Use of SAML for OGSA Authorization

Von Welch, Frank Siebenlist, Sam Meder, Laura Pearlman

Feb 15, 2003

Abstract

This document defines the use of SAML as a message format for requesting and expressing authorization assertions in OGSA. Defining standard formats for these messages allows for pluggability of different authorization systems. OGSA authorization use cases are given and used to derive requirements for OGSA.

Document Status

This document has been submitted to the Global Grid Forum OGSA Security Working Group for consideration as a Recommendation (GWD-R).

The latest version of this document can be found at:

http://www.globus.org/ogsa/Security/

Table of Contents

1 Introduction

There are a number of authorization systems currently available for use on the Grid as well as in other areas of computing, such as Akenti [Akenti], CAS [CAS], PERMIS [PERMIS], VOMS [VOMS]. Some of these systems are normally used in a push model [RFC2904] - they act as services and issue these authorization decisions in the form of authorization assertions that are conveyed to the target resource by the requestor. Others are used in a pull model - they are normally linked with an application or service and act as a policy decision maker for that application.

On the abstract level both of these types of authorization services have similar semantics - they are given a description of the requestor, a description of an action being requested and a target resource, and they provide an authorization decision whether the action should be processed or rejected.

With the emergences of OGSA and Grid Services, it is expected that some of these systems will become OGSA authorization services as mentioned in the OGSA Security Roadmap [Roadmap]. OGSA authorization services Grid Services providing authorization functionality over an exposed Grid Service portType. A client sends a request for an authorization decision to the authorization service and in return receives an authorization assertion. A client may be the resource itself or a requestor who passes the assertion on to resource.

This specification defines the use of SAML as a message format for requesting and expressing authorization assertions from an OGSA authorization service. The SAML AuthorizationDecisionQuery element is defined as the message to request an authorization assertion and the AuthorizationDecisionStatment the method for expressing an authorization assertion. By defining standard message formats the goal is to allow these different authorization services to be pluggable to allow different authorization systems to be used interchangeably in OGSA services and clients.

This specification does not define the mechanism for conveying these messages to and from the authorization service. It is assumed that other specifications, e.g. WS-Trust, will be used to define this.

Section 2 defines the conventions used in this document. Section 3 contains OGSA Grid Service authorization use cases and requirements for authorization assertions. Section 4 contains an overview of the authorization portions of the SAML specification. Section 5 defined how the SAML elements should be used to form OGSA authorization assertions and requests. Sections 6 and 7 contain GGF copyright and intellectual property statements. Section 8 contacts author affiliation and contact information.

2 Conventions use in this Specification

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC-2119 [RFC2119].

3 Overview of OGSA Grid Services

This section is a non-normative description of the OGSA Grid Services and their authorization requirements.

3.1 Grid Authorization Use Cases

In this section we present a number of OGSA authorization use cases and requirements. Most of these sections refer to the model shown in Figure 1, with a requestor making a request of a Grid Service, which then contacts an authorization service for a decision as to whether it should process the request.

