Towards Secure Information Sharing and Management in Grid Environments†

Jing Jin and Gail-Joon Ahn
University of North Carolina at Charlotte
{jjin,gahn}@uncc.edu

Abstract— **The introduction of service-oriented paradigm in Grid and corresponding Web Services standards has recently demanded the evolution of access control solutions to support fine-grained authorization requirements and dynamic natures derived from Grid environments. In this paper, we present a role-based authorization infrastructure for data sharing and management services in Grid. Our infrastructure advocates a dynamic and flexible resource-centric authorization approach with features of distributed role-based access control and systematic delegation of administrative authority. The infrastructure seamlessly integrates the existing XACML-based policy framework and authorization services in the RAMARS framework. We discuss our proof-of-concept prototype system that supports Web Services and SAML based authorization assertions. We also describe how the framework can be deployed in being compatible with an Open Grid Service Architecture.**

I. Introduction

Grid technologies have evolved towards an integrated Open Grid Services Architecture (OGSA) enabling transparent access to Grid resources across distributed virtual organizations (VOs) [6], [7], [8]. As a result, a service-oriented approach has been widely adopted by the Grid community to provide a common representation for Grid resources. Resources in VOs are uniformly treated as Grid services in the form of Web Services that provide a set of well-defined interfaces for dynamic service creation, resource discovery, lifetime management, and so on [7].

In order to accommodate such an emerging service-oriented trend, Grid security solutions need to be evolved to ensure interpretable implementations with the OGSA Security Architecture [15], [22]. Among various security functionalities in OGSA, authorization services play an important role for both service requestors and service providers to specify fine-grained authorization policies and further resolve policy-based access control decisions. On the one hand, the authorization model has to be designed specifically for the targeted Grid service to address its unique access control requirements. On the other hand, it is required from deployment and management perspectives that authorization services be pluggable to Grid applications via well-defined protocols and interfaces [22]. Recently, the Global Grid Forum (GGF) has proposed a SAML AuthZ API [25] using SAML [16] as a standard message format for requesting and expressing authorization assertions. However, the feasibility and suitability of this mechanism for Grid services need further evaluations through explorations and practical experiments.

Grid resources are diverse in their locations, structures, ownerships, access mechanisms and capabilities. From the viewpoint of service-oriented architecture, such complexities should be made transparent to clients through a layer of virtualization service. The Grid data sharing and management service is designed to provide such a unified platform for dynamic discovery of data sources, federated and controlled access to the distributed data, and dynamic data replication [21]. As one of the application development efforts, the Storage Resource Broker (SRB) was introduced as a middleware application to manage a variety of distributed storage resources including file systems, archives and digital libraries [2], [20]. While the SRB provides effective data abstraction and virtualization, its common method for enforcing access control is primitively an ACL-based discretionary access control approach [2]. This approach, however, does not provide a fine-grained control for a dynamic Grid environment and is neither scalable nor flexible in practice.

The Role-based Access Management for Ad-hoc Resource Sharing framework (RAMARS) was originally proposed as a generic approach to achieve effective access control for the resource sharing in heterogeneous collaborative environments [11], [12]. RAMARS advocates a dynamic and flexible resource-centric authorization approach with features of distributed role-based access control and systematic delegation of administrative authority. In this paper, we extend this framework to demonstrate how RAMARS can be adapted for collaborations in Grid environments enabling secure data sharing and management services. The infrastructure integrates existing XACML-based policy framework and the authorization service in RAMARS framework so that the resource owner defines fine-grained authorization policies. And the designated authorization service derives and enforces access decisions on the resource owner's behalf. We also discuss our proof-of-concept prototype system that supports Web Services and SAML based authorization assertions. We then describe how the framework can be deployed in being compatible with OGSA by exploring standard network interfaces and SAML AuthZ APIs for requesting and expressing authorization messages [25].

The rest of this paper is organized as follows. In Section II,

†All correspondence should be addressed to: Dr. Gail-Joon Ahn, Software and Information Systems Department, College of Computing and Informatics, University of North Carolina at Charlotte, 9201 University City Blvd., Charlotte, NC 28223; email:gahn@uncc.edu.

