On Trust Management in Grids

Invited Paper

Alvaro Arenas, Michael Wilson, Brian Matthews
E-Science Centre
STFC Rutherford Appleton Laboratory
Harwell Science and Innovation Campus, Didcot, UK
{A.E.Arenas, M.D.Wilson, B.M.Matthews}@rl.ac.uk

***Abstract*—This paper presents an overview of the different concepts and technologies for managing trust in Grids. It examines the relation between trust and security, introducing the current technology for managing trust. The classical Virtual Organisation lifecycle is augmented with trust management actions.**

Grids; Grid Security; Trust Management; Virtual Organisation

I. INTRODUCTION

A Grid system is a scalable and autonomous infrastructure concerned with the integration, virtualisation and management of services and resources in a distributed, heterogeneous environment that support collections of users and resources (Virtual Organisations – VOs) across traditional administrative and organisational domains. The Grid was initiated as a way of supporting scientific collaboration, where many of the participants knew each other. In this case, there is an implicit trust relation, all partners have a common objective -for instance to realise a scientific experiment- and it is assumed that resources would be provided and used within some defined and respected boundaries. However, when the Grid is intended to be used for business purposes, it is necessary to share resources with unknown parties. Such interactions may involve some degree of risk since the resource user cannot distinguish between high and low quality resource providers on the Grid. The inefficiency resulting from this asymmetry of information can be mitigated through trust mechanisms.

This paper presents an overview of the different concepts and technologies relevant to trust management in Grid systems. It extends previous work on analysing trust and security in Grids [4]. Section II studies the relation between trust and security in distributed systems, and Grids in particular. Next, section III describes the existing mechanisms for managing trust. Next, section IV presents how the classical VO lifecycle can be extended to include trust management. Then, section V shows some examples of trust management systems for Grids. Finally, section VI concludes the paper by summarising the main results.

Autonomics October 28-30, Rome, Italy

II. RELATING TRUST AND SECURITY

This section analyses the concept of trust and its relation with security. There is a vast source of information on the theory and application of trust. For further information, we refer the reader to [4] [22].

A. *Trust Definitions*

In the context of networked and distributed computing systems, remote system needs to be trusted as well as interactions over underlying services such as communication services. As expressed by Grandison and Sloman [12], the significance of incorporating trust in distributed systems is that trust is an enabling technology. Its inclusion will enable secure electronic transactions.

There is not consensus in the literature on what trust is; it is recognised as an important and complex subject relating honesty, truthfulness, competence, reliability, etc. of the trusted person or service.

One of the influential works towards a practical definition of trust is given by Gambetta [11]: "*When we say we trust someone or that someone is trustworthy, we implicitly mean that the probability that he will perform an action that is beneficial or at least not detrimental to us is high enough for us to consider engaging in some form of cooperation with him. Correspondingly, when we say that someone is untrustworthy, we imply that that probability is low enough for us to refrain doing so*." Gambetta's definition stresses that trust is fundamentally a belief or estimation, which has inspired the use of subjective logic as a way of measuring trust [13].

The influential work by Grandison and Sloman [12] surveys various definitions of trust. Following a brief analysis of these definitions, they build their own one as "*the firm belief in the competence of an entity to act dependably, securely and reliably within a specified context*". They argue that trust is a composition of many different attributes - reliability, dependability, honesty, truthfulness, security, competence and timeliness - which may have to be considered and defined depending on the environment in which trust is being specified.

Dimitrakos [10] has defined trust as follows: "*Trust of a party A in a party B for a service X is the measurable belief of A in B behaving dependably for a specified period within a specified context in relation to X*". In his definition, a party can be an individual entity, a collective of humans or processes, or a system; the term service is used in a deliberately broad sense to include transactions, recommendations, issuing certificates, underwriting, etc; dependability is used broadly to include security, safety, reliability, timeliness, and maintainability; a period may be the duration of the service, refers to the past, future (a scheduled or forecasted critical time slot), or always; finally, the term context refers to the relevant service agreements, service history, technology infrastructure, legislative and regulatory frameworks that may apply.

