A Framework for Semantic-Based Dynamic Access Control in Data Grids[1]

Anil L. Pereira, C. Warren Moseley, Dennis L. Ferron, Benjamin A. VanTreese, David L. Goree and Karl M. Kirch

Department of Entrepreneurship and Computer Systems
Southwestern Oklahoma State University
Weatherford, Oklahoma 73096, U.S.A

Contact Author: Anil L. Pereira
580-774-7194
anil.pereira@swosu.edu

Abstract

The abundance of large commercial and scientific data stores have driven the need for Petascale computation and data integration. Technologies such as Grid Computing are being developed to address this need. Grid Computing supports the coordinated sharing of data and resources among different organizations. Data Grids focus on the management of data and resources for analyzing the data. Though Grid computing technologies have been adopted in many scientific and commercial sectors, many Security issues have to be resolved for them to gain wider acceptance. Their true potential will only be realized by developing secure systems that can encompass multiple organizations. In this paper, we consider the security implications of the dynamic interactions that would occur in Data Grids and examine the requirements for a comprehensive security model to support those interactions. We explain that such a model could be constructed by enhancing existing dynamic role-based access control models and semantic-based access control models. Additionally, we present an enumeration of the security requirements for such dynamic interactions. Our work takes into consideration that the co-allocation of resources and job scheduling in Data Grids should not only be based on the user's request, and the available pool and state of resources, but also on the user's access rights, computing environment and the Security policies of resources. Furthermore, we discuss the problem of making access control decisions dynamically during an application's runtime.

Key words: Dynamic access control, Data Grid, role-based access control (RBAC), semantic-based access control (SBAC), Ontology.

[1] This Research was Supported in Part by NASA Oklahoma Space Grant Consortium.

1. Introduction

The advent of large Digital libraries, large Genomic and Biological databases, and data generated by large scientific experiments on particle accelerators like the Large Hadron Collider (LHC) at the European Organization for Nuclear Research - CERN, have driven the need for Petascale computation and data integration. Technologies such as Grid Computing [Foster01] are being developed to address this need. Grid Computing allows for the coordinated sharing of resources across multiple organizations. It does so by providing necessary infrastructure for the formation of dynamic virtual organizations (VOs). With Grid Computing technology, a virtual supercomputer can be created from existing computational resources and networks. Computation and data intensive applications can be highly parallelized on the Grid due to the availability of a large number of processing nodes and data stores, thus providing Petascale computation and data management capabilities.

Data Grids focus on the management of data and resources for analyzing the data. A Data Grid provides a distributed system middleware that allows distributed communities to access and share data, networks, and other resources in a controlled and secure manner [Foster03]. The motivation behind such a system is to address the following considerations [Chervenak01]: (1) Large data set size, geographic distribution of users and resources, and computationally intensive analysis results in complex and stringent performance demands that are not satisfied by any existing data management infrastructure. (2) No integrating architecture exists that allows us to identify requirements and components common to different systems and hence apply different technologies in a coordinated fashion to a range of data-intensive large-scale application domains.

Though Grid computing technologies have been adopted in many scientific and commercial sectors, many Security issues have to be resolved for them to gain wider acceptance. Their true potential will only be realized by developing secure systems that can encompass multiple organizations. In this paper, we consider the dynamic interactions that occur in Data Grids. For example, during an application's runtime new resources may become available and can be used for better performance. While we do not propose any methodology to facilitate such interactions, we consider the security implications that would arise and examine the requirements of a dynamic access control model to support those interactions. To the best of our knowledge no complete specification or framework for such a model exists. Additionally, we present an enumeration of the Security requirements for such dynamic interactions. Our framework takes into consideration that the co-allocation of resources should not only be based on the user's request, and the available pool and state of resources, but also on the user's access rights, the computing environment and the security policy of resources. Furthermore, incorporated into our framework is a solution to the problem of making access control decisions dynamically during an application's runtime.

The organization of the paper is as follows. In Section 2 we enumerate the security requirements for Data Grids. In Section 3, we examine the requirements of a dynamic access control model for Data Grids. Section 4 contains the conclusion and an outline of future work.

2. Security Requirements for Data Grids

In a Data Grid both users and resources are dynamic. Furthermore, those users and resources belong to multiple organizations each with their own diverse security policies and mechanisms. It is feasible to group sets of users and resources that need to be coordinated towards a common goal into Virtual Organizations (VOs). The key requirement is developing access control mechanisms for these VOs which would interoperate with existing local security infrastructure and would allow resource provider's to have ultimate control over their resources.

