust. In particular, we are analyzing the actual trust requirements of the users, what kind of trust policies do they formulate and use, and how effective is the TMF in building trust in the architecture itself and the services that run on it.

This paper is available online at
www.jtaer.com
DOI: 10.4067/S0718-18762010000200006

Acknowledgements

This work has been supported in part by European Commission FP7 TAS^3 project, nr. 216287.

References

[1] M. Alves, C. V. Damásio, W. Nejdl, and D. Olmedilla, A distributed tabling algorithm for rule based policy systems, in Proceedings of the 7th International Workshop on Policies for Distributed Systems and Networks, London, Ontario, Canada, IEEE Computer Society, 2006, pp. 123-132.

[2] D. Artz and Y. Gil, A survey of trust in computer science and the semantic web, Journal of Web Semantics, vol. 5, no. 2, pp. 58-71, 2007.

[3] M. Y. Becker and P. Sewell, Cassandra: Distributed access control policies with tunable expressiveness, in Proceedings of the 5th International Workshop on Policies for Distributed Systems and Networks, Yorktown Heights, New York, USA, IEEE Computer Society, 2004, pp. 159-168.

[4] M. Blaze, J. Feigenbaum, J. Ioannidis, and A. Keromytis, The role of trust management in distributed systems security, in Secure Internet Programming: Security Issues for Mobile and Distributed Objects. LNCS 1603, Springer, 1999, pp. 185-210.

[5] M. Blaze, J. Feigenbaum, and J. Lacy, Decentralized trust management, in Proceedings of 17th IEEE Symposium on Security and Privacy, Oakland, California, USA, IEEE Computer Society, 1996, pp. 164-173.

[6] P. Bonatti, C. Duma, D. Olmedilla, and N. Shahmehri, An integration of reputation-based and policy-based trust management, in Proceedings of the Semantic Web Policy Workshop, Galway, Ireland, 2005, pp. 137-142.

[7] D. W. Chadwick, G. Inman, and N. Klingenstein, A conceptual model for attribute aggregation, Future Generation Computer Systems, vol. 26, no. 7, pp. 1043-1052, 2010.

[8] N. Choi, I. Y. Song, and H. Han, A survey on ontology mapping, ACM SIGMOD Record, vol. 35, no. 3, pp. 34-41, 2006.

[9] M. Czenko and S. Etalle, Core TuLiP logic programming for trust management, in Proceedings of the 23rd International Conference on Logic Programming, Porto, Portugal, Springer, 2007, pp. 380-394.

[10] L. D. Ngan, T. M. Hang, and A. E. S. Goh, Semantic similarity between concepts from different OWL ontologies, in Proceedings of the 4th International Conference on Industrial Informatics, Singapore, IEEE Computer Society, 2006, pp. 618-623.

[11] P. Herings, G. van der Laan, and D. Talman, Measuring the power of nodes in digraphs, Center for Economic Research, Tilburg University, The Netherlands, Discussion paper, 2001.

[12] P. Jaillon, M. Roelens, X. Serpaggi, and H. Vu, Towards an approach for hybrid trust model, in Proceedings of the 8th International Workshop on Policies for Distributed Systems and Networks, Bologna, Italy, IEEE Computer Society, 2007, pp. 279.

[13] A. Jøsang, R. Ismail, and C. Boyd, A survey of trust and reputation systems for online service provision, Decision Support Systems, vol. 43, no. 2, pp. 618-644, 2007.

[14] S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina, The EigenTrust algorithm for reputation management in P2P networks, in Proceedings of the 12th International Conference on World Wide Web, New York, NY, USA, ACM, 2003, pp. 640-651.

[15] J. M. Kleinberg, Authoritative sources in a hyperlinked environment, Journal of the ACM, vol. 46, no. 5, pp. 604-632, 1999.

[16] K. Krukow, M. Nielsen, and V. Sassone, A framework for concrete reputation-systems with applications to history-based access control, in Proceedings of the 12th Conference on Computer and Communications Security, Alexandria, Virginia, USA, ACM, 2005, pp. 260-269.

[17] A. N. Langville and C. D. Meyer, Deeper inside PageRank, Internet Mathematics, vol. 1, no. 3, pp. 335-380, 2004.

[18] A. J. Lee and T. Yu, Towards a dynamic and composable model of trust, in Proceedings of the 14th Symposium on Access Control Models and Technologies, New York, NY, USA, ACM, 2009, pp. 217-226.

This paper is available online at
www.jtaer.com
DOI: 10.4067/S0718-18762010000200006

[19] N. Li, J. Mitchell, and W. Winsborough, Design of a role-based trust-management framework, in Proceedings of the 23rd Symposium on Security and Privacy, Berkeley, California, USA, IEEE Computer Society, 2002, pp. 114-130.