Some aspects of these definitions are common, other are complementary. For example, Gambetta [11] emphasises that trust is in part subjective, a characteristic present in other definitions such as [12][10]. Grandison [12] underlines that trust is a belief in the competence of an entity within a specified context. One entity may trust another entity for one specific business and not in general. Such business relationship can be seen as the context of Gradinson's definition. Finally, the definition in Dimitrakos [10] highlights an important point; trust evolves in time and is measurable.

B. Trust and Security

In general, the purpose of security mechanisms is to provide protection against malicious parties. Traditional security mechanisms typically protect resources from malicious users by restricting access to only authorised users. However, in many situations within distributed applications, one has to protect oneself from those who offer resources so that the problem is in fact reversed. For instance, a resource providing information can act deceitfully by providing false or misleading information, and traditional security mechanisms are unable to protect against this type of threat. As noted in [14], trust systems can provide protection against such threats. The difference between these two approaches to security was first described by Rasmusson and Janssen in [20] who used the term *hard security* for traditional mechanisms like authentication and access control, and *soft security* for what they called social control mechanisms, of which trust is an example.

Grandison and Sloman [12] have defined a trust classification as a useful way of categorising the literature relating to trust in Internet services. We have found such taxonomy helpful in linking trust and security for the purpose of this work. Trust is specified in terms of a relation between a trustor, the subject that trusts a target entity, and a trustee, the entity that is trusted. Grandinson defines the following classes of trust.

1) Service Provision Trust

This type of trust describes the relying party's trust in a service or resource provider. The trustor trusts the trustee to provide a service that does not involve access to the trustor's resources. This type of trust is essential for Grids, and can be seen as a minimal trust requirement in dynamic Virtual Organisations (VOs). Many Grid applications assume this type of trust implicitly; a partner in a VO presupposes a service provision trust as a result of participating in VO, although the VO does not provide mechanisms to enforce it.

In general, service provision trust is related to the reliability or the integrity of the trustee. For instance, in e-banking the customer trusts the bank to support mechanisms that will ensure that passwords are not divulgated, and to maintain the privacy of any information such as name, address and credit card number. The Liberty Alliance Project uses the term "business trust" to describe a provision trust, a mutual trust between companies emerging from contract agreements that regulate interaction between them [18]. Mobile code and mobile agent-based applications also include service provision trust; the mobile code trusts the execution environment provided by the remote system [9].

2) Resource Access Trust

Resource access trust describes trust in principals for the purpose of accessing resources owned by the relying party. A trustor trusts a trustee to use resources that he own or controls. Resource access trust has been the focus of security research for many decades [1], particularly on mechanisms supporting access control. Generally, resource access trust forms the basis for specifying authorisation policies, which then are implemented using access control mechanisms, firewall rules, etc.

Grandison [12] highlights the distinction between trusting an entity to read or write a file on your server and trusting an entity to execute code within your workstation. Simple file access requires that the trustee will follow the correct protocol, will not divulge information read, and will write only correct data, etc. Allowing an entity to execute code on your workstation implies much higher level of trust. The code is expected not to damage the trustor's resources, to terminate within reasonable finite time and not to exceed some defined resource limits with respect to memory, processor time, local file space, etc.

3) Delegation Trust

This type of trust denotes the case when a trustor trusts a trustee to make decisions on his behalf, with respect to a resource or service that the trustor owns or controls. Although delegation is conceptually simple, designing and deploying it within a Grid environment has proved to introduce problems regarding security. Such security implications have been analysed by Broadfoot and Lowe in [6]. A point that is addressed is the level of trust assumed when delegation is employed, in particular the effect of having onwards delegation.

4) Certification Trust

This type of trust is based on the certification of the trustworthiness of the trustee by a third party, so trust would be based on a criteria relating to the set of certificates presented by the trustee to the trustor. Trust systems that derive certification trust are typically authentication schemes such as X.509 and PGP [28]. This class of trust is called *authentication trust* in Liberty Alliance [18] and *identity trust* by Josang in [14]. Grandison [12] views certification trust as a special form of service provision trust, since the certification authority is in fact providing a trust certification service; however Josang [14]

views certification trust and service provision trust as two layers on top of each other, where provision trust normally cannot exist without certification trust; in the absence of certification trust, it is only possible to have a baseline provision trust in an entity.