1-4244-0429-0/06/$20.00 ©2006 IEEE

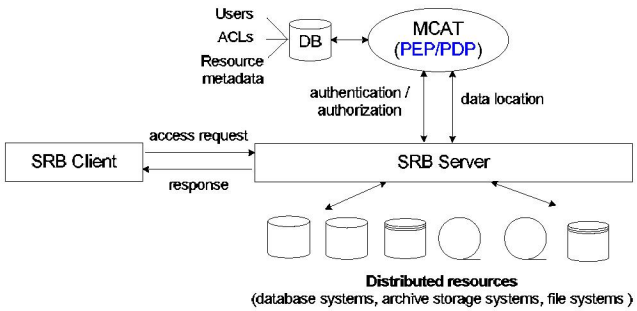

Fig. 1. SRB system overview

we give an overview of the SRB application from which we analyze access control requirements involved in Grid data sharing and management services. We then present the RAMARS framework with its policy framework and major system components. In Section III, we introduce the integrated system design and prototype implementation for RAMARS as an external OSGA authorization service. Brief discussions relevant to system integration issues are described in Section III as well. Section IV presents related works that deal with authorization issues in Grid environments. Section V concludes the paper with future research directions.

II. Background Technologies

A. The Storage Resource Broker and authorization requirements

The SRB system developed by San Diego Supercomputing Center is one of the Grid applications to manage data sharing across various storage devices and institutions [2], [20]. It provides transparent access and management capabilities to Grid data stored in file systems, digital libraries and databases across domains by supporting a logical abstraction of those physical data instances and a set of uniformed operations. The SRB system is a distributed system comprised of three major components: the Metadata Catalogue (MCAT) database, the SRB Server and the SRB Client. The MCAT database is a centralized repository that provides a mechanism for storing the metadata used by the system, including the physical and logical details of the data and its replicas, the user information, and the authorization policies in ACLs. The SRB Server is a middleware application that accepts requests from clients and obtains the necessary datasets. In particular, it queries the MCAT database to gather the authorized datasets and supplies the results back to the client. The SRB Client is an end-user tool that provides a user interface to send requests to the SRB server. Figure 1 shows a high level architecture of the SRB system.

From the authorization perspective, the MCAT serves both as a centralized Policy Decision Point (PDP) that implements an ACL-based discretionary access control and as an Policy Enforcement Point(PEP) that enforces the authorization decisions by controlling the user access to the physical data instances through the SRB server. An SRB account must be created for each user in the MCAT database before the user can access/create SRB datasets. And the dataset owner is responsible to authorize these registered users with a set of pre-defined operations on per-user basis. When the SRB is used in conjunction with the GSI (Grid Security Infrastructure) authentication [15], Gridmap files [23] have to be maintained in MCAT to map between a list of authenticated GSI DNs and their equivalent SRB user accounts. As can be seen, this method of access control is not suitable for the ever-changing users and resources in Grid VOs. It neither allows the resource owner to set a fine-grained policy defining who is allowed to do what, nor does it minimize his management workload. On the contrary, it is an intimidating work to pre-configure the SRB with every user in Grid VOs who is to be authorized to access the data resources. And the situation is getting worse when both users and resources belong to multiple distributed VOs.

Based on the above analysis, we identify several functional requirements that authorization systems should address not only for the SRB system alone, but also for the Grid data sharing and management services in general. Firstly, the authorization service needs to provide effective model and mechanisms for resource owners–called *originators*, to manage user credentials derived from GSI. As users and resources dynamically evolve in VOs, abstractions of users and privileges are needed to achieve the flexibility and manageability. Secondly, delegation should be in place to leverage a systematic way of distributing and propagating resource administration to trusted authorities. Thirdly, the authorization infrastructure should be configured to automatically derive and enforce authorization decisions based on originators' authorization policies without requiring their intervention. Finally, in accommodating the service-oriented trend, it is desirable that the authorization component be deployed as a Grid security service allowing it to be located and used as needed by applications through standard protocols and authorization message exchanges.

B. RAMARS framework

RAMARS framework [11], [12] has been proposed as a generic solution for the resource sharing in distributed collaborative environments. RAMARS advocates a resource-centric approach to determine the scopes of collaboration relationships without assuming any centralized administrative points. In addition, RAMARS applies a flexible role-based approach to enable resource originators to authorize distributed collaborators and control over the resources being shared. The Delegation of Delegation authority (*DoD*) is introduced to systematically achieve user-role assignments in distributed environments.

