Derivation and use of trust and risk management parameters in dynamic federated environments

Latifa Boursas[1] and Wolfgang Hommel[2]

[1] Munich Network Management Team
Technische Universität München
`boursas@tum.de`
[2] Munich Network Management Team
Leibniz Supercomputing Centre
`hommel@lrz.de`

Abstract. Traditionally, the risks associated with granting customers and their users access to services and resources are mitigated by contractual frameworks, such as service level agreements (SLAs). However, in large and highly dynamic federated environments, also external and previously unknown users must be handled in an automated manner, which limits the available options to negotiate SLAs. In this paper we present how risk management on the service provider side and user trust level management frameworks can be combined and applied to policy-based access control mechanisms.

1 Motivation for managing risk and trust in dynamic federated environments

Specifying quality of service parameters and penalties for not fulfilling them is a classical approach to mitigate several of the customer's and the service provider's risks resulting from inter-organizational dependencies and business connections. Service level management and its interfaces to other IT service management (ITSM) processes, especially financial and security management, have been motivated, analyzed, and improved by both researchers and practitioners over the past decades and are impossible to imagine away today.

However, new services, such as distributed collaborative environments, have such high dynamics and fluctuation regarding involved organizations, resources, and users that new technical measures are required to improve the reactivity of ITSM workflows and thus support the underlying business processes.

In this paper, we present a risk based resource protection approach for dynamic federated environments (DFEs), i. e. for inter-organizational scenarios in which the involved entities are bound by a contractual framework but must support the temporary inclusion of external entities. Rather obviously, this results in new requirements for access control mechanisms on the service operation level, because even although sharing resources in such environments must be quick to set up, misuse and unauthorized access must still be detectable and preventable by proper configuration.

Characterized by the *locality over globality* paradigm, the service providers as resource owners must have the possibility to determine how, when, and which resources are available for which kind of access by whom. Granting permissions to a customer's users, which is typically regulated by service level agreements (SLAs), reflects that each of these users is sufficiently *trusted* and that the *risk* of incidents caused by the users is outweighted by the mutual benefits.

Various access control models have successfully been applied to intra-organizational scenarios and have later been extended for inter-organizational and federation scenarios. Several variants of standards like role based access control (RBAC) and its successors, e. g. attribute based access control (ABAC), allow the delegation of administration on the one hand and privileges on the other hand; unfortunately, only seemingly they are a good starting point for the inclusion of external entities in DFEs, because privileges may only be delegated to those principals which are already known in the federation. This means that a digital identity that has been created by one of the involved organizations must be assigned to the user a priori, which causes the very same timeliness, cost, and complexity problems we strive to avoid.

The new approach presented in this paper is based upon our previous work on trust based access management (TBAC) and proposes the combined use of both, formula-based trust quantification and risk assessment, in dynamic access control policies. In DFEs, an external principal may be vouched for by one or more known entities, which themselves may or may not be members of the federation; deriving from how trustworthy each warrantor is, an initial trust level for the external principal can be calculated. This trust level changes over time, typically based on feedback and recommendation mechanisms known from reputation management; however, service providers must always consider the risk of granting resource access to previously unknown users and cannot afford to rely solely on vague trust recommendations, especially because several reputation management approaches that were used in e-commerce environments turned out to be bogus or susceptible to fraud.

The remainder of this paper is structured as follows: In the next section we outline a DFE scenario which serves as an example in the presentation of our risk-based management approach in section 3. Our data model, which is to be used in dynamic access control approaches, and our RDF/LDAP-based implementation are discussed in section 4. Competitive approaches and related work are summed up in section 5; the paper is concluded by a discussion of the current status as well as the next steps of our work.

2   Real-world scenario: Distributed eLearning federations

To illustrate the importance of risk assessment on the one hand and the application area of our solution on the other hand, we present a simplified view of a real-world eLearning scenario in the MNM-Team's environment. Two of the Munich universities, LMU and TUM, offer several joint study courses, e. g. medicine and bio-informatics; students of these study courses are enrolled in both universities and thus must be able to use both universities' IT services, including the learning management systems (LMS).