[20] N. Li and J. C. Mitchell, Datalog with constraints: A foundation for trust management Languages, in Proceedings of the 5th International Symposium on Practical Aspects of Declarative Languages, New Orleans, Louisiana, USA, Springer, 2003, pp. 58-73.

[21] N. Li, W. H. Winsborough, and J. C. Mitchell, Distributed credential chain discovery in trust management, Journal of Computer Security, vol. 11, no. 1, pp. 35-86, 2003.

[22] N. Lin, Foundations of Social Research. New York: McGraw-Hill, 1976.

[23] A. Machanavajjhala, J. Gehrke, D. Kifer, and M. Venkitasubramaniam, l-Diversity: Privacy beyond k-anonymity, in Proceedings of the 22nd International Conference on Data Engineering, Atlanta, Georgia, USA, IEEE Computer Society, 2006, pp. 24.

[24] G. Montagnon, Design requirements, TAS^3 consortium, Research report D1.4, 2008.

[25] W. Nejdl, D. Olmedilla, and M. Winslett, PeerTrust: Automated trust negotiation for peers on the semantic web, in Proceedings of Workshop on Secure Data Management in a Connected World, Toronto, Canada, Springer, 2004, pp. 118-132.

[26] OASIS, eXtensible Access Control Markup Language (XACML) v2.0, OASIS Standard, 2005.

[27] OASIS, Security Assertion Markup Language (SAML) v2.0, OASIS Standard, 2005.

[28] L. Page, S. Brin, R. Motwani, and T. Winograd, The PageRank citation ranking: Bringing order to the Web, Stanford University, California, USA, Technical Report 1999-66, 1999.

[29] R. Rivest and B. Lampson.(1996, October) SDSI – A Simple Distributed Security Infrastructure. [Online]. Available: http://theory.lcs.mit.edu/∼rivest/sdsi11.html.

[30] L. Sweeney, k-anonymity: A model for protecting privacy, International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, vol. 10, no. 5, pp. 557-570, 2002.

[31] S. Trabelsi, V. Salsgeber, M. Bezzi, and G. Montagnon, Data disclosure risk evaluation, in Proceedings of the 4th International Conference on Risks and Security of Internet and Systems, Toulouse, France, IEEE Computer Society, 2009, pp. 35-42.

[32] D. Trivellato, F. Spiessens, N. Zannone, and S. Etalle, POLIPO: Policies & ontoLogies for interoperability, portability, and autonomy, in Proceedings of the 10th International Workshop on Policies for Distributed Systems and Networks, London, UK, IEEE Computer Society, 2009, pp. 110-113.

[33] D. Trivellato, F. Spiessens, N. Zannone, and S. Etalle, Reputation-based ontology alignment for autonomy and interoperability in distributed access control, in Proceedings of the 12th International Conference on Computational Science and Engineering, Vancouver, British Columbia, Canada, IEEE Computer Society, 2009, vol. 3, pp. 252-258.

[34] T. W. Valente and R. K. Foreman, Integration and radiality: Measuring the extent of an individual's connectedness and reachability in a network, Social Networks, vol. 20, no. 1, pp. 89-105, 1998.

[35] A. G. West, A. J. Aviv, J. Chang, V. S. Prabhu, M. Blaze, S. Kannan, I. Lee, J. M. Smith, and O. Sokolsky, QuanTM: a quantitative trust management system, in Proceedings of the 2nd European Workshop on System Security, Nuremburg, Germany, ACM, 2009, pp. 28-35.

[36] W. H. Winsborough, K. E. Seamons, and V. E. Jones, Automated trust negotiation, in Proceedings of the DARPA Information Survivability Conference & Exposition, Hilton Head, South Carolina, USA, IEEE Computer Society, 2000, vol. 1, pp. 88-102.

[37] M. Winslett, An introduction to trust negotiation, in Proceedings of the 1st International Conference on Trust Management, Heraklion, Crete, Greece, Springer, 2003, pp. 275-283.

[38] X. Xiao and Y. Tao, Anatomy: simple and effective privacy preservation, in Proceedings of the 32nd International Conference on Very Large Data Bases, Seoul, Korea, VLDB Endowment, 2006, pp. 139-150.

[39] A. Yamamoto, D. Asahara, T. Itao, S. Tanaka, and T. Suda, Distributed pagerank: A distributed reputation model for open peer-to-peer networks, in Proceedings of Symposium on Applications and the Internet-Workshops, Tokyo, Japan, IEEE Computer Society, 2004, pp. 389.