The role-based approach provides an effective way to abstract privileges using roles. Instead of including every individual user, an originator could simply define his collaborative sharing domain in a collection of resource sharing roles that are independent from any VO definitions, such as "engineer" and "investigator". And each peer collaborator is dynamically included in the sharing domain to gain access privileges by claiming their roles. To identify and assign users to these

resource sharing roles, RAMARS introduces a special type of administrative delegation–*DoD*, as another layer of authority decentralization to achieve the distributed role assignment. Through this particular delegation, an originator could partially delegate the role assignment authority to other trusted third-party authorities. The scheme can be easily adapted into Grid environments by delegating the user-role assignment to centralized VO management authorities or services.

The proposed ideas are realized in a set of XACML-based [18] authorization policy schemas that consist of the following components:

- Root Meta Policy Set (RMPS) is a top-level policy that specifies static resource attributes and the originator(s). The originator's authorization policies are independently defined and located through policy references. This separation enables the distributed policy deployment in RAMARS system architecture.
- Role-based Originator Authorization Policy Set (ROA) specifies real authorization policies defined by an originator. An originator defines the sharing roles for the specific resource, and delegates fine-grained access capabilities to these roles. The direct user-role assignment and the user-role assignment authority delegation are also specified. These are specified in the following sub-policies:
  - Role Policy Set (RPS) specifies a set of collaborator roles in an originator's collaborative sharing domain. Each role is represented as one RPS sub-policy. The permission-role assignment is supported by referencing each RPS sub-policy to a corresponding capability policy (CPS).
  - Capability Policy Set (CPS) specifies the actual capabilities associated with a given role. The role hierarchy can be achieved through capabilities aggregation by referencing the CPS of a senior role to the CPS of its junior role.
  - Delegation of Delegation Policy Set (DoDPS) specifies the delegation of delegation authority (*DoD*) relation so that an originator can delegate the user-role assignment authority to trusted third parties, such as trusted VOs.
  - Role Assignment Policy Set (RAPS) specifies the direct user-role assignment performed by the originator.
- DoD Role Assignment Policy Set (DoD RAPS) specifies the delegated user-role assignment initiated by the authority of each *DoD* delegatee.

In the RAMARS system design, the RMPS policy should be associated with the data resource. The originator's ROA policies and the *DoD* delegatee's RAPS policies are encapsulated in X.509 attribute certificates and maintained separately in one or more LDAP repositories across domains. In order to reduce the complexity of setting authorization policies, a Policy Editor toolkit is developed in helping originators and *DoD* delegatees articulate authorization policies, create policy attribute certificates and store them in their LDAP repositories.

The architecture of RAMARS framework can be segregated into two domains. In the administration domain, resource originators and *DoD* delegatees edit and maintain their authorization policies by using the Policy Editor toolkit. These policies are expected to be automatically evaluated and enforced by the authorization system. In the authorization domain, upon receiving an access request, the PEP formulates and sends an access request along with the root RMPS policy. The PDP implements a "pull" model that retrieves policies based on the locations referenced in the RMPS policy. The PDP then evaluates the request against applicable policies, makes an access decision and sends it back to the PEP for decision enforcement. In the RAMARS prototype system [12], XACML-based requests and responses are implemented as basic message formats that are exchanged between the PEP and PDP. Also, the PDP is implemented only for local interactions and does not support network services. In the subsequent section, we present how an extended RAMARS infrastructure with the notion of Web Services provides authorization decisions for Grid services through uniformed web interfaces and GGF's AuthZ SAML APIs [25].

III. System Design and Implementation Details

SAML specification [16] was proposed to provide a general purpose language as well as a communication mechanism for specifying and conveying security-related information between clients and servers regarding authentication, authorization decisions and attributes. With the emergence of OGSA and Grid services, the SAML authorization callout is considered as an authorization solution to enable legitimate services over an exposed Grid service handler. In particular, the Grid service as the client, sends a SAML authorization decision *Request* to the authorization service, and in return receives a SAML *Response* with authorization decisions over a network channel [25]. In the design of OGSA, the authorization system becomes yet another "service" to provide authorization decisions as the PDP through standard web interfaces and SAML message exchanges. In this section, we discuss how RAMARS infrastructure can be extended to provide such an OGSA compatible authorization service.

A. Extensions to SAML

The SAML authorization specification [16] defines a rudimentary *AuthorizationDecisionQuery* (also called *Query*), flowing from the PEP to PDP with an assertion returned with an *AuthorizationDecisionStatement* (also called *DecisionStatement*). However, the original SAML *Query* is unable to convey additional information that the PDP may use to retrieve the user attributes and render a decision. In addition, the SAML *DecisionStatement* returns a complete list of allowed actions instead of a simple `permit/deny` decision, which may significantly affect the performance of the PEP in parsing the response. Therefore, we adopt two GGF SAML extensions [25] to meet the OGSA authorization requirements.

