Security and Trust Issues in Semantic Grids

Daniel Olmedilla[1], Omer F. Rana[2], Brian Matthews[3], and Wolfgang Nejdl[1]

[1] L3S Research Center and University of Hanover, Germany
{olmedilla,nejdl}@l3s.de
[2] School of Computer Science and Welsh eScience Center, Cardiff University, UK
o.f.rana@cs.cardiff.ac.uk
[3] CCLRC, UK
B.M.Matthews@rl.ac.uk

Abstract. Grid computing allows sharing of services and resources across institutions. However, current Grid security mechanisms for authentication and authorization are too rigid and they lack the ability to determine how "trustworthy" the result obtained from a specific provider is likely to be. This paper describes the different facets associated to Trust and identifies the need for Trust Management approaches in the context of Virtual Organizations lifecycle and resource access control in the Grid.

1 Introduction

A key theme in Grid computing is the capability to share services (representing different kinds of expertise) distributed across multiple institutions, and generally referred to as a "Virtual Organisation" (VO). When integrating services across multiple partners that form a VO, both security and trust issues become significant. The following questions may be used to describe the key requirements that arise:

- Case I: Do I allow user X to use my resources? If so, what requirements must be satisfied by such a user?
- Case II: Do I believe what Provider A says is true and factual?
- Case III: Do I agree with the answer that is provided by a Provider or Group?
- Case IV: Do I believe that a Provider or Group goals and/or priorities match mine?

From these requirements, we can see that Case I relates to the issue of security credentials and policy protection – i.e. whether a particular service provider allows an external user to make use of its resources. Significant work has already been undertaken within the Grid community in this area – generally through the use of X509-based digital certificates to verify identity. Cases II, III and IV refer to issues that are not directly related to security – as they involve a more deeper understanding of the interactions taking place between a provider and user.

Current trend in Grid Security Infrastructure (GSI) is to enable trust relationships to be established in the Grid community – generally through the use of X509-based digital certificates, and more recently, through the use of security assertions (SAML) and role-based access management (PERMIS and Shibboleth). However, such security mechanisms still do not scale. Among the existing problems we can identify mechanisms that are too rigid for authentication and authorization, in terms of access control (Case I), and they lack the ability to determine how "trustworthy" the result obtained from a specific provider is likely to be. Therefore, generally they are not able to address Cases II, III and IV identified above.

Trust management provides us with the basis to overcome these points of view. However, the general notion of "trust" is excessively complex, and appears to have many different meanings depending on how it is used. Trust is seen as a multifaceted issue and may be related to other themes such as risk, competence, security, beliefs and perceptions, utility and benefit, and expertise. In addition, policy-based trust management is understood as statements guiding a process where two strangers are able to commit a specific transaction. Therefore, the aim of this paper is to identify the advantages/uses/requirements/threads of applying trust on Grid computing from the following two complementary points of view:

- Access Control: some of the existing problems can be addressed by extending Grid Security Infrastructure with trust negotiation mechanisms. These extensions, can provide the Grid with property-based authorization mechanisms, automatic gathering of required certificates, bidirectional and iterative security negotiations and policy based authorizations.
- Provision of Service: the notion of trust transcends beyond the restrictive security issues that are currently being explored in the Grid community. Where security issues are primarily concerned with ensuring that the result being provided is to come from a traceable source, trust issues can also relate to the degree of belief a user has in a particular provider, and is therefore much more subjective in nature. Some questions to be answered would be: "Can a trust rating be associated with a Grid Service? If so, how is this calculated and what does it mean?"

2 Related Work

The term *trust management*, introduced in [6] as "a unified approach to specifying and interpreting security policies, credentials, and relationships which allow direct authorization of security-critical actions", has been given later a broader definition, not limited to authorizations [15]: "Trust management is the activity of collecting, encoding, analyzing and presenting evidence relating to competence, honesty, security or dependability with the purpose of making assessments and decisions regarding trust relationships". Two main approaches are currently available for managing trust: policy-based and reputation-based trust management.