Additionaly, more than 30 higher education institutions (HEIs) in the German state of Bavaria are carriers of the so-called Virtual University Bavaria (VHB); the VHB acts as a broker between the students and each HEI's local LMS, which results in a highly distributed federated environment with a focus on eLearning services. Given the naturally high fluctuation of students and the regular changes concerning which eLearning courses are offered, this scenario represents a DFE as discussed in the introduction.

Regarding the SLA for a typical LMS and the privileges derived thereof, we naturally need to distinguish between users and resources. Resources include the various types of LMS content, e.g. lecture notes, exercises, and presentation slides. To handle the masses of users efficiently, RBAC roles, such as students, lecturer, and LMS administrator, are defined. It is noteworthy that the same terminology for at least a subset of the RBAC roles is also used for the description of business roles, which are utilized in the textual formulation of SLAs; for large federations, this implies that a common terminology is required, which is often hard to achieve (for example, the terms student, faculty, staff, and alum have slightly different semantics in the USA and in Europe).

On the technical level, a LMS system can be broken down into two types of objects which are essential for the formulation of access control rules and policies, as shown in figure 1:

- *Learning Content Objects* (LCOs) basically represent the course material created or coached by the trainers and consumed by the learners. This learning content is usually stored in object-oriented multimedia databases along with various metadata; in our solution, we extend the latter to include risk parameters that can be evaluated within access control policies.
- *Identity Information* (IDI) provides relevant information about the LMS users. Traditionally, the attributes of each user profile object link it to one or more of the defined RBAC roles, which are more efficient to use in access control policies than long lists of usernames that would have the same privileges. However, in order to improve the dynamics of role definitions, we use individual user trust levels that complement the object risk parameters in our solution.

However, as also shown in figure 1, an institution's LMS often is a distributed system itself. In our scenario, the Leibniz Supercomputing Centre (LRZ) operates the multimedia databases and streaming servers of TUM's LMS; these two services are also used by other LRZ customers, which necessitates an additional access control layer on the LRZ side. Furthermore, LCOs are managed by different content suppliers, and trainers as well as learners can be affiliated with more than one HEI. In practice, especially concerning the medicine study courses, the LMS must additionally support the handling of third party LCO vendors, external instructors, and guest students.

SLAs exist between TUM and its external suppliers, and contractual frameworks, e.g. for the students, exist; because several study courses cannot be completed anymore without taking tests involving certain eLearning classes, guarantees regarding several classical quality of service parameters, such as service avaibility and mean time to repair, must be made. The typically short lifetime of eLearning classes, which is about 10–12 weeks, and the skew that all the classes start at the same day at the beginning of
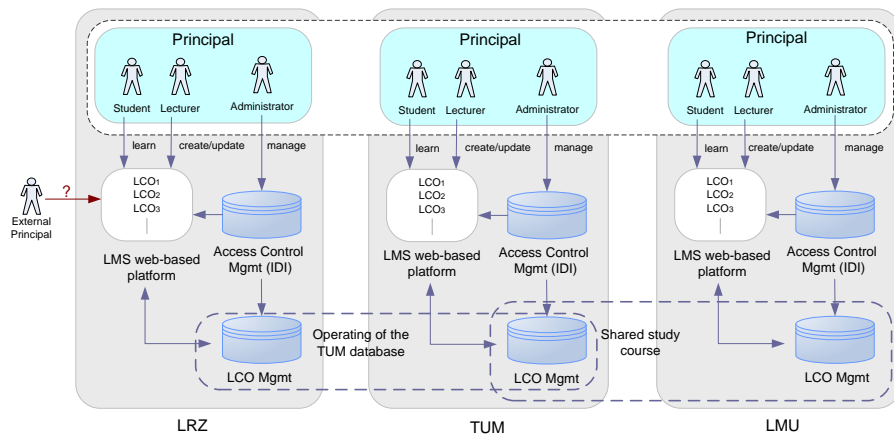
Fig. 1. A dynamic federated environment for eLearning services

each semester, make traditional service level management next to impossible to handle on a per-service-instance-and-involved-party basis. Here, a dynamic TBAC approach, which considers the so-called trust level of each user, can greatly reduce the administrative overhead. We will provide details about the derivation of the trust level and its application for risk management in the next section.

3 A risk and trust based access control management approach for DFEs

Due to the great variety of trust and risk metrics available on both, the algorithmic and the management level, we first define the terms and data structures we use throughout our work. We then discuss the workflows we use for the quantification of trust levels and risks. The implementation and application of the presented concepts are discussed in section 4.