- *ExtendedAuthorizationDecisionQuery*: it allows the PEP to specify whether a simple or full authorization decision

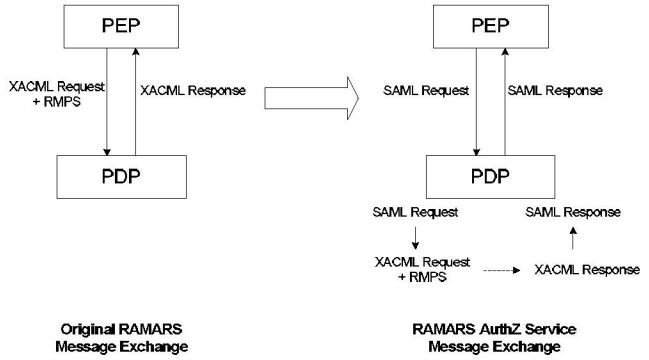

Fig. 2. Message exchanges extension

needs to be returned by the PDP as a boolean value of the *RequestSimpleDecision* attribute. In addition to the common elements specifying the requested *Subject*, *Resource* and *Action*, a *SubjectAttributeReferenceAdvice* element is added allowing the PEP to convey additional information such as where the PDP may obtain the subjects's attributes in the "pull" mode of operations.

- *SimpleAuthorizationDecisionStatement*: it defines a simpler response by removing the enumeration of whole authorized actions, which in turn could improve the efficiency for the PEP to process the response. The return of this simpler response is determined by the boolean value of the *RequestSimpleDecision* attribute specified in the PEP's request.

B. Extensions to RAMARS

The efforts towards building an OSGA compatible RAMARS authorization service lie in two major extensions: the extended message exchange between the PEP and PDP, and the network interface implementation of RAMARS PDP.

In order to support the SAML messages exchanged between the PEP and PDP, the extensions have to be made on the original RAMARS system architecture. As shown in Figure 2, within the OGSA authorization service scheme, the PEP has to convey all information to PDP in a single *ExtendedAuthorizationDecisionQuery* encapsulated in the SAML *Request*. On the PDP side, the PDP has to understand such SAML *Request* and return the access decision in a SAML *Response*. In achieving these procedures while making minimum changes to the system, two message conversions are introduced on the PDP side. On the one hand, the XACML *Request* and the RMPS policy have to be reconstructed from the SAML *Request*. On the other hand, the XACML *Decision* drawn from the PDP engine has to be converted to a SAML *Response* with the form of *SimpleAuthorizationDecisionStatement*. In our mapping schema, elements of *Subject*, *Resource* and *Actions* in SAML *ExtendedAuthorizationDecisionQuery* are transformed accordingly to the elements in the XACML *Request*. The elements of *Resource* and *SubjectAttributeReferenceAdvice* are transformed into the RMPS policy. For the *Response* transformation, the *Decision* value in the XACML *Response*

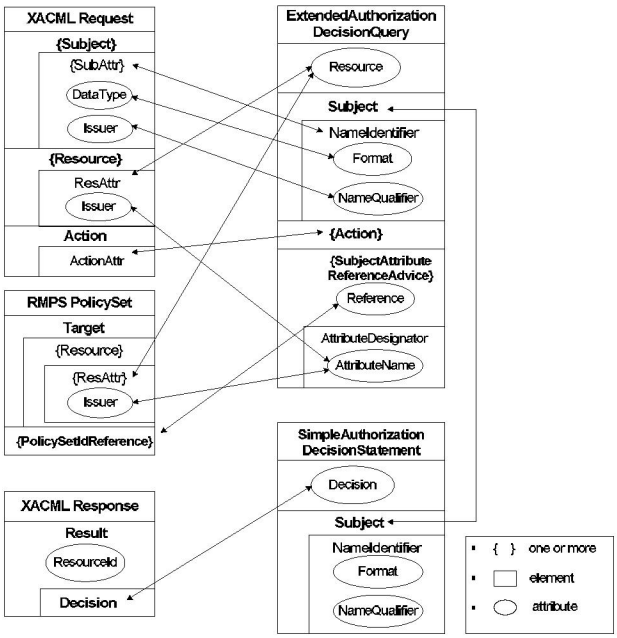

Fig. 3. Message mapping schema

is encoded as the *Decision* attribute in the SAML *SimpleAuthorizatinDecisionStatement*. Figure 3 illustrates the mapping schemas.