Certification trust has played an important role in Grid environments; it is present with the inclusion of certification authorities, which play a central role in the Grid Security Infrastructure [25] and has been exploited in production grids.

5) Context Trust

Finally, context trust describes the extend to which the relying party believes that the necessary systems and institutions are in place in order to support the transaction and provide a safety net in case something should go wrong. It refers to the base context that the trustor must trust. This type of trust is called *infrastructure trust* in [12], here we prefer to use the broader term of context trust used by [14], which also involves social and legal factors such as insurance and legal system and law enforcement.

C. Trust and Reputation

The concepts of trust and reputation are closely related. According to [1], reputation is an expectation about an agent's behaviour based on information about or observations of its past behaviour Reputation can be considered as a measure of trustworthiness, based on the referrals or ratings from members in a community.

Several properties should be taken into consideration when selecting a reputation model for distributed system, in particular grids [22].

1) The Computational Model.

Because grids are based on a distributed computational model, the first property of interest is if the trust mechanism is centralized or decentralized. In classical grids, where security is achieved through certificates and central certification authorities exist, a centralized model could be of interest. In such systems, a reputation service could be interrogated about the reputation of a user or, more generally, of a resource. The reputation service in this case serves as a point of centralization. Centralized models have the disadvantage of a single-failure point; therefore in some grids such as desktop grids, decentralized systems would be preferable.

2) Metrics for Trust and Reputation.

When referring to a metric for trust and reputation we consider the value that express the reputation (and trust) of an entity as provided by the reputation mechanism. We must make a distinction between the reputation value of an agent and the feedback one is required to provide at the end of a transaction. Continuous metrics are considered more expressive than discrete ones. Usual, these values are scaled between -1 and 1, or between 0 and 1. If the reputation scheme uses values scaled between 0 and 1 these values can have the meaning of a probability.

3) Reputation Feedback.

Reputation information might be positive or negative one. Some systems are based on collecting both type of information with regard to an entity, while other systems are based only on negative/positive information. Regarding an accomplished transaction, the reviewer can supply with binary, discrete or continuous values. Again, continuous values are more expressive but for the sake of simplicity, a lot of approaches use discrete feedback and later on aggregates this feedback in continuous reputation or trust. Ideas of utility computing could be used to generate feedback on transactions as presented in [21].

III. Technologies for Trust Management

Trust Management Systems (TMS) are responsible for managing trust in a distributed environment. This section describes the main technologies to achieve the type of trust presented in section B.

A. Certification Trust

One of the technologies playing a central role in certification trust is Public Key Infrastructure (PKI), which defines message formats and protocols that allow entities to securely communicate claims and statements. The most used assertions are those that bind identity and attributes statements to keys. The most popular PKI is defined by the IETF's PKIX working group, which defines a security system used for identifying entities (users and resources) through the use of X.509 identity certificates. In this PKI, highly trusted entities know as certificate authorities (CA) issue X.509 certificates where essentially a unique identity name and the public key of an entity are bound through the digital signature of that CA.

One of the challenges encountered in key management include the need of users of having different credential, since users may play different roles or be part of several projects which have elected to trust different CAs. While PKI could handle this situation by signing the same public key into several different certificates, in practice the user may end up with numerous key pairs to manage. To link these different identities, the notion of federated identities has been developed, as shown in the Liberty Alliance project [18].

Revocation is vital for authentication, for example when a key is compromised or when a user's project ends. PKI relies upon the periodic distribution of Certificate Revocation Lists (CRLs) in order to allow those relying upon certificate to gain confidence in their present validity. The use of CRLs needs careful management, particularly in relation to the frequency of updates.