3.1 Data structures for quantification and calculations

In previous work, we have shown how the trust metric defined in our TBAC approach can be applied to solve authorization and usage control problems in federated environments (cp. [2, 3]). We have demonstrated that a quantification of the trust in a principal can be derived from examining service and action specific evidence of prior interactions; this reflects that, for example, an instructor who repeats the same eLearning class for the fifth time without any incidents *may* be considered more trustworthy than a first-time participant – please note that the selection and weight of input parameters is of course specific to each scenario. In the following, we summarize these results and introduce the variables we are using in this approach:

Trust levels: A trust-based access control decision is primarily based on a set of access control policies $P : \{p_1, p_2, \ldots, p_n\}$ that must define which subject set $S : s_1, s_2, \ldots, s_n$ (i.e. roles or individual users) may perform which action set $A : \{a_1, a_2, \ldots, a_n\}$ (e.g. create, modify, delete, or download) on which resource set $R : \{r_1, r_2, \ldots, r_n\}$ (an LCO is an example of such a resource) under which condition set $C : \{c_1, c_2, \ldots, c_n\}$ at any point in time t within the cooperation lifecycle T. An example for the mapping to the rules contained in policies will be detailed below.

We observe that trust, quantified as a user's trust level tl, depends on the attempted actions, the involved resources, and the point in time: $tl_{(s,A',R',t)} = trust(s, A', R', H, T_D)$, where H represents the user's reputation history and T_D assigns weights to principal introductory protocols as discussed below. We normalize the result to a continuous scale in the range $tl \in [0, 1]$, where 1 indicates absolute trust and 0 indicates absolute distrust. tl may also take the value of -1 in case the trust level cannot be determined, e.g. due to missing input parameters.

Derived from our work presented in [2], we distinguish T_D as follows:

– *Trust by reputation*, i.e. the principal's reputation is defined to be the quantification of conclusions drawn from observations of previous interactions that the principal was involved in, which must be witnessed either by the judging principal or relying on other sufficiently trusted entities. In this work we implement this mechanism by defining the current reputation $\rho_{(s,t)} = \rho_{(s,t-1)} + e(A', \chi_{s'}, H_s)$, i.e. the new reputation value is to be derived from the previous reputation, adjusted by evidence e of the action set A', which is reported with a witness-specific judging confidence of $\chi_{s'}$, under consideration of the user's reputation history H_s, which serves as a smoothing factor to prevent too frequent automated changes of the user's privileges. The formula for calculating ρ, with t_0 being the first time when the principal requests access to resources, is weighted based on the number of transactions $N_{A'}$, i.e. the number of audited sets of actions within A', at the given point of time as follows:

$$\rho_{(s,t)} = \rho_{(s,t_0)} + \frac{\sum_{j=0}^{j=t-1} e(A'(j), \chi_{s'}(j), H_s(j))}{\sum_{k=0}^{k=t-1} N_{A'}(k)}$$

The collection of evidence statements e from reliable entities is based on our work in [3], which presented an algorithm to construct a trust graph and associate each edge with the appropriate trust level. The algorithm then searches the optimal path to the (previously unknown) subject, in such a way that more trusted edges dominate the result.

– *Trust by certificate chain*: The trust level derived from chained PKI certificates has proven to be helpful to gather more information about the requester when multiple a priori known entities already have had an arbitrary number of indirect relationships with the external principal. The requester presents a certificate, for example in X.509v3 format, that can be verified to ensure whether these credentials are signed by a third party related to an entity known in the DFE.

The process of trust quantification may use further optional parameters which are not used in this paper; an overview is given in [7] and [11].

Credential submission: Along with each request to perform an action on a resource, the subject must submit credentials. Unlike traditional access control, our approach enables the use of multiple credentials in a single request, which eliminates the necessity of negotiation protocols to determine the most suitable single credential. As typically used credentials, such as usernames/passwords and biometric profiles, do not work for unknown external users, we focus on assertions and recommendations from reliable entitites, such as SAML authentication assertions or WS-Security tokens.