During a Data Grid application's runtime some more resources may become available in addition to the ones that it was allocated initially. For example, due to the increase in available bandwidth and availability of storage near a computational resource it may become feasible to copy the required data sets from their source and then process them. Alternatively, a computational resource may become available near the datasets and it may be prudent to move whole or part of the computation there. Another scenario could involve the availability of additional nodes for computation and it may be feasible to spawn additional processes. Also sometimes, it may be better to stall the start up of an application until additional resources become available. However, given these scenarios, fault tolerance becomes even more critical in Data Grids. For example, remote computational nodes

could fail or the security policies for resources may change dynamically, which may result in the termination of certain processes. Such dynamic scenario requires allocation/de-allocation of resources during an application's runtime. The decisions involved in the resource management would be the realm of an intelligent planning and execution module. This module could allow applications to communicate with a scheduler/resource broker and receive information about the availability of resources. When additional resources become available the application can make requests for further allocation of resources and process creation. For example, it may do so after deciding that better performance and security would result from the new allocation. While we do not propose any methodology to facilitate such dynamic interactions, we consider the security implications that would arise and examine the framework for a security model to support those interactions.

One of the implications for security would be that, in addition to the authorization decision made at the VO level when first deploying an application, decisions will also have to be made during the application's runtime. Also, for an application, resource allocation should not only be based on the user's request, and the available pool and state of resources, but also on the user's access rights, computing environment and the security policies of resources. Current scheduling and resource allocation decisions are done irrespective of the user's access rights, computing environment and the security policies of resources. As a result the user can be allocated resources to which it may not have the required access privileges, leading to unnecessary overheads [Bertino04]. For example, a distributed application can be deployed and only some components of it may start up on the resources to which the user has access, while the other components cannot be started up. If those components need to interact then either the whole application will have to be redeployed or the components that do start up will have to wait until the other components are deployed on resources to which there is access. Additionally, the decisions to authorize users at the local level have to be made unnecessarily at the resources to which the users did not have access in the first place. This can be avoided if the user's access privileges are taken into account when making the scheduling and co-allocation decisions. The set of resources allocated would basically be the intersection of the set of the user's rights, the set of available resources, the set of computing environment variables and the security policies of resources.

Other considerations include the location context of users. For example, a user may access Data Grid resources through a secure link from his office and later move to his/her home PC from where a secure link cannot be established. The implications for security could be that his access privileges may need to be altered because the data he receives may be sensitive. Alternatively, while accessing data, the server load may exceed a certain threshold. Due to this and the fact that a user's task may not have high enough priority the server may decide to curtail the user's privileges [Zhang03].

Users without Grid Credentials and who are not members of VOs, may want to access Grid resources via Web portals and Web-based applications. Such users can be given pseudonym identities. For this purpose, a VO can maintain a repository of credentials each containing a unique pseudonym identity. The user could be granted access, for example, based on the payment and identification information obtained from their credit card transaction. If the user produces a valid credit card number then this number can be tied with the pseudonym identity. Based on the credential the user can be granted membership on a "Guest" role and the role can be assigned privileges according to the user's request and purchase amount. The implications on security would be that, such users would have to be assigned privileges on the fly, based on their purchase amount and resource requirements. In addition, their privacy may need to be protected while allocating the appropriate resources.

Often, in VOs users might be assigned specific tasks and there could be constraints related to the execution of those tasks. For example, a user may have access to data only during certain days of the week or certain tasks may be deemed mutually exclusive for a user, i.e. within a certain set of tasks, any two or more tasks cannot be executed at the same time. In certain instances, a user may wish to delegate only a subset of its rights to an application or Grid Service to act on its behalf. This requirement can usually arise in systems where there is a limited Trust relationship established between entities. For example, a user may contact a data mining service to mine certain data sets that the user has access to. If the trust between the user and the service is limited, then the user might want to delegate only a specific

subset of its rights to the service, thus enabling it to complete only the required task and nothing more. We believe that these and other security requirements of Data Grids enumerated in this paper can be met by framing and enforcing policies based on a dynamic access control model that enhances existing role-based access control (RBAC) models [ANSI03] and semantic-based access control (SBAC) models [Pan06].