B. Resource Access Trust

A good description of the current state of resource access trust in Grid computing appears in [8]. There are several architectural proposals for handling authorisation in Grids. One of the earliest attempts at providing authorisation in VOs was in the form of the Globus Toolkit Gridmap file. This file simply holds a list of the authenticated distinguished names of the Grid users and the equivalent local user account names that they are to be mapped into. Access control to a resource is then left up to the local operating system and application access control mechanisms. As can be seen, this neither allows the local resource administrator to set a policy for who is allowed to do

what, nor does it minimise his/her workload. The Community Authorisation Service (CAS) [19] was the next attempt by the Globus team to improve upon the manageability of user authorisation. CAS allows a resource owner to grant access to a portion of his/her resource to a VO (or community hence the name CAS), and then let the community determine who can use this allocation. The resource owner thus partially delegates the allocation of authorisation rights to the community. This is achieved by having a CAS server, which acts as a trusted intermediary between VO users and resources. Users first contact the CAS asking for permission to use a Grid resource. The CAS consults its policy (which specifies who has permission to do what on which resources) and if granted, returns a digitally self-signed capability to the user optionally containing policy details about what the user is allowed to do. The user then contacts the resource and presents this capability. The resource checks that the capability is signed by a known and trusted CAS and if so maps the CAS's distinguished name into a local user account name via the Gridmap file.

The EU DataGrid and DataTAG projects developed the Virtual Organisation Membership Service (VOMS) [3] as a way of delegating the authorisation of users to managers in the VO. VOMS has gone through a number of iterations in its development. Initially it was a system for dynamically creating Gridmap files from LDAP directories containing details about VO users. Resources could pull a Gridmap file from this periodically. Thus the resource owner never had to actually create or manage the Gridmap file. This system, however, was not scalable. Work within these EU projects then evolved into a push system in which the VOMS server digitally signed a "pseudo-certificate" for the VO user to present to the resource. This pseudo-certificate could contain a local user account name, in which case no Gridmap file would be needed or it could contain other privileges or group membership details, in which case software would be needed by the resource to interpret this information and grant appropriate rights. The software they developed for this is called the Local Centre Authorisation Service (LCAS) [23]. LCAS makes its authorisation decision based upon the user's certificate and the job specification, which is written in job description language (JDL) format.

IV. TRUST MANAGEMENT IN VIRTUAL ORGANISATIONS

Previous work has enriched the lifecycle for managing VOs with security information [5][26]. Here, we revisit the VO lifecycle, augmenting it with actions for trust management.

Following [7], Trust Management Systems (TMS) can be divided into two main types: policy-based TMS and reputation-based TMS.

In policy-based TMS, the different entities that constitute the system exchange and maintain credentials to establish the trust relationships. The main goal in this kind of systems is to enable access control by verifying the credentials –certification trust- and restricting access to credentials-based predefined policies –resource access trust-.

In reputation-based TMS, there exists a mechanism by which a system requesting a resource evaluates the trust of the system providing the resource. It is closed related to context trust. The trust values can be a function of the global and local reputation along with the different policies. Key elements in this type of systems are the reputation model, the metrics and how feedback is generated.

In relation to the VO lifecycle, we distinguish four main phases: Identification, Formation, Operation, Evolution, and Dissolution.

A. *VO Identification*

The identification phase addresses setting up the VO - this includes selection of potential VO partners from the network of organisations by using search engines or looking up registries. Depending on the resource types, the search process may consist in a simple matching (e.g., in the case of computational resources, processor type, available memory and respective data may be considered search parameters with clear cut matches) or in a more complex process, which involves adaptive, context-sensitive parameters. For an example, the availability of a simulation program may be restricted to specific user groups or only for certain data types, like less confidential data, etc.

Before starting this phase, the creator of a VO should select the trust management systems to be used (policy-based or reputation-based TMSs) and the trust policies that will be used. Such information may be taken into account for searching for potential VO-partners. For instance, the parameters for the search may include in addition to service/resource descriptions, trust and reputation ratings, security grades, etc. The process may also involve metadata such as security and trust policies or Service Level Agreement (SLA) templates with ranges of possible values and/or dependencies between them.

The identification phase ends with a list of candidates that potentially could perform the roles needed for the VO, taking into consideration trust and security information.