Risk quantification: Resource owners must specify the risk levels of their resources. Given a risk calculation algorithm $risk$, the resource's risk level γ depends on the action to be performed at a certain point in time on the resource, but is independent of the user: $\gamma_{(r,a,t)} = risk(r, a, t)$

In real-world scenarios, each organization must define its own risk level assignment rules. Generally, they are based on legal and regulatory compliance responsibilities, the SLA impact if the resource federation rules are not met, and the threats resulting from unauthorized access. For our examples, we use a quartile-based approach, resulting in the trust levels *low*, *medium*, *high*, and *critical*.

3.2 Risk and trust based access decision workflow

Pseudo-code listing 1 demonstrates the workflow for balancing of trust and risk exemplarily using the four risk levels defined above. Trust level thresholds of 0, 0.5, and 0.9 are used for access to resources with *low*, *medium*, and *high* risk respectively. In this example, decisions are delegated to an external policy decision point in two cases:

1. If the request is made by an external user which is yet unknown. This allows to handle anonymous access or self-enrolment on a per-service basis.
2. If the risk is *critical* and the user is fully trusted; this adds another layer of resource-local access control and can be used to combine traditional access control mechanisms with TBAC, which is a typical prerequisite in real-world scenarios.

The listing also demonstrates the use of two additional hooks. First, if the decision is *deny*, the user can be notified about the reason why her access attempt failed. Second, the access control result of all requests is logged to a tamper-proof database, which can, for example, be used for auditing purposes.

4 Implementation and preliminary experiences

The implementation of this approach is built on top of our previous work on TBAC [2]. We extended the data model of the existing repositories to store the newly relevant trust and risk information. In this section, we first describe the LDAP schema extension

---

Algorithm 1 Exemplary trust and risk assessment

Input parameters:
Subject s, action a, resource r, condition set C,
subject's credential $Cred_s$, subject's action and resource specific trust level tl_s
resource's risk level $\gamma_{(r,a,t_{now})}$
Output parameter:
Access control decision, i. e. *permit* or *deny*
if $\exists s$ **then**
    return **assessAccess**$(tl_s, \gamma_{(r,a,t_{now})}, C, Cred_s, a, r)$
else
    // Set the default trust level for unknown users and delegate the decision
    $tl_s := -1$
    return **delegateDecision**$(tl_s, Cred_s, a, r)$
end if

function assessAccess$(tl_s, \gamma_{(r,a,t_{now})}, C, Cred_s, a, r)$:
access := *deny*   *// deny access by default*
if $(\forall c \in C : $**evaluateCondition**$(c) == true)$ **then**
    if (
    $(\gamma_{(r,a,t_{now})} == low$ **and** $tl_s \geq 0)$ **or**
    $(\gamma_{(r,a,t_{now})} == medium$ **and** $tl_s \geq 0.5)$ **or**
    $(\gamma_{(r,a,t_{now})} == high$ **and** $tl_s \geq 0.9)$) **then**
        access := *permit*
    end if
    / If the risk is critical, even fully trusted users may not access the resource without additional resource-local ruling */*
    if $(\gamma_{(r,a,t_{now})} == critical$ **and** $(tl_s == 1)$) **then**
        return **delegateDecision**$(tl_s, Cred_s, a, r)$
    end if
end if
if access == *deny* **then**
    notify$(C, Cred_s)$ *// notify user about rejection reason*
end if
log$(t_{now}, s, r,$access$)$
return access
end function

---

implemented for identity repositories to store trust information. Then, the RDF-based storage of risk related information will be presented. This implementation overview is concluded by an outline of the modified policy evaluation and decision workflow, as well as a discussion of possible future improvements.

4.1 Trust data representation in an LDAP Directory

Using LDAP for our implementation is an obvious choice, as trust information is glued to entities, and most identity management solutions, which use this data, are based on LDAP directory services. This approach thus avoids the necessity of additional data repositories, which reduces the complexity of our overall architecture. Furthermore, LDAP is a standardized request-/response-based protocol, so our implementation is independend of vendor-specific drivers, such as those required for relational database management systems.