2.1 Policy-based Trust Management

This approach has been proposed in the context of open and distributed services architectures [7, 20, 14, 5, 8] as well as in the context of Grids [4] as a solution to the problem of authorization and access control in open systems. The focus here is on trust management mechanisms employing different policy languages and engines for specifying and reasoning on rules for trust establishment. The goal is to determine whether or not an unknown user can be trusted, based on a set of credentials and a set of policies.

In addition, it is possible to formalize trust and risk within rule-based policy languages [25, 16] in terms of logical formulae that may occur in rule bodies.

Currently, policy-based trust is typically involved in access control decisions. Declarative policies are very well suited to specifying access control conditions that are eventually meant to yield a boolean decision (the requested resource is either granted or denied). Systems enforcing policy based trust typically use languages with well-defined semantics and make decisions based on "non-subjective" attributes (e.g., requester's age or address) which might be certified by certification authorities (e.g., via digital credentials). In general, policy-based trust is intended for systems with strong protection requirements, for systems whose behavior is guided by complex rules and/or must be easily changeable, as well as for systems where the nature of the information used in the authorization process is exact.

2.2 Reputation-based Trust Management

This approach has emerged in the context of electronic commerce systems, e.g. eBay. In distributed settings, reputation-based approaches have been proposed for managing trust in public key certificates, in P2P systems, mobile ad-hoc networks, and, very recently, in the Semantic Web. The focus here is on trust computation models capable to estimate the degree of trust that can be invested in a certain party based on the history of its past behavior.

The main issues characterizing the reputation systems are the trust metric (how to model and compute the trust) and the management of reputation data (how to securely and efficiently retrieve the data required by the trust computation) [2].

Marsh [22] made one of the early attempts at formalizing trust using simple trust metrics based on linear equations. This model has been further extended by Abdul-Rahman and Hailes to address reputation-based trust in virtual communities [1]. A number of reputation mechanisms for P2P systems, such as [2, 10, 18, 12], followed similar trust and reputation models.

Typically, reputation-based trust is used in distributed networks where a system only has a limited view of the information in the whole network. New trust relationships are inferred based on the available information (following the idea of exploiting world's information). In these scenarios, the available information is based on the recommendations and the experiences of other users, and it is typically not signed by certification authorities but (possibly) self-signed by the

source of the statement. This approach supports trust estimates with a wide, continuum range and allows the propagation of trust (e.g., transitive propagation) along the network as well as weighting of values (e.g., fresher information vs. older information).

3 Security Issues of Relevance

Grid environments provide the middleware needed for access to distributed computing and data resources. Distinctly administrated domains form virtual organizations and share resources for data retrieval, job execution, monitoring, and data storage. Such an environment provides users with seamless access to all resources they are authorized to. In current Grid infrastructures, in order to be granted access at each domain, user's jobs have to secure and provide appropriate digital credentials for authentication and authorization. However, while authentication along with single sign-on can be provided based on client delegation of X.509 proxy certificates [26] to the job being submitted, the authorization mechanisms are still mainly identity based. Due to the large number of potential users and different certification authorities, this leads to scalability problems calling for a complementary solution to the access control mechanisms specified in the current Grid Security Infrastructure (GSI) [17].

Current authentication and authorization mechanisms in the Grid are too rigid and do not scale. For example, there are some general implicit assumptions which are necessary for a submitted job to succeed:

- One credential for all. A job is submitted together with a proxy certificate signed by a Certification Authority (CA) which is further used to authenticate to other resources. This assumes that all resources must trust such a CA.
- Identity Based Authorization. Resources, where a job is allowed access, have to know in advance the identity of the certificate.
- Simple Authentication/Authorization. It is based on a one-shot process where the job requests access and the resource grants or denies it.
- Manual Credential Fetching. Users need to find out in advance which credentials are required to access each resource and program their jobs to fetch and give these credentials while authenticating/requiring authorization.