B. *VO Formation*

In the formation phase, the list of potential VO candidates is reduced to the set of VO members. A central component in this phase is the VO manager, who negotiates with the VO candidates their participation in the VO; selects the VO members, and distributes VO-level configuration information such as policies, SLAs, etc. The negotiation process include trust negotiation

An important process in this phase is trust negotiation: the process by which all trust information –credential, reputation metrics, policies- is negotiated between the VO manager and the VO members.

In principle, the intended formation may fail due to at least two reasons: (a) no provider (or not enough providers) is able to fulfil all given requirements comes to SLA, trust, security, etc. or (b) providers are not (fully) available at the specified time. In order to circumvent these problems, either the requirements may be reduced ("choose the best available") or the actual formation may be delayed to be re-launched at a more suitable time. Obviously there may be the case, where a general restructuring of the requirements led to a repetition of the identification phase.

C. VO Operation

The operational phase could be considered the main life-cycle phase of a VO. During this phase the identified services and resources contribute to the actual execution of the VOs task(s) by executing pre-defined business processes (e.g. a workflow of simulation processes and pre- and post processing steps). A lot of additional issues related to management and supervision are involved in this phase in order to ensure smooth operation of the actual task(s). Such issues cover recording of and reacting to participants' performance, updating and changing roles and therefore access rights of participants according to the current status of the executed workflow, carrying out financial arrangements (accounting, metering), etc. In certain environments persistent information of all operations performed may be required to allow for later examination e.g. to identify fault-sources.

Throughout the operation of the VO, service performance will be monitored. This will be used as evidence when constructing the reputation of the service providers. Any violation -e.g. an unauthorised access detected by the access control systems- and security threats -e.g. an event detected by an intrusion detection system- need to be notified to other members in order to take appropriate actions. Unusual behaviours may lead to both a trust re-assessment and a contract adaptation. VO members will also need to enforce security at their local site. For example, providing access to services and adapting to changes and the violations.

D. VO Evolution

Evolution is actually part of the operational phase: as participants in every distributed application may fail completely or behave inappropriately, the need arises to dynamically change the VO structure and replace such partners. This involves identifying new, alternative partner(s) and service(s), as well as re-negotiating terms and providing configuration information as during identification, respectively formation phase.

One of the main problems involved with evolution consists in re-configuring the existing VO structure so as to seamlessly integrate the new partner, possibly even unnoticed by other participants. Ideally, one would like the new service to take over the replaced partners' task at the point of its leaving without interruption and without having to reset the state of operation. There may other reasons for participants joining or leaving the VO, mostly related to the overall business process, which might require specific services only for a limited period of time - since it is not sensible to provide an unused, yet particularly configured service to the VO for its whole lifetime, the partner may request to enter or leave the VO when not needed.

E. VO Dissolution

During the dissolution phase, the VO structure is dissolved and final operations are performed to annul all contractual binding of the partners. This involves the billing process for used services and an assessment of the respective participants' (or more specifically their resources) performances, like amount of SLA violations and reputation. The latter may of particular interest for further interactions respectively for other potential customers. Additionally it is required to revoke all security tokens, access rights, etc. in order to avoid that a participant may (mis)use its particular privileges. Generally the inverse actions of the formation phase have to be performed during Termination. Obviously partial termination operations are performed during evolution steps of the VO's operation phase.

V. Examples of Trust Management Systems

This section examines some trust management systems. It is worth mentioning that most systems resulted from research in the fields of e-commerce and peer-to-peer (P2P) computing; however, the solution they advocate apply generally to distributed systems and can also be adapted for grid computing.

The EigenTrust Approach

The EigenTrust algorithm was initially designed for P2P systems [15], and it has been adapted for grid systems in [24]. The EigenTrust approach is based on the notion of transitive trust: a peer i has a high opinion of those peers who have provided it good services and therefore, peer i is likely to trust the opinions of those peers. The idea of transitive trust leads to a system where global trust values correspond to the left principal Eigenvector of a matrix of normalized local trust values.