Data in LDAP servers is structured hierarchically and typically represented as a tree. The nodes of this tree are objects with an arbitrary set of attributes; each object is identified by its distinguished name (DN), which reflects the path in the tree from the object to the root. As can be seen in figure 2, user objects include attributes such

as the user's name. For the management of external users, we added a new subtree `ou=ExternalUsers`; `ou` means *organizational unit* and is the standard structuring element for LDAP trees. To store the trust related data, we designed a new LDAP objectClass `trustData`. An arbitrary number of `trustData` objects can be assigned to each user by placing them as leafs in the LDAP tree beneath the corresponding user object.

Each `trustData` object has the following mandatory attributes, i. e. it cannot be created without specifying values for

- `trustCredentialType`: This attribute specifies the types of credentials which have been submitted by the user. As discussed above, this influences the trust level calculation.
- `trustCredential`: This attribute stores the submitted credentials. This is a structured data type (cp. [2]) which is stored BASE64-encoded in LDAP, similarly to other binary data types.
- `trustAction` and `trustResource`; they specify the policy targets this object shall be applied to.
- `trustLevel`, i. e. the current action- and resource-specific user trust level.

Additional, in LDAP terms so-called optional, attributes can be used to store further details about the access and reputation history as well as recommendation chains if the user has been introduces by other known entities.
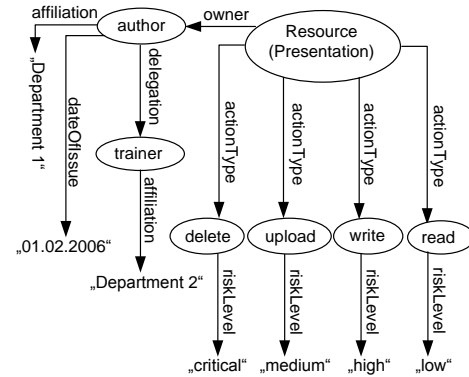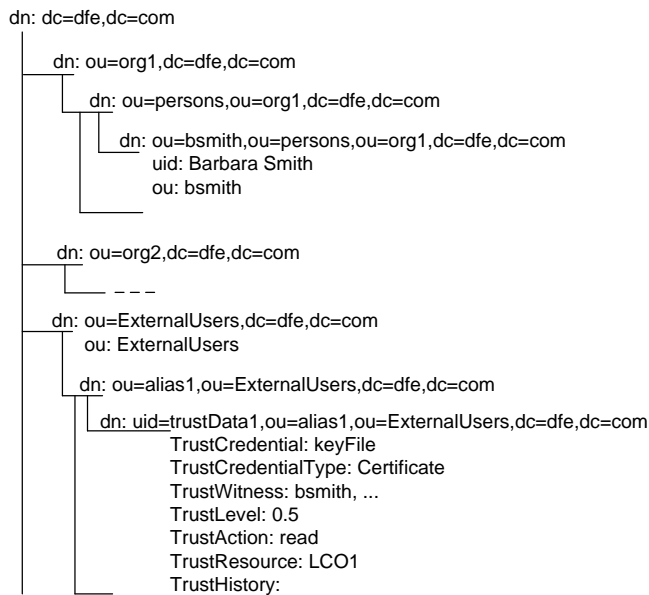
4.2 Resources risk description in RDF

The second part of our implementation realizes the representation of the different types of resources and services which are shared in the DFEs and may be accessed by external users; like the user profiles, this data is required during the policy evaluation process. Resource descriptions are related to several ITSM processes, such as configuration management, where risk specific resource attributes may be added to their respresentations, e. g. as configuration items in an ITIL CMDB. However, no widely deployed standards exist for this purpose, so our approach chooses to be generic by using RDF (Resource Description Framework, [9]), an XML-based language, for resource modelling.

In RDF, resources are identified by URIs and have *properties*, similar to LDAP attributes. These properties associate the resource either with values or with other resources, which in turn have their own properties. Resources are identified as nodes and properties are defined as directed, labeled edges, which are also known as RDF arcs.