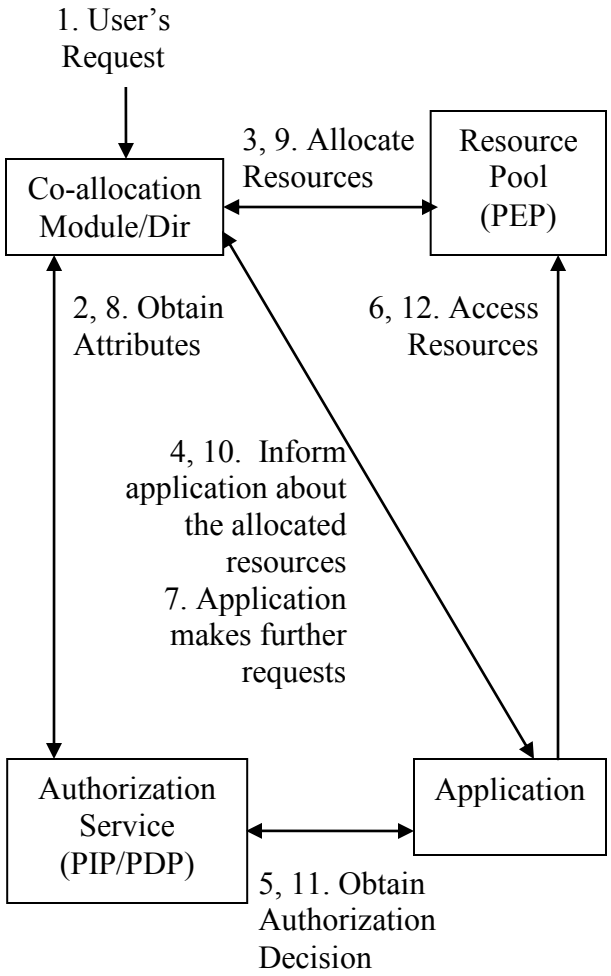

Figure 1: Dynamic Access Control for Data Grids

3. Dynamic Access Control for Data Grids

In this section we examine the requirements for a dynamic access control model for Data Grids. We hypothesize that schedulers/resource brokers and applications will use a co-allocation module for the allocation of resources. As shown in Figure 1, on receipt of a user's request the co-allocation module can obtain attributes for the user's rights, the computing environment and Security policies of resources from the authorization service. In this case, the authorization service acts as the Policy Information point (PIP). A PIP releases attributes to a requesting service or resource. The co-allocation module can then make intelligent decisions based on the user's request, location, and rights, and the availability and state of resources and the status of other jobs. It can maintain a Directory service to store information about the status of resources and jobs. Making these decisions is a significant problem and we do not offer a solution here. Instead we leave it as a topic for future work and assume that the module possesses the ability to make those decisions. The authorization service can be either centralized, for example the Community Authorization Service (CAS) [Pearlman02], or distributed, for example the Shibboleth Attribute Authorization Service (SAAS) [Welch05, Carmody01]. CAS can manage identity-based or role-based Security policies [Canon03] centrally by using a Relational Database which is the Policy Administration Point (PAP). A PAP is where the Security policies and Attributes are managed. Shibboleth can transfer the desired Attributes to a requesting Grid Service via the GridShib [Welch05] software interface. However, it needs to interface to a distributed privilege management system such as Signet [Welch05, Signet08] to manage Security policy in a distributed environment. In this case Signet would be the PAP.

Once the resources are allocated, an application can be informed of the allocated resources and can then be authorized access to each resource based on the attributes obtained directly from the authorization service (*push mode*). Alternatively, the resources can contact the authorization service to receive authorization information for the user and permit access to the application (*pull mode*). In this case, the authorization service acts as the Policy Decision Point (PDP) and each individual resource acts as the Policy Enforcement Point (PEP). A PDP gives the authorization decision to a requesting service, resource or application, and the PEP executes the decision of the PDP. The PDP receives the policies and attributes from the PAP. The terms PAP, PIP, PDP and PEP are defined within the eXtensible Access Control Markup Language (XACML) [OASIS03] authorization model. The authorization information can be in the form of attributes, such as the user's institutional affiliation or role in a VO. Furthermore during its runtime, the application can make requests for additional resources

3.1 Dynamic Role-based Access Control Models (RBAC)

Role-based access control (RBAC) models [ANSI03] are now widely replacing traditional Mandatory and Discretionary access control models. With RBAC, Security policies are expressed with respect to the roles in an organization (e.g. manager) rather than with respect to individual users, thus making Security policy management and enforcement more scalable. The ANSI RBAC model is defined in terms of four *model components*: Core RBAC, Hierarchical RBAC, Static Separation of Duty Relations, and Dynamic Separation of Duty Relations [ANSI03].