In implementing the OGSA authorization service of RAMARS, a network-based interface of RAMARS PDP needs to be visible for Grid applications. As shown in Figure 4, the RAMARS PDP implements a network interface containing one method called *authorize*. It accepts a standard SAML *Request* and returns a SAML *Response*. For the security concern of this method, the messages are encapsulated in SOAP bindings and transferred over SSL protocol.

Figure 5 illustrates the detailed sequence of authorization procedures in RAMARS authorization service, named RAMARS AuthZ. The PEP formulates a SAML *Request* with an *ExtendedAuthorizationDecisionQuery* including the requester's X.509 identity, the actions towards the resource and the locations of originators' ROA policies. This SAML *Request* is sent to the PDP through its network interface–AuthZ Intf. Once the PDP's network interface obtains the request, the first step is to parse the SAML *Request* and gather information to the RAMARS XACML policy engine. In our design, this functionality is implemented in the Context Handler, which is part of the RAMARS AuthZ service itself. The Context Handler analyzes the request to construct the RMPS policy and the XACML *Request* based on an XSL stylesheet that implements the above mentioned mapping schemas. Then the Context Handler contacts the LDAP directories specified in the RMPS to retrieve originators' authorization policies embedded in attribute certificates (ACs). These ACs are validated and parsed by the Context Handler. If the originator defines the *DoD* policy for the user-role assignment, the delegatee's policy ACs are retrieved and parsed as well by the Context Handler.

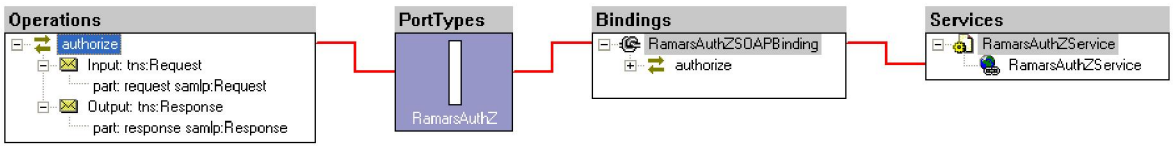

Fig. 4. WSDL of the RAMARS authorization service

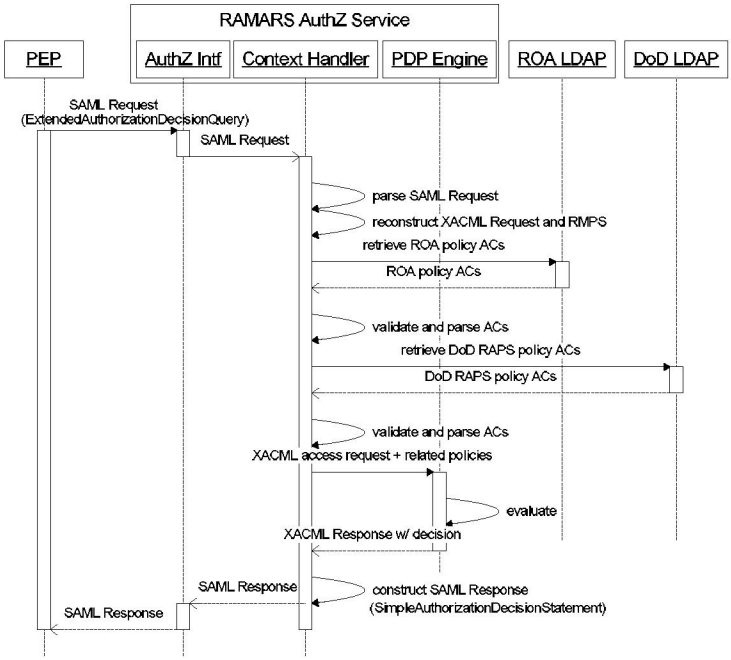

Fig. 5. Sequence diagram for extended RAMARS infrastructure

The next step is to leverage the policy engine to render the request against the originators' policies. The core PDP engine evaluates the applicability of the request against the available policies. Matched policies are then evaluated to determine a positive or negative decision (i.e. `permit` or `deny`). This result, in an XACML *Response*, is converted to a *SimpleAuthorizatinDecisionStatement* and encapsulated in a SAML *Response* by the Context Handler. The response is finally sent back to the PEP for the decision enforcement through the AuthZ Intf. As we may see, the Context Handler serves as a SAML-XACML translator while the major components developed in the original RAMARS is not necessarily to be changed.