Figure 2.b shows an exemplary application of our RDF model for describing resources with risk levels, based on the scenario described above: The resource, e. g. a set of presentation slides, has the property *actionType*. Each action that can be performed on the resource, such as *upload* or *delete*, is in turn associated with the appropriate risk level. Using this approach, the complete content of our eLearning system could be described, including the identity information, such as author profiles; in practice, this task must be automated due to the large number of objects.

dn: dc=dfe,dc=com

dn: ou=org1,dc=dfe,dc=com

dn: ou=persons,ou=org1,dc=dfe,dc=com

dn: ou=bsmith,ou=persons,ou=org1,dc=dfe,dc=com
uid: Barbara Smith
ou: bsmith

dn: ou=org2,dc=dfe,dc=com
_ _ _

dn: ou=ExternalUsers,dc=dfe,dc=com
ou: ExternalUsers

dn: ou=alias1,ou=ExternalUsers,dc=dfe,dc=com

dn: uid=trustData1,ou=alias1,ou=ExternalUsers,dc=dfe,dc=com
TrustCredential: keyFile
TrustCredentialType: Certificate
TrustWitness: bsmith, ...
TrustLevel: 0.5
TrustAction: read
TrustResource: LCO1
TrustHistory:

a.   Users data representation in a LDAP Directory

b.   Resource definition in RDF

Fig. 2. Representation of the data structure

Due to its distributed nature, the effective use of metadata among several organizations within the DFE requires common data semantics, syntax, and structure. Conflicts are efficiently prevented by defining a name space to avoid object name clashes between organizations and systems. The following listing illustrates a generic RDF description with appropriate name space specifications:

```
<?xml:namespace ns="http://www.w3.org/RDF/RDF/" prefix="RDF" ?>
<?xml:namespace ns="http://uri-of-name-space-1" prefix="DFE" ?>
<?xml:namespace ns="http://uri-of-name-space-n" prefix="NSn" ?>
..
<RDF:RDF>
  <RDF:Description RDF:HREF = "http://uri-of-Resource-1">
    <DFE:Property1>...</DFE:Property1>
    <DFE:Property2>...</DFE:Property2>
    ..
  </RDF:Description>
  ..
  <RDF:Description RDF:HREF = "http://uri-of-Resource-n">
    <NSn:Property1>...</NSn:Property1>
    <NSn:Property2>...</NSn:Property2>
    ..
  </RDF:Description>
</RDF:RDF>
```

4.3 Implementation of the access control model

Our implementation encompasses two major components that carry specific responsibilities, as shown in figure 3:

– `Trust Broker`: This component collects both, the relevant information about the requester for computing the prospective trust level, and the corresponding RDF definition of the requested resource. It also is the service access point for the user.
– `Policy Engine`: We have integrated the trust and risk assessment rules into a policy engine which processes the information collected from the trust broker and triggers a policy decision point (PDP) for the eXtensible Markup Access Control Language (XACML) [4]. The PDP decides solely based on the provided information, which also includes the relevant access control policies and environmental information such as the current date and time.
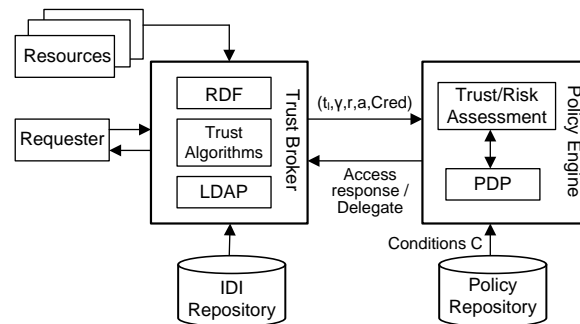

Fig. 3. Architectural overview

5 Related work and competitive approaches

This section gives a short overview of the related work and its influence on our approach.

Mayer et al. [8] complement the integration of security aspects in requirements engineering by adapting and integrating risk analysis in the iterative cycle of information system development. While this proposal strives to identify the existence of risk that affects the assets of IT systems, it does not assess the level of risk quantitatively. The same limitation is encountered in the approach of Lee et al. [10], which investigates the interactions between various models within their framework, and considers the relationships between security requirements and risk assessment. This framework investigates the mappings that exist between the security requirements enforced by the standard of the Department of Defense Information Technology Security Certification

and Accreditation Process (DITSCAP) [5] and the elements of risk assessment to drive a justifiable risk assessment process. However, this risk assessment process is exclusively bound to the DITSCAP ontological characteristics and lacks from establishing common-understanding risk metrics.