An authorization service can be used to assign users memberships on roles and take care that the role assignments do not violate the Static Separation of Duty (SSoD) constraints. This can be done by having the service maintain a set of mutually exclusive roles and verify the user assignments against this set.

Before activating a role a resource needs to ensure that the Dynamic Separation of Duty (DSoD) constraints are not violated. For this purpose, among a user's other attributes, a single bit can be maintained by the authorization service for each user. Every time a role in the DSoD set is activated for the user by a resource, this bit can be turned on. Any further role activation for the user by a resource can be made after checking the status of this bit.

Several sophisticated RBAC models that extend the ANSI RBAC model have been described in the literature [Bhatti05, Sandhu96]. However these models have several drawbacks when it comes to access control within Grid environments. In [Jin05] they propose an RBAC model for Grids. They assume each VO is an autonomous administrative domain and describes some levels of interaction between the domains. Further they assume each domain has its own access control mechanism. Their model specifies access control requirements only within a domain and does not take into consideration access control for entities belonging to multiple domains. However, in Grids, entities such as users and resources can belong to multiple VOs. Usually access control decisions have to be made by aggregating attributes from different VOs. Furthermore their model is a simplistic and cannot apply to all the cases in Grids.

The assignment of privileges to a role should be done dynamically. For example, if when the server load is low one set of privileges can be assigned to the local role and when the server load becomes high another set of privileges can be dynamically assigned. In [Zhang03], the use of a finite state machine has been proposed to switch role-privilege assignments according to the state of a resource.

In [Bhatti05], the authors formalize a model for access control in multi-domain environments by extending the ANSI RBAC model to define additional components such as administrative domain hierarchy, administrative users, roles and privileges. These components allow for the de-centralized administration of RBAC policies by allowing delegation of administrative privileges to administrators of sub domains. The authors further formalize temporal constraints on user-role assignments and role-permission assignments. In [Zhang03] they extend the ANSI RBAC model to allow dynamic user-role assignments and role-permission assignments with respect to changes in environmental context such as user location, security of the network through which access is made or the state of resources.

These models do not include constrained delegation and dynamic delegation which are important in Grids. A user should be able to dynamically delegate his/her rights to other users, applications and Grid services without administrative intervention. For example, a VO supervisor with access rights to a new resource may wish to delegate his/her rights to applications run by certain users. The Grid Security Infrastructure (GSI) of the Globus Toolkit [Foster99] achieves such delegation through the use of temporary proxy credentials which are generated based on user credentials [6]. However, it does not allow the specification of constraints on the delegation of rights. For example, a VO supervisor may be allowed to delegate his/her rights only to programmers, but not to operators.

3.2 Semantic-based Access Control (SBAC)

While addressing the important issue of scalability, RBAC does not address issues of information interaction and semantics of organization workflows. However, by virtue of being modular, RBAC allows more sophisticated access control models to be layered on it. Such models can include task-based access control (TBAC) [Zhang06] models and semantic-based access control (SBAC) models. Some models have been proposed in the literature [Zhang06, Pan06], but none can adequately address the security requirements that arise in Grid environments. SBAC models can deal with

issues like information inference and interaction from data flow within and between organizations, but they do not address scalability. By linking SBAC models with RBAC models this problem can be resolved. However, the resulting models have to be enhanced in order to take into account important requirements that arise in Data Grids, such as dynamic delegation of rights and dynamic resource allocation/de-allocation.

We propose the use of the RBAC profile of the eXtensible Access Control Markup Language (XACML) standard [OASIS05a] and Security Assertion Markup Language (SAML) [OASIS05b] based scoped attributes to express SBAC policies in Data Grids. The RBAC profile of XACML can form the basis for the expression of SBAC policies. However, the RBAC profile of XACML has several drawbacks for the access control in Data Grids. It does not address dynamic delegation of rights and dynamic user-role assignments.