In our RAMARS AuthZ service implementation, the authorization service is deployed in the Apache Axis SOAP engine working with Jakarta Tomcat web server. We implemented the SAML extensions based on an open source SAML implementation, called OpenSAML [10]. With Web Services, we do not assume the client side uses Java technologies and understands Java objects implemented in OpenSAML. Instead, in exchanging the SAML requests and responses, the RAMARS AuthZ service implements the *document*-style Web Services which is capable to exchange complex SAML *Request*/*Response* in the form of XML documents over the network rather than simple data types such as "strings" in the RPC-style. In particular, the RAMARS PEP sends the SAML *Request* to the RAMARS AuthZ Service as an XML document. There is no direct mapping between the SAML objects implemented in OpenSAML and the values in XML documents, while the PEP and RAMARS AuthZ Context Handler take care of mappings of the XML data values.

C. Discussions

1) Integration of SRB and RAMARS AuthZ service: The Web Services based PDP engine and standardized SAML callouts have enabled the RAMARS to serve as an external authorization service for the SRB system. Figure 6 depicts an integrated architecture of SRB data sharing system with RAMARS authorization service. The RAMARS PEP is a loadable module that should be integrated with the MCAT service in the SRB system. It replaces the existing ACL-based authorization functionality on the resources maintained in the SRB system. The PEP module interacts with the RAMARS PDP through its Web Services interface. SAML requests and responses are exchanged in standard SOAP envelops over the SSL protocol. In addition, the resource originators are able to edit and maintain their personalized authorization policies using the RAMARS Policy Editor through its web-based interfaces.

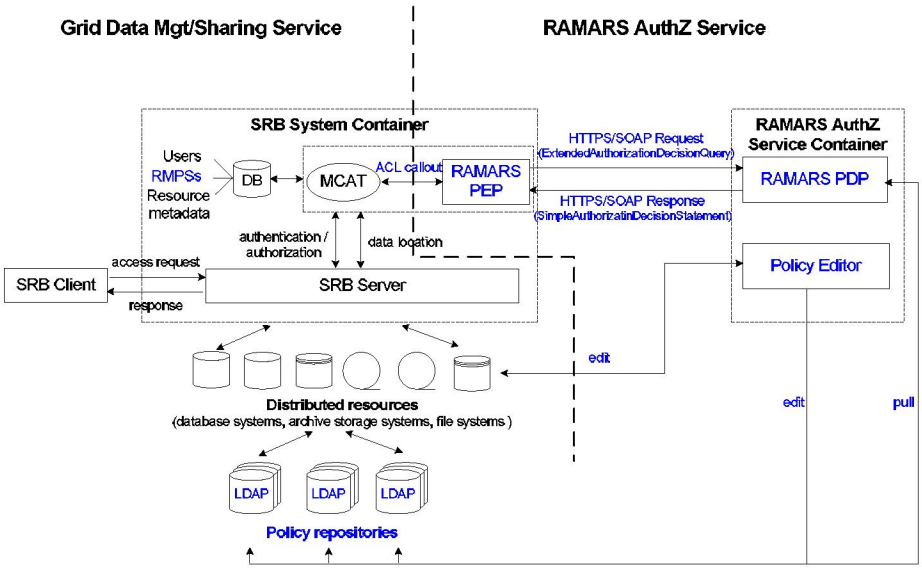

Fig. 6. SRB-RAMARS integration overview

2) Obligated authorization decisions: Obligations are a set of instructions provided associated with an authorization decision statement or response. These instructions may be targeted at the PEP to describe how a requested service, if allowed, should be confined and monitored during its execution. In the XACML policy model, these obligations are encoded in the XACML *Response* element along with the `permit/deny` authorization decisions. As our XACML-based policy schema and PDP engine are able to specify and process such obligations, we omitted it in our RAMARS AuthZ service. Implementing such obligations would result in finer-grained policies, yet it requires higher mutual confidence between the PDP and PEP in understanding and translating these obligations, which in turn, reduces the interoperability of the authorization service. Due to this reason, the current GGF SAML specification does not recommend such SAML assertions with obligations either. More research efforts and experiments have to be dedicated in exploring new SAML extensions to achieve this goal [1].