The original model considers that each peer stores locally its trust values for the rest of the peers [15]. They do not enforce a method for obtaining these trust values, but they suggest that trust values could be obtained by evaluating each previous transaction between peers, thus being a form of direct trust. Each peer normalizes these trust values obtaining values in the interval [0, 1], 1 being assigned to the most trusted peer. In order to obtain a global view of the network, each peer can ask referrals from its neighbours regarding a third peer. The received trust values can be aggregated using the local trust values for the neighbour as weights. Therefore, using one set of queries that is investigating the neighbourhood graph on a distance of 1, a peer can obtain a trust vector including witnesses of first order. Iterating and querying the neighbours of the neighbours, the global trust vector becomes much refined. Kamvar et al. proved that by further iterations, the global trust vector converges to a value that is unique for the network and is the left principal Eigenvector of the initial matrix of normalized trust values. Therefore, by a repeated query process, each agent can obtain the global trust vector, while still storing locally only its own trust values regarding the rest of the peers. This model has also a remarkable probabilistic interpretation, as a peer might interrogate its neighbours with the probability given by the neighbours local trust value. In order to make the model more resistant to collusion, they propose to consider the founders of the network as a-priori trusted nodes and at each iteration step, to take a part of the trust as being the trust given by these nodes. Addressing the distribution of the storage of the data, the paper lets each node to store also its global trust number part of the global trust vector, besides the normalized trust values. Doing this, the initial a-priori trusted nodes get lost in the network anonymity, making the model more reliable.

PeerTrust

PeerTrust [27] is a P2P trust management systems whose trust metric consists of two parts. The first part is a weighted average of the amount of satisfaction a peer receives for each transaction. The weight takes into account the credibility of feedback source to counter dishonest feedback, and transaction context to capture the transaction-depended characteristics. The second part of the metric adjusts the first part by an increase or decrease of the trust value based on community-specific characteristics.

Trust information is stored in a distributed manner over the network. Each peer or node in the network has a trust manager -that is responsible for feedback submission and trust evaluation-, and a data locator for placement and location of trust data over the network. The trust data can be distributed in the different peers based on any existing data management mechanism that used the Distributed Hash Table mechanism of distributing data across the network.

NICE

NICE [17] adapts ideas of social networks to the structure and security requirements of a fully decentralized P2P network, equipped with a PKI infrastructure. In NICE, each agent comes to the system with a pair of public and private keys and the messages are signed by the peers who are creating them. Therefore, after each transaction between a peer client A and a servant B, the peer A generates a cookie with its perceived feedback (trust value) for the transaction. The trust value of a node B at a node A is a measure of how likely the node A believes a transaction with node B will be successful. Trust values scales from 0 to 1. Peer A sends the cookie to B and peer B can store the cookie as a reference of its effectiveness in other transactions. Peer B can decide which cookies to store and how long to store such a cookie. More, each peer could posses its own algorithm for updating and storing the trust values it receives from transaction partners.

When a peer A deliberates to enter a transaction with peer B, a cookie might exist between A and B and in this case, this cookie contains the trust peer A has for B. If previous transactions did not exist or are discarded, peer A will ask its partners about having cookies for B and the partners will continue to spread the request into the network till a path between A and B is established. As a response to its request, peer A will collect the cookies that link it to B and therefore, will have the graph structure of the social network. In this graph structures, paths between A and B are evaluated either by selecting the minimum trust value on the path or by multiplying the trust values. Therefore, the strongest path can be selected.

Refinements mechanisms are presented with regard to generating cookies requests. One of them is to allow users to store negative cookies. It is obvious that after a defective transaction, when peer A will generate a cookie for peer B with a low trust value, peer B will simply discard the cookie, as it does not help him. But instead, peer A can retain the cookie as a blacklist, and never entering transactions with peer B.

This approach shows how ideas from multi-agent research can be successfully employed in P2P computation. As NICE is concerned with resource bartering, this environment comes closer to a fully distributed and decentralized grid.

PathTrust

PathTrust [16] considers the problem of using a reputation system for member selection in the VO formation phase. To enter the VO formation process, a member must register with an enterprise network (EN) infrastructure by presenting some credentials. Besides user management, EN supplies with a centralized reputation service. At the dissolution of the VO, each member leaves feedback ratings to the reputation server for other members with whom they experienced transactions. The feedback ratings can be positive or negative ratings. The system requires each transaction to be rated by the participants.