3.3 Semantic-based Access Control (SBAC) Models and Policy Management

A SBAC model layered over the RBAC model has been proposed in [Pan06]. This model addresses Ontology mappings between the semantic scopes of data objects and concepts, and data translation. However, all these models have several drawbacks when it comes to access control within Grid environments. They are too simplistic and cannot address many security requirements such as constrained delegation and dynamic delegation of rights. In [Jin05] they propose an RBAC model for Grids. They assume each VO is an autonomous administrative domain and describes some levels of interaction between the domains. Further they assume each domain has its own access control mechanism. However, they do not take into consideration access control requirements for entities belonging to multiple domains. In Grids, entities such as users and resources can belong to multiple VOs. Usually access control decisions have to be made by aggregating attributes from different VOs.

In [Bhatti05], the authors formalize a model for access control in multi-domain environments by extending the ANSI RBAC model to define additional components such as administrative domain hierarchy, administrative users, roles and privileges. These components allow for the de-centralized administration of RBAC policies by allowing delegation of administrative privileges to administrators of sub domains. The authors further formalize temporal constraints on user-role assignments and role-permission assignments. In our view, the administrative domain hierarchies and other components defined in [Bhatti05] can be mapped to VOs and their sub-groups. This is because VOs and their subgroups form partially ordered domain hierarchies. In [Zhang03] they extend the ANSI RBAC model to allow dynamic user-role assignments and role-permission assignments with respect to changes in environmental context such as user location, security of the network through which access is made or the state of resources.

The SBAC and RBAC models do not address constrained delegation and dynamic delegation of rights which are important in Grids. A user can dynamically delegate his/her rights to other users, applications and Grid services without administrative intervention during their runtime. The Grid Security Infrastructure (GSI) of the Globus Toolkit achieves such delegation through the use of temporary proxy credentials which are generated based on user credentials [Butler00]. However, it does not allow the specification of constraints on the delegation of rights. For example, a VO manager may be allowed to delegate his/her rights only to supervisors, but not to other employees. To address constrained and dynamic delegation in the SBAC model based on RBAC, the model in [Pan06] can be enhanced by the use of delegation roles with partial administrative and delegation privileges coupled with delegation constraints. The model can also be enhanced to specify mapping rules between VO roles and local roles, and constraints on those roles. We identify the basic components of a SBAC model for Data Grids as shown in Figure 2. Privileges are associated with concepts instead of being directly associated with objects. An object could be a particular relational table, a database, a portion of network bandwidth or CPU time. In [Pan06] they define the semantic scopes of objects and concepts, and then use them to define the relations used in Ontology. Ontology represents a set of concepts within a domain and the relationships between those concepts, and is used to reason about the objects within that domain. In [Pan06], Ontology mapping rules between the semantic scopes of objects and concepts, and between concepts themselves are specified. The authors assume the semantic scope of each object is narrower than or equivalent with the semantic scope of one or more concepts in the Ontology. Ontology can comprise of a set of concepts that have different

types of relations among them: {equivalentClass, subClassOf, disjointWith, ...}. The meaning of semantic scope is quite intuitive. For, example the semantic scope of object "Ford Car VIN ID 123456789" is narrower than that of concept "car"; the semantic scopes of "car" and "truck" being disjoint means there exists no instance that belongs to both concepts. Ontology and standards are required to establish the semantic relations between entities (users, applications and services), information and workflows within VOs, and diverse security mechanisms and policies.

The work in [Pan06] addresses heterogeneity in databases across multiple organizations, but has to be extended to address information that can be generated on-the-fly (such as that which could be produced by a data mining service) and VO workflows (such as a business process comprising of interaction between different Grid services).

