sGUTS: Simplified Grid User Trust Service for Site Selection

Ioanna Dionysiou
Department of Computer Science
University of Nicosia
Nicosia, Cyprus
dionysiou.i@unic.ac.cy

Harald Gjermundrod
Department of Computer Science
University of Nicosia
Nicosia, Cyprus
gjermundrod.h@unic.ac.cy

***Abstract*—Even though trust plays a significant role during decision-making in open collaborative environments, still Grid user trust mechanisms have not been widely deployed in Grid computing settings. In this paper, a conceptual framework that is an extension of a novel Grid user trust service (GUTS) is presented, which aims at leveraging Grid functionality with trust mechanisms with a special focus on achieving end-user trust in an intuitive and practical manner. Trust in GUTS is utilized during the grid site selection process, where sites are ranked based on expected service requirements for a user grid project. In the proposed conceptual framework, the center of the trust management process is the user who decides and specifies the needs of his/her project, which in turn are mapped to trust requirements.**

***Keywords*-grid computing, trust mechanisms, user perceptive**

I. INTRODUCTION

Trust is an abstraction of individual beliefs that an entity has for specific situations and interactions. It encompasses even more than message confidentiality and source authentication, which have been the traditional trust scopes. Trust's broader scope covers not only security issues but behavioral and Quality of Service (QoS) issues as well. Consider a data dissemination service, that operates on the following policy: valid and non-malicious information (behavioral requirement) is publicly available but must not be tampered with (security requirement) and must be received in a timely manner (QoS requirement). In order to enforce this policy the appropriate security, behavioral, and QoS mechanisms must be in place to implement the policy. Digital signing algorithms can guarantee message integrity but they offer no assurance about the quality of the message contents; this is the task of behavioral mechanisms that deduce behavioral patterns and trends for the information producer. Finally, QoS mechanisms are needed to provide guarantees that the information producer and the network will meet the QoS properties as contracted. We call behavior, security and QoS the three *general trust facets*, which are further refined into more specific facets called *requirements*. Requirements include authentication, competence, and delivery rate. Those, could be further refined into attributes. Any trust requirement for a distributed application can be categorized as security, behavioral, or QoS requirement.

While trust is an integral part of decision making in collaborative models, there is no unique way to determine the right level of trust, or which facets to use. Researchers have defined trust concepts for many perspectives, with the result that trust definitions overlap or even contradict each other. The reason is that decisions about how to evaluate each facet lie with the evaluator and can differ substantially from situation to situation. End-to-end trust is essential for topologies where interactions are dynamic and they always involve the collaboration of multiple entities to disseminate data from its source to its destinations. Needless to say, trust is useful only if it is managed in an appropriate and systematic manner. An entity's beliefs are not static but they change as time progresses and new information is processed into knowledge. Trust must evolve in a consistent manner so that it still abstracts the entitys beliefs accurately. In this way, an entity continuously makes informed decisions based on its current beliefs.

Collaborative settings, such as grid environments, where risk and uncertainty are inherent due to their open nature could greatly benefit from using trust as an integral part of decision-making. For example, a grid user could select the most *trustworthy* site from a pool of available sites to submit a job. A grid user could specify *trust requirements* in a parametrized job description. Sites, offering computational resources, could be rated based on their *reputation* among grid users. Trust in the Grid environment has been analyzed and various systems have been proposed (a summary of such systems can be found in [5]). The difference between these systems and the one proposed in this paper is that the former have focused on how to define/model trust in a Grid environment from the system point of view, while sGUTS tries to abstract away the notion of trust from the end user and present him/her with a set of questions that specify the trust needs for a project. Based on these questions, the trust requirements will be automatically derived. To the best of our knowledge, trust is not utilized in this manner by existing trust frameworks for grid infrastructures.

This paper extends a framework that utilizes trust for ranking grid sites and in consequence, allowing grid users to select a site that is the most appropriate for their specific needs. The proposed framework is at the current stage a

 ISBN: 978-1-61208-201-1

conceptual framework and implementation is currently under progress. The primary contribution is the simplification of the service requirements specification by the end-user, which is done in an intuitive manner. It is not always apparent to the Grid user what is the most appropriate configuration for a particular job, something that is vital for selecting the site that best matches the job requirements. In order to address this limitation, the configuration is automatically generated upon the user responses to a predefined set of questions.