Some of them have recently been subject of research in order to relax them. Still, these requirements become liabilities when the Grid grows more complex and the number of resources a job has to access increases. Mapping identities raises serious scalability problems due to the large number of potential users, even more when we take into account the difficult requirement of having a single trusted Certification Authority (which is hard enough to have within one Grid and not feasible when trying to integrate different Grids). Because of these reasons, property based certificates have started to appear (PRIMA [21], VOMS [3], CAS [24] and X.509 attribute certificates [13]) even though there is no standard interface for using them yet. Furthermore, access control is not a simple task, as

both a job and a resource might want to specify constraints in the way they disclose their certificates. This asks for extension of the usual one-shot mechanism to an iterative process where the level of trust increases at each iteration [27]. Finally, resources should advertise the credentials they require thus allowing the job to perform dynamic credential fetching. This not only frees job owners of "what resource requires what credential" problem [1] or that of coding credential fetching within their jobs but also allows dynamic selection of resources as they would be self explanatory in terms of authorization requirements.

We argue that Grid with support for specifying, advertising and enforcing service level access control policies together with capabilities for automatic fetching of credentials can indeed suit large scale collection of resources, enabling dynamic negotiation for authorization and access granting based on parties properties.

Therefore, we propose a scheme in which grid services advertise their authorization requirements as access control policies, clients are able to query for these policies and fulfill them through automatic credential fetching. Credentials can be protected by policies that have to be satisfied by the requesting part before having the credential revealed. Acting in a self describing environment, services and clients automatically negotiate authorization by iteratively increasing their trust relationship through credential disclosure until a decision regarding authorization can be made. This approach can be easily implemented e.g. on top of the Globus Toolkit 4.0 [9].

4 Trust Issues of Relevance

The general notion of "trust" is excessively complex, and appears to have many different meanings depending on how it is used. There is also no consensus in the computer and information sciences literature on a general definition of "trust" - although its importance has been widely recognized in the increasing number of publications that utilize it. Using trust as a basis for decision making has been widely cited by many authors however, and this is the primary motivation on which we base our definition. We may therefore define trust as:

Definition 1. *Trust of a party A to a party B for a service X is the measurable belief of A in that B behaves dependably for a specified period within a specified context (in relation to service X)*

According to this definition, trust is evaluated relative to a specific service. In the simplest case, we may assume that a service provider only offers one service. Hence, one may associate service trust with provider trust. In cases where a service provider offers more than one service, it may be necessary to find some mean

[1] Currently, users must keep track of the right/required credential for each resource he might access. Application needs may be difficult to predict prior to its execution and the user might not be available when a certain credential is required. On the other hand providing the application with all user credentials is not feasible as it may imply revealing sensitive information.

value of trust, based on each of the services offered. It is also important to note that different trust relationships appear in different business contexts – hence, evaluating a reliability context may be different from a performance/execution time context. The measurement of trust may be absolute (e.g. a probability) or relative (e.g. dense order). Trust must be evaluated with reference to a 'period' – which may be in the past (history), the duration of the service (from now until the end of a service), future (a scheduled or forecasted critical time slot), or always. Dependability is deliberately understood broadly to include aspects such as safety, reliability, timeliness, maintainability. However, some of these aspects are easier to evaluate than others – for instance, reliability can be measured, whereas safety cannot.

Trust is therefore a multifaceted issue and may be related to other themes such as risk, competence, security, beliefs and perceptions, utility and benefit, and expertise. Hence, a service user may only be interested in evaluating the trust of a service provider if there is likely to be some risk to the service user directly. Overall, two main approaches may be deduced from literature. The first is based on allowing "agents" in a system to trust each other and therefore there is a need to endow them with the ability to reason about the reliability or honesty of their counterparts. This ability is captured through trust models. The second is based on allowing agents to calculate the amount of trust they can place in their interaction partners. This is achieved by guiding agents in deciding how, when and who to interact with. An agent in this context refers to either a service user or a provider. However, in order to do so, trust models initially require agents to gather some knowledge about their counterparts. This may be achieved in three ways:

1. A Presumption drawn from the agent's own experience: trust is computed as a rating of the level of performance of the trustee. The trustee's performance is assessed over multiple interactions to check how good and consistent it is at doing what it says it will [1,12]. This aspect of trust may utilize a pre-agreed contract between a service user and provider. Violation of a contract is likely to impinge on the trust that the service user has in the provider. This is also, currently, the predominant approach applied in existing Grid computing projects. A Service Level Agreement (SLA) is established between a service user and a provider, and trust is then evaluated with reference to such an SLA.
2. Information gathered from other agents: trust in this approach is computed indirectly from recommendations provided by others. As the recommendations could be unreliable, the agent must be able to reason about the recommendations gathered from other agents. The latter is achieved in different ways: (1) deploying rules to enable the agents to decide which other agents' recommendation they trust more; (2) weighting the recommendation by the trust the agent has in the recommender [19] [23]. Such a referral mechanism may involve a multi-hop interaction.
3. Socio-Cognitive Trust: trust in this case is the capability to characterize the likely motivations of other agents. This involves forming coherent beliefs

about different characteristics of these agents, and reasoning about these beliefs in order to decide how much trust should be put in them [11].

Client agents involved in a Grid system may need to choose between a set of partners to interact with. This situation arises when a client requires a service, and multiple providers are available to offer such a service. Selection between such a set of providers incurs a computational cost – which may increase as the number of interactions and service execution requests increase. We therefore define the concept of a "community", within which participants can interact with a higher level of trust on each other. In this instance, trust may be viewed with reference to each of the three criteria mentioned above. A reason for the formation of such communities may be to reduce the subsequent cost of interaction once the community has been established. An agent may therefore decide to incur an initial cost to determine which community it should participate in, what actions it should undertake within the community (its role), which other participants it should communicate with (its interactions), and when to finally depart from the community. Based on such an analysis, an agent would have to pay an initial cost to make some of these decisions. Subsequently, the agent will only incur an "operational" cost – much lower than that for making some of these initial decisions. Formation of such communities may be implicit - i.e. based on analysing interactions of agents and determining the level of trust that can be placed on other participants, or explicit - i.e. a system administrator may determine which set of agents must be placed in a community.

4.1 Trust Lifecycle

To better understand trust issues within VOs, we relate specific instances where trust arises with reference to a VO lifecycle – which includes:

- Phase I: Service (Provider) Identification
- Phase II: Formation and Service (Provider) Invitation
- Phase III: Operation and Service (Provider) Interaction
- Phase IV: Dissolution

In Phase I, the service providers that need to interact are identified. It is generally assumed that this is undertaken through a manager entity – who is forming the VO in order to underake a particular activity. In Phase II, the identified providers are asked to join the VO. This phase may involve negotation between the manager entity and the providers (or directly between the providers) to ensure that a Service Level Agreement (SLA) is established between the entity and each provider (or directly between the providers). In Phase III, the providers interact to perform the particular activity desired by the manager entity. In Phase IV, the VO is disbanded. We can now associate particular trust and security issues within such a VO lifecycle, and this is illustrated in Table 1.

The trust lifecycle therefore involves the following operations – in relation to a VO lifecycle:

VO Lifecycle Phase	*Trust and Security*	*Service Level Agreement*	*Collaborative Process*
Phase I	Policy Specification. Verify Participant Credentials	Identify SLA Requirements	Define VO Objectives
Phase II	Evaluate Trust using Reputation Repositories	Negotiate SLA(s)	Evaluate Community Trust and Provision Resources
Phase III	Trust Maintenance	Monitor SLA Record SLA Violations	Trust based Service Invocation
Phase IV	Terminate Trust Relationship. Publish data into Reputation Repository	Update SLA	Disengage Resources

Table 1. Trust Lifecycle

- Phase I: A policy specification is used to determine the type of participants that should be allowed to participate in a VO. For each discovered participant, the credentials of the participant are checked – generally by verifying the digital certificate (and certificate issuing authority) for the participant. The SLA requirements for each participant are identified (i.e. the type of terms which should constitute the SLA), based on the type of service the participant is providing within the VO. The SLA is also defined with reference to the overall objectives of the VO – for instance, the VO manager entity may generate subcontracts for each VO participant depending on the overall activity for which the VO is being formed.
- Phase II: If no prior experience about interacting with a particular VO participant exists, scores from a Reputation Repository may be used to assign a trust value to each participant in the VO by the manager entity. Based on these trust values, an SLA is negotiated between entities within the VO. The manager entity (or each participant) may evaluate the success of fulfilling SLA obligations based on the trust score assigned to a particular provider. Resource provision can then be undertaken based on the negotiated SLAs. For instance, of a manager entity has a low trust score for a particular provider, it may be necessary to request over provision of a particular resource offered by such a provider.
- Phase III: During the operating phase of the VO, the SLAs that have been negotiated are monitored. Any violation is flagged to the provider, and recorded for later update of the reputation repository. The type of monitoring tool used is dependent on the parameters identified in Phase I above.
- Phase IV: Feedback from SLA violations – obtained from Phase III are now recorded for use by other providers. The SLA established is now formally terminated, and resources associated with the SLA are released. It is also possible at this stage to utilise a reasoning engine to evaluate feedback from Phase III, and generate feedback that covers multiple providers for use by the manager entity.