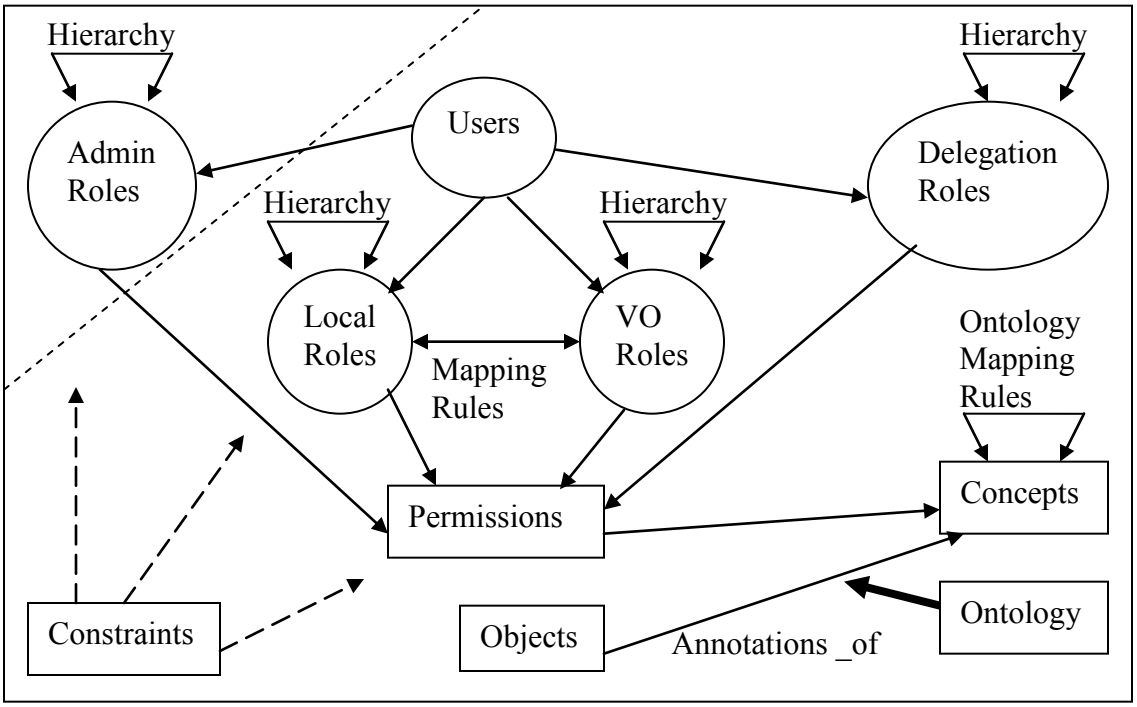

Figure 2: Basic Components of SAC model for Data Grids.

This can be done with the incorporation of task-based access control models [Zhang06] which allow for the graphical representation of data and information workflows. The model in [Pan06] can also be further enhanced to specify mapping rules between VO roles and local roles, and constraints on those roles.

We propose the use of the RBAC profile of the XACML standard [OASIS05a] as the base to express SBAC policies. The RBAC profile of XACML is not designed to specify the semantic scope of objects and concepts, but by incorporating SAML [OASIS05b] based scoped attributes this problem can be addressed. A scoped attribute is a combination of a value and its scope. Scope identifies the domains and sub-domains in which the values are defined. We will express the semantic relation between objects and concepts through the values and their scope contained within the scoped attributes. For example, a value "Ford Car VIN ID 123456789" with scope "car" can mean that the semantic scope of the object "Ford Car VIN ID 123456789" is contained within the semantic scope of the concept "car". XACML allows the specification of rules and conditions which can be used to specify Ontology mapping rules containing relations such as "subClassOf" or "disjointWith". The XACML profile does not directly support scoped attributes. Mapping SAML to XACML allows the systems using XACML to store SAML attributes [OASIS05b]. For richer Semantics, XACML can be linked with the Web Ontology Language (OWL) [W3C04] which has attracted wide interest in industry and academia.

4. Conclusion

In this paper, we examined the framework for a dynamic access control model for Data Grids. This framework take into account that authorization decisions should be made not only by considering the user's request and the pool of available resources, but also the user's access rights and the security policy of resources. Furthermore, access control decisions should be made dynamically during an applications runtime. Future work would involve a formal specification and implementation of the dynamic access control model.

References

[ANSI03] *American National Standard for Information Technology — Role Based Access Control*, Secretariat of Information Technology Industry Council (ITI), http://csrc.nist.gov/rbac/rbac-std-ncits.pdf, 2003.

[Bhatti05] R. Bhatti, B. Shafiq, E. Bertino, and A. Ghafoor, "X-GTRBAC Admin: A Decentralized Administration Model for Enterprise-Wide Access Control," *ACM Trans. on Information and System Security*, vol. 8, no. 4, pp. 388–423, 2005.

[Bertino04] E. Bertino, P. Mazzoleni, B. Crispo, S. Sivasubramanian and E. Ferrari, "Towards supporting Fine-Grained Access Control for Grid Resources," *Proc. of 10th Int'l Workshop on Future Trends in Distributed Computing Systems*, 2004.