The remaining of the paper is organized as follows: Section II discusses existing trust approaches in grid infrastructures, followed by Section III that presents an overview of Grid User Trust Service (GUTS), a trust-based ranking framework applicable to grid interactions. Section IV extends this framework by simplifying the trust specification process. Finally, Section V concludes.

II. Trust Management in Grid Environments

A computational Grid [11],[12],[10] is a collection of distributed, possibly heterogeneous resources that can be used as an ensemble to execute computational-intense applications, such as earth observation, climate modeling, and biology applications. The two pillars of the Grid paradigm are access to shared services and support of multi-user collaboration, while the resource owner is always in control. Sites are organized in one or more virtual organizations, thus creating federations of central services, such as cross-domain authentication, authorization, job-site matching, and job dispatching. Authorized users access computational and storage resources of a site by contacting either the central services or the site itself.

The Grid must be managed to allow coordination of the dynamic cross-organizational resource sharing among virtual organizations not only in an efficient manner but securely as well. This is nontrivial to achieve, mainly due to the self-managed and unpredictable nature of the virtual organizations. Nevertheless, there are deployed mechanisms that provide a number of security services. For instance, a single sign-on authentication mechanism is already available via proxy certificates. Authorization is implemented via access control lists. X.509 certificates could be used not only to authenticate a user but to encrypt traffic flows.

Humphrey et al. [15] analyzed a comprehensive set of Grid usage scenarios with regard to security requirements. However, cryptographic algorithms and access control schemes cannot be used to reason about the more general concept of trust, – the *belief* that an entity will behave as expected under certain conditions – as there are no provisions for a number of security, behavioral, and QoS issues such as data privacy, site administrators qualifications, and service reliability provided by the various sites. An authenticated and authorized user has no guarantees that the Grid infrastructure will successfully carry out the execution of a submitted job. The Grid user remains defenseless against job failures, which according to a recent study [18] account for a large percentage of all submitted jobs, and attempts to compensate for any potential failures by submitting the same job to multiple sites.

The failures could be attributed to security, behavioral, or QoS factors, thus making the Grid environment the ideal setting for deploying trust as an integral part of the decision-making. In the recent years, there has been an increasing interest in addressing specific trust challenges in Grid environments.

In [21], the Trust domains establishment is mentioned as being one of the three key functions in a Grid Security Model, where virtual organizations establish trust among users and resources that is expressed in policies and proxy certificates. The authors in [6] leverage the authentication and authorization capabilities of the Grid security framework using trust negotiation with PeerTrust policy language whereas the importance of trust negotiation is reiterated in [17]. Similarly, [1] uses trust federation and dynamic authorization supported by GRIA middleware to demonstrate the dynamic federation of resources capability. The research work in [7] focuses on a decentralized resource access control scheme using trust chains and an extended SPKI/SDSI that allow intermediate levels of trust to be expressed per chain, rather as a binary model of valid or invalid. In [20], [4], and [16] trust management systems for Grids are presented, which assist in evaluating the trust value of the various Grid sites and specify how to set the metrics trust evaluation.

A more general approach to trust is presented in the survey by Arenas et al. in [3], which discusses the trust classifications in Internet services [14] from the Grid perspective. Furthermore, [2] investigates the possibility of exploiting reputation systems for managing virtual organizations. The SCOUT [22] middleware assists the user in belief calculation and evidence source trust calculation in order to use service in a Service Oriented Architecture.

Still, there is no implementation of a suite of trust mechanisms that the average Grid end-user could utilize to specify its trust requirements and incorporate them in decision-making. Although, GridAdmin [19] is a trust management system to be used by administrators of Grid sites, and not end users. The proposed work in [8] investigates the emerging technological challenges associated with the support of such a comprehensive user-oriented adaptive trust framework deployed in Grid infrastructures.