PathTrust arranges the participants in a graph structure similar with the one of NICE. Each edge in the graph is weighted with the trust between the nodes at the ends of the edge. This trust is computed by accounting the number of positive feedback let by participant i for participant j and subtracting the number of negative feedback weighted by the report between the total positive feedback and total negative feedback participant i has submitted. If the report is less than 1 that is i submitted more negative feedback, then the weight is 1.

The above trust value is normalized by the total number of transactions and therefore, it is less than 1. To distinguish between no transactions experience at all and some existing experience, the trust value is lower bounded by some small value (0.001). The weight of a path in the graph is the product of the weights of the edges that compose that path. As in NICE, for assessing the reputation between two nodes in the graph, the PathTrust algorithm selects the path with the maximum weight. Like in the EigenTrust approach, the trust value is seen as the probability of selecting a participant from the list of possible alternatives.

The PathTrust scheme was evaluated against the EigenTrust algorithm and against attacks by reporting fake transactions in the system. It seems that with EigenTrust, a cheater can gain more profit than with PathTrust. The second test they performed was against random selection of participants. The results show that EigenTrust looses its advantage over random selection once cheating was introduced in the system. This loss occurs also with PathTrust, but is much lower. Therefore, to prevent cheating, the authors propose the usage of a transaction fee.

PathTrust is one of the first attempts to apply reputation methods to grids by approaching VO management phases. They approached only partner selection and did not tackled organizational aspects. Their model still lacks of dynamics, as the feedback is collected only at the dissolution of the VO. But, the advance in the field is given by the fact that ideas from previous research were successfully transferred in the area of VOs and grids.

VI. CONCLUSIONS

Managing trust in autonomic and dynamic systems like Grids is of paramount importance. This paper has analysed several classes of trust and their use in Grids: service provision,

resource access, delegation, certification and context trust. Current technologies for managing trust have been also discussed.

The concept of Virtual Organisations is central to Grids. We have enriched the classical VO lifecycle with trust management activities. Trust values and trust policies are created before starting the VO identification phase. In the VO Identification phase, trust information such as reputation could be taken into account when selecting potential VO candidates. The VO formation phase includes all activities related to trust negotiation. During VO operation, trust values are computed and distributed among the VO participants. In VO dissolution, trust information such credentials and access rights are revoked to avoid mis-use of the information.

ACKNOWLEDGMENT

This work is funded by the European Commission under the IST FP6 projects CoreGRID -project No.004265- and GridTrust -project No. 033827-.

REFERENCES

[1] A. Abdul-Rahma, S. Hailes. Supporting Trust in Virtual Communities. In Hawaii Int. Conference on System Sciences 33, 2000.

[2] M.D. Abrams and M.V. Joyce. Trusted Computing Update. Computers and Security, 14(1):57–68, 1995.

[3] R. Alfieri, R. Cecchini, V. Ciaschini, L. dell Agnello, A. Frohner, A. Gianoli, K. Lorentey, F. Spataro. VOMS: An authorization system for virtual organizations. In F. Fernandez Rivera, M. Bubak, A. Gomez Tato, and R. Doallo, editors, Grid Computing: First European Across Grids Conference, volume 2970 of Lecture Notes in Computer Science, pages 33–40. Springer, 2004.

[4] A.E. Arenas (ed). Survey material on trust and security in grids. CoreGRID Project Deliverable D.IA.03, October 2005.

[5] A.E. Arenas, I. Djordjevic, T. Dimitrakos, L. Titkov, J. Claessens, C. Geuer- Pollman, E. Lupu, N. Tuptuk, S. Wesner, L. Schubert. Towards web services profiles for trust and security in virtual organisations. In A. Ortiz L.M. Camarinha-Matos, H. Afsarmanesh, editor, Collaborative Networks and their Breeding Environments. Springer, 2005.

[6] P.J. Broadfoot, A.P. Martin. A critical survey of grid security requirements and technologies. Oxford University Computing Laboratory Technical Report, PRG-RR-03-15, 2003.