5 Reputation Mechanisms

Whereas trust is generally seen from the perspective of a service user towards a service provider (or vice versa), we may be "Reputation" as the collective view of a group of users/providers towards another user/provider. Reputation is therefore the aggregate trust that a group of agents reveal about other agents.

Reputation mechanisms can be compared with those found in electronic commerce systems, such as `ebay.com` or `amazon.com`, where users can leave feedback about a particular seller (in `ebay`) or about a particular product (in `amazon`). Such existing mechanisms do not provide any means to verify the accuracy of such reputation scores, often allowing one agent to assign multiple (sometimes conflicting) scores for a particular provider. Nevertheless, the availability of such scores can provide a useful basis for selecting between providers offering similar services. The exact incentive structure for an agent to provide such openly accessible reputation scores needs further work.

5.1 Reputation Repositories

A Reputation Repository is a store that contains aggregate scores for a particular service (or provider). If such a score is assigned to a service, it is necessary that some persistent identity exists for it. Hence, it would not be possible to assign a reputation score to a service instance, only to the algorithm (class file, executable, etc) from which the service instance has been generated. Hence, the score provided by a reputation repository is useful only if an agent can make decision for using the service/provider. Such a repository must be managed by a trusted third party (i.e. different from the service user and provider agents). This is similar to the availability of a certificate authority, able to verify the identity of a user and generate a digital certificate.

6 Conclusion

The term trust has usually different meanings depending on the context in which it is used. However, two main approaches to Trust Management exist: policy-based trust management and reputation-based trust management. This paper describes both approaches and identifies, based on four questions derived from usage of services within a VO, the motivation for evaluating security and trust issues within a Virtual Organization in Grid environments. This paper also shows how managing trust arises in the context of Virtual Organizations lifecycle and addresses current limitations regarding to resource protection and access control in the Grid.

References

1. A. Abdul-Rahman and S. Hailes. Supporting trust in virtual communities. In *Proceedings of 33rd Hawaii International Conference on System Sciences*, 2000.