III. Grid User Trust Service (GUTS) Overview

The Grid User Trust Service (GUTS) framework is a trust management service tailored to the needs of a typical Grid user [9]. It comprises of three main components, as shown in Figure 1. The first component, **Grid Middleware-Agnostic Trust Specification**, allows a user to specify the trust requirements for a Grid service. The second component,

 ISBN: 978-1-61208-201-1

Grid Middleware-Dependent Trust Specification, maps those general requirements to the specific Grid infrastructure, yielding the trust profile of a project. Finally, the **Trust Management and Visualization** component gathers and evaluates evidence provided by the specific Grid infrastructure, updates the trust profiles accordingly, and produces a ranking list of the various Grid sites.

Starting with the Grid Middleware-Agnostic Trust Specification, the objective is to formulate an XML schema that captures the trust requirements for a grid service. Those are being abstracted to the user as a set of attributes along with their types and associated value ranges. The XML schema is used to instantiate valid XML trust requirements documents for a Grid project. Attributes that could be specified by the user include administrator certification, host site information, security level, proximity to local site, uptime, job failures, and hardware profile, and all of them are irrespective of the underlying Grid middleware. Two different methods are provided to help the user supply the trust requirements. For the Grid novice e-scientist, a wizard is available and for the Grid-aware e-scientist, a multi-paged editor is available. In GUTS, the wizard concept is used to guide the user in creating a trust requirements document and it consists of both required and optional dialogs. Similarly, the GUTS multi-page editor consists of tabs performing the same task - the exact number of tabs depends on the XML schema and on the way the set of attributes can conceptually be grouped together.

Proceeding with the Grid Middleware-Dependent Trust Specification component, those trust requirements specified in the previous component are translated and mapped into specific requirements that could actually be evaluated, based on the information supplied or deduced by the specific Grid middleware. The GUTS framework supports a specific Grid infrastructure/middleware only in the case where plug-ins for that specific Grid middleware are available. GUTS plug-ins and abstract interfaces will be accessible to the developer for extension as to support new Grid middleware.

Trust is not useful unless it becomes part of the decision process. In the case of the Grid, an end-user could utilize trust knowledge to choose the most suitable site for the specific job. An important aspect though is the presentation of trust results to the user. The final component is the trust management component, which is responsible for not only managing the project trust profiles that are stored in a database but also for presenting the user with a ranking list that could serve multiple purposes such as becoming a decisive factor when choosing a site to submit a job and provide the current "trustworthiness" of the various Grid sites that are available to the e-scientist. In addition, the user could also initiate to view past ranking lists as well as generate list where the rank over time for a specific site is provided.

It has been demonstrated in [9] that conceptually GUTS could be integrated with g-Eclipse client [13] that supports Grid/Cloud middleware like gLite, GRIA, and Amazon Web Services.

IV. sGUTS: Simplified GUTS

GUTS allows a user to specify the project needs by assuming that the user is knowledgable when it comes to technical specifications. However, this shouldn't be expected and instead of having the user profiling the grid service, it is wiser to have GUTS profile the project and map the profile into a set of trust requirements that the service needs to fulfill. This section presents sGUTS, an extension to GUTS, that abstracts the service requirements, and in consequent offloads the user from technical jargon and assists in a more appropriate site selection.

A. Diversity of Service Requirements for Grid Projects

The popularity of grid computing mainly lies on the emerging needs of scientists to process and store vast amount of heterogeneous data and at the same time increase collaborations among laboratories and research institutions, not necessarily in the same geographical location. Nevertheless, the changing scale and scope of science should not have an impact on a very simple premise: A scientist wants to do science (or e-science) and not computer science. The user interface that serves as the gateway to the grid infrastructure resources must be simplistic and intuitive for the average e-scientist.

It has been observed that grid failures or security incidents occur due to misconfigurations, with the source of these often being the lack of technical knowledge by the user. The service provider overwhelms an e-scientist with technical jargon, resulting in either specifying too strict requirements or too few. Depending on the nature of the experiment, the service requirements can greatly vary. Below, are examples illustrating the aforementioned claim.

1) Scenario 1: Molecular Modeling for Drug Discovery Experiment: Drug design using molecular modeling techniques, or molecular docking, helps scientists to predict how small molecules bind to an enzyme or a protein receptor. It is both a computationally and data intensive process to dock each molecule in the target chemical database. The grid infrastructure could facilitate the parallel and distributed processing of molecular docking. The average e-scientist would expect high computational accuracy from the site resources and data communication integrity during the data traversal of the communication network. If primary data were to be stored in the grid, then the storage has to be reliable with a very low possibility of loss of data. The classical requirements on databases, such as durability, consistency, reliability, scalability are needed for critical experiments. On the other hand, performance is not a primary concern as the focus is on correct and reliable results.