IV. Related Works

As the Grid mapfile approach for security in Globus toolkit [23] has various limitations in the flexibility and scalability, many alternative solutions such as Globus Community Authorization Service (CAS) [19], Virtual Organization Membership Service (VOMS) [1], and PERMIS [5], [4], [3] have been proposed in the Grid community.

The CAS [19] framework segregates the administration of resources from the administration of Grid communities. Every Grid community instantiates a CAS server representing that community and controlled by a community administrator. The community administrator manages fine-grained authorization permissions among community users based on the community-specific trust relationships. Community members can access resources by obtaining individual credentials in the form of X.509 proxy certificates [9]. Another similar community-based authorization framework is realized in VOMS [1]. The VOMS-based system differs from the CAS framework in its representation of the community privileges with attribute certificates rather than X.509 proxy certificates. However, what neither VOMS nor CAS provides is the ability for the resource originator to define authorization policies while letting the authorization infrastructure enforce the policies on his behalf. In addition, both CAS and VOMS implement the "push" model that an extra burden has been laid upon Grid users to contact the CAS or VOMS servers for authorization credentials and push them to the resources.

PERMIS [5] is based on Privilege Management Infrastructure that leverages the role based access control. The RBAC policy, in a self-defined XML format, is used to control access to all resources within the policy domain and is composed of a number of sub-policies. In PERMIS, policies and user attributes are held in attribute certificates. PERMIS provides a PDP that reads in the authorization policies and makes access control decisions. While PERMIS maintains policies in a centralized repository that may suffer from a single point of failure, our RAMARS approach enables a more robust distributed policy deployment scheme, and the delegation of administrative authority achieves more flexibility in providing authorization in distributed environments. As discussed in [4], [3], PERMIS infrastructure has recently been applied in Grid environments to provide authorization services by implementing same SAML extensions.

In [14], the authors proposed a design of resource broker to integrate policy evaluation with scheduling to achieve fine-grained access control in large Grids. The integration at the resource scheduling layer could prevent users from sending

[1]For example, OASIS has proposed a possible extension of SAML messages to directly encapsulate XACML requests and/or responses [17].

unauthorized jobs to the resource allocated by the scheduler. While our RAMARS AuthZ service is mainly focusing on providing authorization decisions to Grid applications on the job execution phase, we consider the resource broker as a complementary work as it handles different phases of a Grid lifecycle.

V. Conclusion and Future Work

In this paper, we have presented the RAMARS authorization infrastructure that addresses the generic access control requirements involved in Grid data sharing and management services. The infrastructure proposed a versatile role-based approach with systematic administrative authority delegation to achieve the manageability of administrative tasks. Therefore, it enables the originator to authorize and protect data resources in Grid services. Our rule-based policies, as those encoded in the associated XACML policy framework, have overcome the limitations of the ACL-based approach in the SRB system. The distributed architecture in RAMARS separates the application dependent access control PEP from a policy controlled and application independent PDP. Hence, we can support adaptive authorization services while there is no need to modify applications as the authorization policies changes. Our practice in implementing Web Services and SAML standards for exchanging security tokens has demonstrated that the scalability and suitability of the RAMARS as an advanced authorization infrastructure to serve the Grid services in VOs.

In our future work, we will further explore the interoperability of our RAMARS AuthZ service to fully integrate with other Grid data sharing systems, such as OGSA-DAI [13] and GridFTP [24]. In addition, performance will be analyzed to evaluate the overhead introduced in the policy retrieval and the authorization message exchanges via SAML and SOAP.

Acknowledgments

The work was partially supported by the grants from National Science Foundation (NSF-IIS- 0242393) and Department of Energy Early Career Principal Investigator Award (DE-FG02-03ER25565).

References

[1] R. Alfieri, R. Cecchini, V. Ciaschini, L. dell'Agnello, A. Frohner, A. Gianoli, L. Lorentey, and F. Spataro. VOMS, an Authorization System for Virtual Organizations. In *Proceedings of 1st EuropeanAcross Grids Conferences*, 2003.

[2] C. Baru, R. Moore, A. Rajasekar, and M. Wan. The SDSC Storage Resource Broker. In *Proceedings of CASCON'98*, 1998.

[3] D. Chadwick. Authorisation in Grid Computing. *Information Security Technical Report*, 10(1):33–40, 2005.

[4] D. Chadwick, S. Otenko, and V. Welch. Using SAML to Link the GLOBUS Toolkit to the PERMIS Authorisation Infrastructure. In *Proceedings of Eighth Annual IFIP TC-6 TC-11 Conference on Communications and Multimedia Security*, Windermere, UK, 2004.