2. K. Aberer and Z. Despotovic. Managing trust in a peer-2-peer information system. In *Proceedings of 10th International Conference on Information and Knowledge Management*, pages 310–317, 2001.
3. R. Alfieri, R. Cecchini, V. Ciaschini, L. dell'Agnello, A. Frohner, A. Gianoli, K. Lőrentey, and F. Spataro. Voms: An authorization system for virtual organizations. In *Proceedings of the 1st European Across Grids Conference*, Santiago de Compostela, Feb. 2003.
4. J. Basney, W. Nejdl, D. Olmedilla, V. Welch, and M. Winslett. Negotiating trust on the grid. In *2nd WWW Workshop on Semantics in P2P and Grid Computing*, New York, USA, may 2004.
5. M. Y. Becker and P. Sewell. Cassandra: distributed access control policies with tunable expressiveness. In *5th IEEE International Workshop on Policies for Distributed Systems and Networks*, Yorktown Heights, June 2004.
6. M. Blaze, J. Feigenbaum, and J.Lacy. Decentralized trust management. In *Proceedings of IEEE Conference on Security and Privacy*, 1996.
7. P. Bonatti and P. Samarati. Regulating service access and information release on the web. In *CCS '00: Proceedings of the 7th ACM conference on computer and communications security*, pages 134–143. ACM Press, 2000.
8. P. A. Bonatti and D. Olmedilla. Driving and monitoring provisional trust negotiation with metapolicies. In *6th IEEE International Workshop on Policies for Distributed Systems and Networks (POLICY 2005)*, pages 14–23, Stockholm, Sweden, jun 2005. IEEE Computer Society.
9. I. Constandache, D. Olmedilla, and W. Nejdl. Policy based dynamic negotiation for grid services authorization. Technical report, L3S Research Center, 2005.
10. E. Damiani, S. D. C. di Vimercati, S. Paraboschi, P. Samarati, and F. Violante. A reputation-based approach for choosing reliable resources in peer-to-peer networks. In *Proceedings of ACM Conference on Computer and Communications Security*, pages 202–216, 2002.
11. R. de Lemos. Architecting web services applications for improving availability. In *Architecting Dependable Systems III*, volume 3549 of *Lecture Notes in Computer Science*. Springer, 2005.
12. C. Duma, N. Shahmehri, and G. Caronni. Dynamic trust metrics for peer-to-peer systems. In *Proceedings of 2nd IEEE Workshop on P2P Data Management, Security and Trust (in connection with DEXA'05)*, August 2005.
13. S. Farrel and R. Housley. An internet attribute certificate profile for authorization, rfc3281.
14. R. Gavriloaie, W. Nejdl, D. Olmedilla, K. E. Seamons, and M. Winslett. No registration needed: How to use declarative policies and negotiation to access sensitive resources on the semantic web. In *1st European Semantic Web Symposium (ESWS 2004)*, volume 3053 of *Lecture Notes in Computer Science*, pages 342–356, Heraklion, Crete, Greece, may 2004. Springer.
15. T. Grandison. *Trust Management for Internet Applications*. PhD thesis, Imperial College London, 2003.
16. T. Grandison and M. Sloman. Specifying and analysing trust for internet applications. In *Towards The Knowledge Society: eCommerce, eBusiness, and eGovernment, The Second IFIP Conference on E-Commerce, E-Business, E-Government (I3E 2002)*, IFIP Conference Proceedings, pages 145–157, Lisbon, Portugal, oct 2002. Kluwer.
17. http://www.globus.org/security/overview.html. *Grid Security Infrastructure.*

18. S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina. Eigenrep: Reputation management in p2p networks. In *Proceedings of 12th International WWW Conference*, pages 640–651, 2003.
19. S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina. The eigentrust algorithm for reputation management in p2p networks. In *Proceedings of 12th International WWW Conference*, 2003.
20. N. Li and J. Mitchell. RT: A Role-based Trust-management Framework. In *DARPA Information Survivability Conference and Exposition (DISCEX)*, Washington, D.C., Apr. 2003.
21. M. Lorch, D. Adams, D. Kafura, M. Koneni, A. Rathi, and S. Shah. The prima system for privilege management, authorization and enforcement in grid environments. In *Proceedings of the 4th Int. Workshop on Grid Computing - Grid 2003*, Phoenix, AZ, USA, Nov. 2003.
22. S. Marsh. *Formalising Trust as a Computational Concept*. PhD thesis, University of Stirling, 1994.
23. L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford Digital Library Technologies Project, 1998.
24. L. Pearlman, V. Welch, I. Foster, C. Kesselman, and S. Tuecke. A community authorization service for group collaboration. In *Proceedings of the IEEE 3rd International Workshop on Policies for Distributed Systems and Networks*, 2002.
25. S. Staab, B. K. Bhargava, L. Lilien, A. Rosenthal, M. Winslett, M. Sloman, T. S. Dillon, E. Chang, F. K. Hussain, W. Nejdl, D. Olmedilla, and V. Kashyap. The pudding of trust. *IEEE Intelligent Systems*, 19(5):74–88, 2004.
26. V. Welch, I. Foster, C. Kesselman, O. Mulmo, L. Pearlman, S. Tuecke, J. Gawor, S. Meder, and F. Siebenlist. X.509 proxy certificates for dynamic delegation. In *3rd Annual PKI R&D Workshop*, Apr. 2004.
27. W. H. Winsborough, K. E. Seamons, and V. E. Jones. Automated trust negotiation. DARPA Information Survivability Conference and Exposition, IEEE Press, Jan 2000.