 ISBN: 978-1-61208-201-1

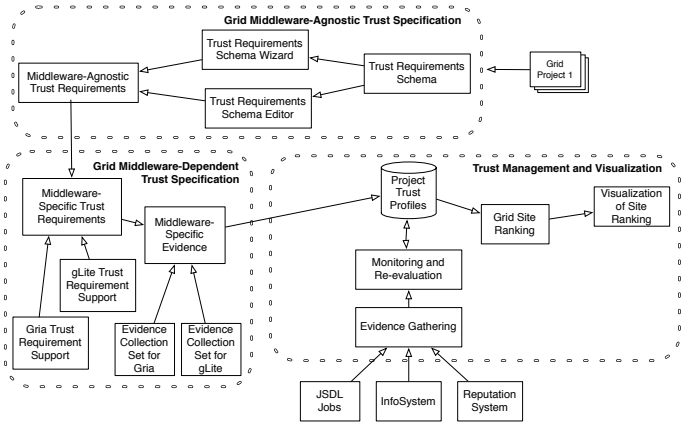

Figure 1. GUTS Framework

2) Scenario 2: Environmental Phenomena Prediction Task: Real time data about environmental phenomena could be processed, modeled, and correlated to predict natural disasters, leading into an early warning system. Grid computing could facilitate such a system, having appropriate sensors integrated with the underlying infrastructure at various locations, leading into the collection and distribution of the measurements to applications that use them to make predictions. Such an application will impose soft real time delivery on data: it is essential to use fresh data. However, data loss could be acceptable as prediction algorithms usually operate on incomplete data sets. Furthermore, the nature of the data does not justify any confidentiality or access control restrictions as the data is public information.

3) Scenario 3: Training Event: Grid federations or funded grid projects often offer initial training events for end-users. An induction course on grid technologies usually aims at demonstrating the capabilities of the underlying grid infrastructure. A successful event heavily relies on the availability of the resources. A site under maintenance or a site that is down could disrupt the normal flow of the training. Thus, high availability is expected, without too much concern on other security and QoS parameters. Jobs that are submitted during a training or educational experience could tolerate slower execution than normal or unencrypted traffic or even non-authenticated provider. Data loss is acceptable as well.

B. Abstraction of Service Requirements

The service requirements for a project depend on its nature, as demonstrated above. Table I illustrates the space (note: this space is not exhaustive, as it is a work in progress to derive a comprehensive list) of the requirements that must be imposed on the grid service for a successful project outcome. Each requirement could be further partitioned into a set of attributes. For example, physical properties could consist of storage room specifications and room temperature whereas confidentiality could be comprised of encryption algorithms and key lengths.

Table I
SERVICE REQS FOR GRID PROJECT

Requirements	Possible Attributes
Security	
authentication	(username+password), (X509 cert.), (biometrics)
integrity	(digital signing algorithm), (key length)
confidentiality	(encryption algorithm) (key length) ((a)symmetric)
availability	(uptime), (downtime frequency)
access control	(ACL), (RBAC), (authentication token)
privacy	(sensitive), (public), (ACL for data)
Behavioral	
Competence	(job failures), (administrator certification)
Motivation	(sysadmin: student, staff, faculty), (host site type)
Physical Info	(hardware/server-room profile), (location)
QoS	
Latency	(proximity), (site infrastructure)
Bandwidth	(site infrastructure), (country infrastructure)
Comp. Accuracy	(hardware/server-room profile)
Database Storage	(RAID), (hardware/server-room profile)

C. Deducing Service Requirements from Coarse-Grained Specifications

The idea is to abstract the process of explicitly specifying all these properties, a task that is carried out by the user. For example, the e-scientist of Scenario 2 may select to encrypt data without knowing that encryption is a costly operation unnecessary for the project needs, that could affect the latency of the received data. Therefore, the vision is to automatically populate the entries of Table I based on the responses that the user will supply to GUTS regarding high-level coarse-grained specifications of the desired

 ISBN: 978-1-61208-201-1

project. This trust project profile, in turn, will be mapped against existing grid services specifications that are ranked according to the level of matching. Similar to GUTS, the user will be given the opportunity to edit the generated trust profile using a multi-paged editor.