[Butler00] R. Butler et al., "A National-Scale Authentication Infrastructure," *IEEE Computer*, vol. 33, no. 12, pp. 60–66, 2000.

[Canon03] Shane Canon, Steve Chan, Doug Olson, Craig Tull, and Von Welch, "Using CAS to manage role-based VO sub-groups," *Proc. of Computing in High Energy Physics*, 2003.

[Carmody01] S. Carmody, "Shibboleth Overview and Requirements," Shibbololeth Working Group Document, available at http://shibboleth.internet2.edu/docs/draft-internet2-shibboleth-requirements-01.html, 2001.

[Chervenak01] A. Chervenak, I. Foster, C. Kesselman, C. Salisbury and S. Tuecke, "The Data Grid: Towards an Architecture for the Distributed Management and Analysis of Large Scientific Datasets," *Int'l Journal of Network and Computer Applications*, vol. 23, no. 3, pp. 187–200, 2001.

[Foster99] I. Foster and C. Kesselman, "The Globus Toolkit," in *The Grid: Blueprint for a New Computing Infrastructure*, I. Foster and C. Kesselman (Eds.), Morgan Kaufman, pp. 259–278, 1999.

[Foster01] I. Foster, C. Kesselman and S. Tuecke, "The Anatomy of the Grid: Enabling Scalable Virtual Organizations," *Int'l Journal of Supercomputer Applications and High-Performance Computing*, vol. 15, no. 3, pp. 200–222, 2001.

[Foster03] I. Foster and R. L. Grossman, "Data Integration in a Bandwidth-Rich World," *Communications of the ACM*, vol. 46, no. 11, pp. 51–57, 2003.

[Jin05] H. Jin, W. Qiang, X. Shi and D. Zou, "RB-GACA: A RBAC Based Grid Access Control Architecture," *Int'l Journal of Grid and Utility Computing*, vol. 1, no. 1, pp. 61-70, 2005.

[OASIS03] *Extensible Access Control Markup Language (XACML) v1.0*, Organization for the Advancement of Structured Information Standards (OASIS), http://www.oasis-open.org/committees/xacml/, 2003.

[OASIS05a] *Core and hierarchical role based access control (RBAC) profile of XACML v2.0*, Organization for the Advancement of Structured Information Standards (OASIS), http://docs.oasis-open.org/xacml/2.0/access_control-xacml-2.0-rbac-profile1-spec-os.pdf, 2005.

[OASIS05b] *SAML 2.0 profile of XACML v2.0*, Organization for the Advancement of Structured Information Standards (OASIS), http://docs.oasis-open.org/xacml/2.0/access_control-xacml-2.0-saml-profile-spec-os.pdf, 2005.

[Pan06] C. Pan, P. Mitra, and P. Liu, "Semantic Access Control for Information Interoperation," *Proc. of the 11th ACM Symposium on Access Control Models and Technologies*, pp. 237–246, 2006.

[Pearlman02] L. Pearlman, V. Welch, I. Foster, C. Kesselman, and S. Tuecke, "A Community Authorization Service for Group Collaboration",

Proc. of the 3rd IEEE International Workshop on Policies for Distributed Systems and Networks, 2002.

[Sandhu96] R. Sandhu, E.J. Coyne, H.L. Feinstein, and C.E. Youman, "Role Based Access Control Models," *IEEE Computer*, vol. 29, no. 2, pp. 38–47, 1996.

[Signet08] http://middleware.internet2.edu/signet/index-wg.html, 2008

[W3C04] *OWL Web Ontology Language Guide*, World Wide Web Consortium (W3C) Recommendation, http://www.w3.org/TR/owl-guide, 2004.

[Welch05] V. Welch, T. Barton, K. Keahey, and F. Siebenlist, "Attributes, Anonymity, and Access: Shibboleth and Globus Integration to Facilitate Grid Collaboration," *Proc. of the 4th Annual PKI R&D Workshop*, 2005.

[Zhang03] G. Zhang and M. Parasher, "Dynamic Context-Aware Access Control for Grid Applications," *Proc. of the 4th Int'l Workshop on Grid Computing*, pp. 101–108, 2003.

[Zhang06] C. Zhang, Y. Hu, and G. Zhang, "Task-Role Based Dual System Access Control Model," *Int'l Journal of Computer Science and Network Security*, vol. 6 no. 7B, pp. 211–215, 2006