[7] A. Chakrabarti. Grid computing security. Springer, 2007.

[8] D. Chadwick. Authorisation in grid computing. Information Security Technical Report, 10(1):33–40, 2005.

[9] F. D'andria, J. Martrat, T. Kirkham, S. Naqvi, J. Gallop, A.E. Arenas. The evolving use of service level agreements and the influence of trust within the support and development of grids to enable a next generation of business models. International Workshop on Service Oriented Computing: a Look at the Inside (SOC@Inside'07), Austria, 2007

[10] T. Dimitrakos. System models, e-risk and e-trust: towards bridging the gap? In V. Tschammer B. Schmid, K. Stanoevska-Slabeva, editor, Towards the E-Society: E-Business, E-Commerce, and E-Government. Kluwer Academic Publishers, 2001.

[11] D. Gambetta. Can we trust trust? In D. Gambetta, editor. Trust: Making and Breaking Cooperative Relations. University of Oxford, 2000.

[12] T. Grandison, M. Sloman. A survey of trust in internet applications. IEEE Communications Survey and Tutorials, 3, 2000.

[13] A. Josang. An algebra for assessing trust in certification chains. In J. Kochmar, editor, Proceedings of the Network and Distributed Systems Security Symposium (NDSS'99). The Internet Society, 1999.

[14] A. Josang, R. Ismail, C. Boyd. A survey of trust and reputation systems for online service provision. Decision Support Systems 43(2), pages 618-644, 2007.

[15] S.D. Kamvar, M.T. Schlosser, and H. Garcia-Molina. The EigenTrust algorithm for reputation management in P2P networks. In WWW '03: Proceedings of the 12th international conference on World Wide Web, pages 640–651, New York, NY, USA, 2003. ACM Press.

[16] F. Kerschbaum, J. Haller, Y. Karabulut, and P. Robinson. Pathtrust: A trust-based reputation service for virtual organization formation. In iTrust2006, 4th International Conference on Trust Management, Vol. 3986, Lecture Notes in Computer Science, pp. 193–205. Springer, 2006.

[17] S. Lee, R. Sherwood, B. Bhattacharjee. Cooperative peer groups in NICE. In IEEE INFOCOM, 2003.

[18] J. Linn (ed). Liberty trust models guidelines. Liberty Alliance Project, version 1, 2004.

[19] 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.

[20] L. Rasmusson, S. Janssen. Simulated social control for secure internet commerce. In C. Meadows, editor, Proceedings of the 1996 New Security Paradigms Workshop. ACM, 1996.

[21] G.C. Silaghi, A.E. Arenas, L.M. Silva. A utility-based reputation model for service-oriented computing**.** Toward Next Generation Grids, Proceedings of the CoreGRID Symposium 2007. Springer, 2007.

[22] G.C. Silaghi, A.E. Arenas, L.M. Silva. Reputation-based trust management systems and their applicability to grids. CoreGRID Technical Report TR-0064, 2006.

[23] M. Steenbakkers. Guide to LCAS. Version 1.1.16. Documentation of the European DataGrid Project, 2003.

[24] G. von Laszewski, B.E. Alunkal,, I. Veljkovic. Towards reputable grids. Scalable Computing: Practice and Experience, 6(3):95–106, 2005.

[25] V. Welch, F. Siebenlist, I. Foster, J. Bresnahan, K. Czajkowski, J. Gawor, C. Kesselman, S. Meder, L. Pearlman, S. Tuecke. Security for grid services. In International Symposium High Performance Distributed Computing, pp. 48-57, 2003.

[26] S. Wesner, L. Schubert, T. Dimitrakos. Dynamic virtual organizations in engineering. In Proceedings of German-Russian Workshop, 2005

[27] L. Xiong, L. Liu. PeerTrust: supporting reputation-based trust for peer-to-peer electronic communities. IEEE Transactions of Knowledge and Data Engineering, vol. 16, no. 7, pp. 179-194, 2004.

[28] P.R. Zimmermann. The Official PGP User's Guide. MIT Press, 1995.