The questions below are forming the coarse-grained specifications of a project, and these are grouped into three categories, namely Project Needs, Data Needs, and Computational Needs. Based on the answers that the user provides, a project trust profile is created, and thus a set of service requirements.

1) Project Needs: The first category of questions is directly related to the overall needs of the project. Depending on the user responses, some of the questions in the other two categories will either not be asked or the answer options may be reduced. The questions of this category are the following:

Q1.1: *Is your project computational intensive or data intensive or an equal share of both?*
This will help the system decide what tradeoffs to apply when the choice is between data and computational needs.

Q1.2: *Is the computation more important than the storage of the data or an equal share of both?*
Even though it is a computational intensive project the e-scientist may be more concerned abut how the result is stored than how the computation was performed.

Q1.3: *What is the expected life-time of the resulting data of the project?*
The lifetime of the project may influence where the resulting data should be stored.

Q1.4: *By whom the results of the project will be used for?*
The usage of the results will guide the system to decide which security mechanisms will be needed. The user will choose one of the following predefined choices:

- Single user (me)
- Small group size (<10)
- Medium group size (<100)
- Large group size (>99)
- Public access, anyone can access the result

2) Data Needs: The second category prompts the user to provide input regarding the needs of handling the data related to the project. This includes both the input data and any derived results from computations. The questions are used to profile the requirements on the reliability, integrity, and privacy for the project data.

Q2.1: *Is the provenance of the stored data of the project necessary?*
The provenance is important for certain applications in proving that the data has not been tampered with/alternated.

Q2.2: *Where would you prefer the data to be stored?*
Due to the nature of the data, local government laws may prohibit export of the data in another country. Furthermore, the e-scientist may wish to store the data close to the local site. The predefined choices are:

- Close to my site
- Preferable in my country/region
- Location is not important

Q2.3: *What would the consequences of data loss be?*
Depending on whether or not the input data is primary data or derived data, the loss of the resulting data can greatly vary. Similarly, if the computation cost of deriving the data is extremely high, the cost of recomputing it may not be feasible. The predefined choices for this question are.

- Danger for loss of life (the user should be warned that the Grid environment may not be the most appropriate service provider in this case)
- Scientific findings may be lost (forever)
- Loss of investments made by doing the study
- Inconvenience of having to rerun the computation
- No loss due to nature of the data

Q2.4: *What would be the consequences of any modification (accidental or deliberate) to the stored data?*
Similar to Q2.3, except that the data is not lost but modifications have occurred. The choices that the user is presented with are the same as Q2.3.

Q2.5: *Does the input/resulting data contain any sensitive information?*
This question will be split into two questions, one for the input data and one for the resulting data. This question will define what are the privacy needs with regard to the data of the project:

- Highly sensitive (lose of life may result if leaking occurs)
- Sensitive (personal privacy laws apply)
- Moderate (lose of business secrets)
- Low (prefer to keep the data secret)
- N/A (publicly available data)

3) Computational Needs: The third category captures the computational needs for the project, and to be more specific its performance, reliability, and integrity requirements.

Q3.1: *What are the consequences of missed deadline of the completion of the computation?*
Deadlines can be missed in case of server failure, power failure, sever room overheating, server miss configuration, or accidental shutdown of the site. The user will choose one of the following predefined choices:

- Danger for loss of life

 ISBN: 978-1-61208-201-1

- Scientific findings may be lost (forever)
- Loss of investments made by doing the study
- Inconvenience of having to rerun the computation
- No consequence due to nature of the project

Q3.2: *What the delay of receiving computational data result in?*
This question is similar to Q3.1, except that here we are referring to the case where the initial computation service fails and the complete computation will have to be redone on other resources. The choices that the user can pick from are the same as those for Q3.1.

Q3.3: *What would be the consequences of any modification (accidental or deliberate) to the computation?*
Most Grid computational services have *server class* infrastructure (including hardware and server room), but not all. It could be the case that bit-flipping could not be detected, hence what would be the consequences of such problems. The choices that the user can pick from are the same as those for Q3.1.