[5] D. W. Chadwick and A. Otenko. The PERMIS X.509 role based privilege management infrastructure. In *Proceedings of the 7th ACM symposium on Access control models and technologies (SACMAT)*, 2002.

[6] I. Foster and C. Kesselman. *The Grid: Blueprint for a New Computing Infrastructure*, chapter Globus: A Toolkit-Based Grid Architecture, pages 259–278. Morgan Kaufmann, 1999.

[7] I. Foster, C. Kesselman, J. Nick, and S. Tuecke. The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration. In *Open Grid Service Infrastructure WG, Global Grid Forum*, June 2002.

[8] Globus. The Open Grid Services Architecture. http://www.globus.org/ogsa/.

[9] R. Housley, W. Polk, W. Ford, and D. Solo. Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile. RFC3280, http://rfc.net/rfc3280.html, 2002.

[10] Internet2. OpenSAML - an Open Source Security Assertion Markup Language implementation. http://www.opensaml.org/.

[11] J. Jin and G.-J. Ahn. Policy-Driven Access Management for Ad-hoc Collaborative Sharing. In *Proceedings of 2nd International Workshop on Pervasive Information Management (PIM 2006)*, 2006.

[12] J. Jin and G.-J. Ahn. Role-based Access Management for Ad-hoc Collaboration. In *Proceedings of 11th Symposium on Access Control Models and Technologies (SACMAT06)*, 2006.

[13] K. Karasavvas, M. Antonioletti, M. Atkinson, N. C. Hong, T. Sugden, A. Hume, M. Jackson, A. Krause, and C. Palansuriya. Introduction to OGSA-DAI Services. *Lecture Notes in Computer Science*, 3458, 2005.

[14] P. Mazzoleni, B. Crispo, S. Sivasubramanian, and E. Bertino. Efficient Integration of Fine-grained Access Control in Large-scale Grid Services. In *Proceedings of IEEE International Conference on Services Computing*, 2005.

[15] N. Nagaratnam, P. Janson, J. Dayka, A. Nadalin, F. Siebenlist, V. Welch, I. Foster, and S. Tuecke. The Security Architecture for Open Grid Services. http://www.cs.virginia.edu/ humphrey/ogsa-sec-wg/OGSA-SecArch-v1-07192002.pdf, July 2002.

[16] OASIS. Assertions and Protocol for the OASIS Security Assertion Markup Language (SAML) v1.1. http://www.oasis-open.org/committees/download.php/3406/oasis-sstc-saml-core-1.1.pdf, 2003.

[17] OASIS. SAML 2.0 Profile of XACML. http://docs.oasis-open.org/xacml/access_control-xacml-2.0-saml/profile-spec-cd-02.pdf, 2004.

[18] OASIS. XACML 2.0 Core: eXtensible Access Control Markup Language (XACML) version 2.0. http://docs.oasis-open.org/xacml/2.0/access_control-xacml-2.0-core-spec-os.pdf, February 2005.

[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] A. Rajasekar, M. Wan, and R. Moore. MySRB and SRB-Components of a Data Grid. In *Proceedings of International Symposium on High Performance Distributed Computing (HPDC)*, 2002.

[21] V. Raman, I. Narang, C. Crone, L. Haas, S. Malaika, T. Mukai, D. Wolfson, and C. Baru. Data Access and Management Services on Grid. http://www.cs.man.ac.uk/grid-db/papers/dams.pdf, July 2002.

[22] F. Siebenlist, V. Welch, S. Tuecke, I. Foster, N. Nagaratnam, P. Janson, and A. Nadalin. OGSA Security Roadmap. http://www.globus.org/toolkit/security/ogsa/draft-ggf-ogsa-sec-roadmap-01.pdf, July 2002.

[23] B. Sotomayor. The Globus Toolkit 3 Programmer's Tutorial – Access Control with Gridmaps. http://www.cnaf.infn.it/ferrari/seminari/griglie05/lezione02/progtutorial_0.4.pdf.gz, 2003.

[24] The Globus Alliance. GridFTP. http://www.globus.org/grid_software/data/gridftp.php.

[25] V. Welch, R. Ananthakrishnan, F. Siebenlist, D. Chadwick, S. Meder, and L. Pearlman. Use of SAML for OGSI Authorization. http://www.ggf.org/Public_Comment_Docs/Documents/Oct-2005/draft-ogsi-authz-saml-sep25-05a.pdf, August 2005.