In the area of risk quantification, the SECURE project [6] worked on a framework that considers the trust in a principal as well as the risk for granting her request. The policy language used in SECURE uses a simple grammar, which is not sufficiently expressive to encode risk metrics. Similarly, [1] and [12] consider policy-driven decision making by evaluating trust and the impact of countermeasures; these two approaches make use of thresholding in their policy language for comparing the trust-values with a certain level of reliability. However, these threshold values are statically determined and fail to consider any run-time evaluation of trust and risk values, which obviously limits their flexibility.

6   Current Status and Next Steps

The overall goal of our research is to design a trust management framework for DFEs that enables the members to form, update, and exchange trust levels of external users. In this paper, we addressed issues of combining the trust information with the risk information in trust-based access control. Based on these results, our future work will discuss the delegation of trust decisions and its automation in more complex scenarios, for example when it is an invalid assumption that a chain of intermediate entities exists which can be contacted on demand to acquire reputation information about the unknown entity.

Furthermore, it will be part of our future work to explore topics related to keeping trust information up-to-date and accurate (e. g., ways to recover from a bad reputation when freshly obtained trust information reflects a considerable increase in the confidence) as well as run-time evaluation of risk parameters over the lifetime of the federated environment.

We are currently working on the design and formalization of the trust management framework that meets these additional requirements. Afterwards, we plan to evaluate the performance of our approach with respect to the promptness at which reputation information is collected, the accuracy of the obtained trust judgments as well as the adaptability of the model to the DFE member's distributed access control policies.

References

1. Bharat K. Bhargava and Yuhui Zhong. Authorization based on evidence and trust. In *DaWaK 2000: Proceedings of the 4th International Conference on Data Warehousing and Knowledge Discovery*, pages 94–103, London, UK, 2002. Springer-Verlag.
2. L. Boursas and V. Danciu. Dynamic inter-organizational cooperation setup in circle-of-trust environments. In *To appear in the Proceedings of the 20th IEEE/IFIP Network Operations and Management Symposium NOMS08*, Salvadore, Brazil, April 2008.
3. L. Boursas and H. Reiser. Propagating trust and privacy aspects in federated identity management scenarios. In *Proceedings of the 14th Annual Workshop of HP Software University Association (HPSUA 2007)*, Leibniz Supercomputing Center, Munich, Germany, July 2007.
4. T. M. (Editor). Oasis extensible access control markup language (xacml) 2.0, core specification. OASIS XACML Technical Committee Standard, 2005.
5. M. Mastrorocco J. Eller and B. Stauffer. *Department of Defense Information Technology Security Certification and Accreditation Process (DITSCAP)*. US-Department of Defense, July 2000.
6. C. D. Jensen. Secure environments for collaboration among ubiquitous roaming entities secure. In *First Internal iTrust Workshop on Trust Management in Dynamic Open Systems*, Glasgow, Scotland, sep 2002.
7. H.D. McKnight and N.L Chervany. The meanings of trust. Technical Report 94-04, Department Carlson School of Management, University of Minnesota, 1996.
8. N. Meyer, A. Rifaut, and E. Dubois. Towards a risk-based security requirements engineering framework. *Workshop on Requirements Engineering for Software Quality. In Proc. of REFSQ'05*, 2005.
9. Resource description framework (rdf). http://www.w3.org/RDF/.
10. R. Gandhi S. Lee and G. Ahn. Security requirements driven risk assessment for critical infrastructure information systems. In *SREIS'05*, 2005.
11. N. Shadbolt. A matter of trust. In *IEEE Intelligent Systems*, pages 2–3, February 2002.
12. Bin Yu and Munindar P. Singh. An evidential model of distributed reputation management. In *AAMAS '02: Proceedings of the first international joint conference on Autonomous agents and multiagent systems*, pages 294–301, New York, NY, USA, 2002. ACM.

Biography

Latifa Boursas received a diploma degree in Computer Engineering from the Universite des Sciences et de la Technologie (USTHB) of Alger, Algeria. She is a PhD candidate at the Technische Universität München (TUM), and is working in the project IntegraTUM at the Leibniz Supercomputing Center that deals with establishing a user-friendly and integrated information and communication infrastructure at TUM. Her research interests include Trust Management and Access Control in distributed federated environments.

Wolfgang Hommel has a PhD in computer science from Ludwig Maximilians University, Munich, and heads the identity management team at the Leibniz Supercomputing Center. His current research focuses on IT security/privacy management in large distributed systems, including identity federations and Grids.