4) Demonstration of Service Requirements Mapping from Specifications: It is beyond the scope of the paper to present the actual workings of the mapping, as the objective is to present a conceptual proof-of-concept of the usefulness of such a mechanism. Table II illustrates how user responses get translated to specific service requirements for Scenario 1. In this scenario, the user is a university researcher analyzing data to discover a cure for a disease, hence there is no need for secrecy of results but accuracy is vital for the success of the project.

Table II
SERVICE REQUIREMENTS FOR SCENARIO 1

	Attributes	**User Response**
Security Req.		
authentication	(X509 certificate)	(default)
integrity	(MD5)	Q3.3 - 4th option
confidentiality	(N/A)	Q1.4 - 5th option
availability	(best effort)	Q3.1, Q3.2 - 4th option
access control	(public)	Q1.4 - 5th option
privacy	(public)	Q1.4 - 5th option
Behavioral Req.		
Competence	(N/A)	Q2.3, Q3.1- 4th option
Motivation	(N/A)	Q2.3, Q3.1- 4th option
Physical Properties	(high quality)	Q3.3 - 4th option
QoS Req.		
Latency	(N/A)	Q3.1, Q3.2 - 4th option
Bandwidth	(N/A)	Q3.1, Q3.2 - 4th option
Comp. Accuracy	(high quality)	Q1.2 - computational
Database Storage	(average quality)	Q2.3, Q2.4 - 4th option

V. CONCLUSION

This paper presented sGUTS, an extension of GUTS, that supports automatic generation of service requirements for a grid project based on user responses on a set of predefined coarse-grained questions that capture the project nature and its data and computational needs. In this way, the average e-scientist does not need to be knowledgeable on technical grid details as the system maps his/her answers to service requirements that are further used to select the most appropriate grid service to satisfy the project at hand.

Even though research efforts exist for managing trust in the grid environment, still the focus is on how trust is perceived by the system (or site administrator) rather than on attempting to simplify the interpretation of trust for the end user. In the proposed conceptual framework, the center of the trust management process is the user who decides and specifies the needs of his/her project, which in turn are mapped to trust requirements. The proposed techniques leverage the functionality of the trust management system to include user input in an intuitive manner. The research effort presented in this paper is still under development.

As far as future directions are concerned, the applicability of sGUTS in cloud services will be investigated. In these settings, there is an additional constraint, which is the budget allocated for the project, that must be utilized in an efficient way and at the same time fulfill the aforementioned service requirements.

REFERENCES

[1] Mehran Ahsant, Mike Surridge, Thomas Leonard, Ananth Krishna, and Olle Mulmo. Dynamic trust federation in grids. In Ketil Stølen, William H. Winsborough, Fabio Martinelli, and Fabio Massacci, editors, *iTrust*, volume 3986 of *Lecture Notes in Computer Science*, pages 3–18. Springer, 2006.

[2] Alvaro Arenas, Benjamin Aziz, and Gheorghe Cosmin Silaghi. Reputation management in grid-based virtual organisations. In *International Conference on Security and Cryptography (SECRYPT08)*, pages 538–545, 2008.

[3] Alvaro Arenas, Michael Wilson, and Brian Matthews. On trust management in grids. In *Autonomics '07: Proceedings of the 1st international conference on Autonomic computing and communication systems*, pages 1–7, ICST, Brussels, Belgium, Belgium, 2007. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering).

[4] B. Ashrafijoo, A. Habibizad Navin, M.-K. Mir Nia, S. Abedini, and N. Azari. Trust management in grid computing systems based on probability theory. In *Education Technology and Computer (ICETC), 2010 2nd International Conference on*, volume 4, pages V4–316 –V4–320, June 2010.

[5] Benjamin Aziz, Alvaro Arenas, Fabio Martinelli, Paolo Mori, and Marinella Petrocchi. *Trust Modeling and Management in Digital Environments: from Social Concept to System Development*, chapter Trust Management for Grid Systems, pages 149 – 178. IGI Global, 2010.

[6] J. Basney, W. Nejdl, D. Olmedilla, V. Welch, and M. Winslett. Negotiating trust on the grid. In *In 2nd WWW Workshop on Semantics in P2P and Grid Computing*, pages 1–20, 2004.

 ISBN: 978-1-61208-201-1

[7] José de R. P. Braga, Jr, Alexandre C. T. Vidal, Fabio Kon, and Marcelo Finger. Trust in large-scale computational grids: an spki/sdsi extension for representing opinion. In *MCG '06: Proceedings of the 4th international workshop on Middleware for grid computing*, pages 7–12, New York, NY, USA, 2006. ACM.

[8] Ioanna Dionysiou, Harald Gjermundrød, and David E. Bakken. An initial approach for adaptive trust in grid environments. In *Proceedings of 1st Workshop on Computational Trust for Self-Adaptive Systems (SELFTRUST'09)*, pages 719–722, Athens, Greece, November 2009.

[9] Ioanna Dionysiou, Harald Gjermundrød, and David E. Bakken. Guts: A framework for adaptive and configureable grid user trust service. In *6th International Workshop on Security and Trust Management (STM 2010)*, pages 84–99, Athens, Greece, September 2010.

[10] Ian Foster. What is the grid? - a three point checklist. *GRIDtoday*, 1(6), July 2002.

[11] Ian Foster and Carl Kesselman. The globus toolkit. In Ian Foster and Carl Kesselman, editors, *The Grid: Blueprint for a New Computing Infrastructure*, pages 259–278. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1999.

[12] Ian T. Foster. The anatomy of the grid: Enabling scalable virtual organizations. In *Euro-Par '01: Proceedings of the 7th International Euro-Par Conference Manchester on Parallel Processing*, pages 1–4, London, UK, 2001. Springer-Verlag.

[13] H. Gjermundrod, M. D.Dikaiakos, M. Stuempert, P. Wolniewicz, and H. Kornmayer. An integrated framework to access and maintain grid resources. In *Proceedings of the 9th IEEE/ACM International Conference on Grid Computing (Grid 2008)*, pages 57–64, 2008.

[14] Tyrene Grandison and Morris Sloman. A survey of trust in internet applications. *IEEE Communications Surveys and Tutorials*, 3(4):2–16, 2000.

[15] Marty Humphrey and Mary R. Thompson. Security implications of typical grid computing usage scenarios. *Cluster Computing*, 5(3):257–264, 2002.

[16] P.D. Manuel, S. Thamarai Selvi, and M.I.A.-E. Barr. Trust management system for grid and cloud resources. In *Advanced Computing, 2009. ICAC 2009. First International Conference on*, pages 176 –181, Dec. 2009.

[17] Dilal Miah. A matter of trust: enabling grid security through bilateral negotiation. International Science Grid this week (ISGTW), 2008. http://www.isgtw.org/?pid=1001540, last accessed April 6, 2012.

[18] K. Neokleous, M. D. Dikaiakos, P. Fragopoulou, and E. Markatos. Failure management in grids: The case of the egee infrastructure. *Parallel Processing Letters*, 17(4):391–410, 2007.

[19] T.B. Quillinan, B.C. Clayton, and S.N. Foley. Gridadmin: decentralising grid administration using trust management. In *Parallel and Distributed Computing, 2004. Third International Symposium on/Algorithms, Models and Tools for Parallel Computing on Heterogeneous Networks, 2004. Third International Workshop on*, pages 184 – 192, July 2004.

[20] Kai Wei and Shaohua Tang. A cloud-based recommendatory trust processing method for trust management system of grid. In *Communications and Mobile Computing (CMC), 2010 International Conference on*, volume 1, pages 289 –293, April 2010.

[21] Von Welch, Frank Siebenlist, Ian Foster, John Bresnahan, Karl Czajkowski, Jarek Gawor, Carl Kesselman, Sam Meder, Laura Pearlman, and Steven Tuecke. Security for grid services. In *HPDC '03: Proceedings of the 12th IEEE International Symposium on High Performance Distributed Computing*, pages 48–57, Washington, DC, USA, 2003. IEEE Computer Society.

[22] Chern Har Yew and H. Lutfiyya. A middleware-based approach to supporting trust-based service selection. In *Integrated Network Management (IM), 2011 IFIP/IEEE International Symposium on*, pages 407 –414, May 2011.

 ISBN: 978-1-61208-201-1