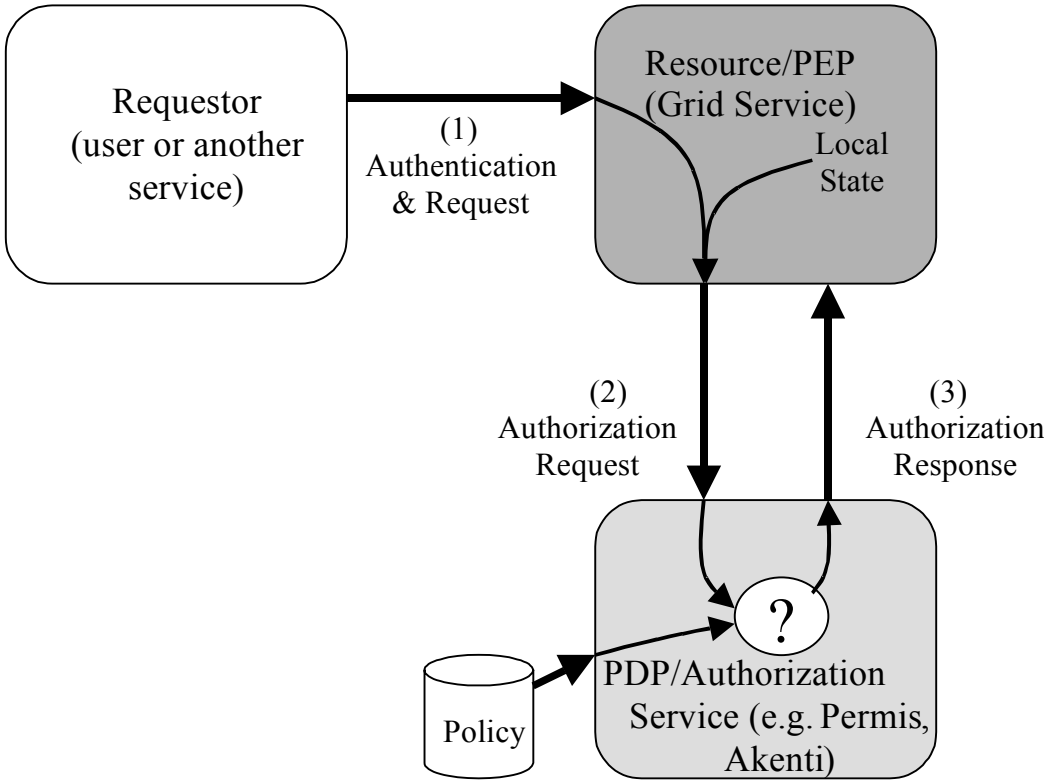

Figure 1: Authorization Pull Model. Requestor, at left, makes request of target resource. Resource queries authorization service to determine if it should process the request.

Grid Services, as defined in [OGSI], expose two basic mechanisms for interaction: *operations* and *service data*:

- *Operations* are the means by which external entities invoke the Grid Service. Operations are grouped into *portTypes*, which each portType forming, by it's collection of operations, an interface for some class of interaction (e.g., job initiation, policy management, etc.). A Grid Service may implement several different portTypes and some portTypes may themselves be composed of multiple portTypes.

- *Service Data* is the means by which a Grid Service can expose its internal state and allow it to be manipulated. Service Data is composed of *Service Data Elements* (SDEs), which each SDE holding a particular piece of data about the Grid Service's internal state.

We expect that much of the authorization policy on a Grid Service can be expressed in terms of a requestor's ability to invoke operations and access SDEs. While not ruling out other types of policy decisions, this specification defines conventions for these types of requests in particular.

3.1.1 Operation Invocation

The requestor attempts to invoke an operation on a target resource. The resource authenticates the requestor and wants to know if it should process the request or deny it. It will contact an authorization service with the information about the user, the requested operation name and an identifier for the resource itself. The authorization service will respond with a decision to the resource indicating if it should process the request.

3.1.2 Service Data Access

The requestor attempts to access a SDE on the resource, either to obtain its contents or to modify it. In this case the resource will need to contact the authorization service with the requestor's identity, a specification of the SDE, the nature of the request (i.e. read/write) and an identifier for the resource itself. The authorization service will respond with a decision to the resource indicating if it should process the request.

3.1.3 Fine-Grained policies

We expect that different polices may apply to actions at different levels of granularity. Policies may range from coarse-grained, e.g. domain or resource level, to fine-grained, e.g. not only on operations but their parameters as well.

Examples of this include:

- The requestor is attempting to invoke some operation on the resource and the policy in the authorization service is conditional not only on the specific operation but on the parameters that the requestor supplies.
- The requestor is querying for a large number of SDEs using a filter of some sort. The user may only be authorized to a subset of the set of SDEs defined by the filter.
- The requestor is attempting to set an SDE to a given value and the policy depends on the value.

One possibility for some of these scenarios would be to have the resource supply all parameters of invocation and SDE modification requests to the authorization service. However having the resource constantly supply all the operation parameters or all the SDEs in question is a large overhead since they may be quite large and may not always be required.

Instead it may be desirable to enable authorization services to be able to return conditional responses that actually contain policies to be applied by the resource. This

allows authorization services to handle situations where their policy is finer-grained than the information supplied by its client.

3.1.4 Request for All Rights on a Resource

Requestors may have long-term sessions with a resource in which they make a number of requests over a relatively short period of time. Having the resource contact the authorization service for each request in these situations is not optimal. Instead resources may want to optimize message overhead by getting a single assertion from the authorization service that contains all the rights of the requestor on the resource.

3.1.5 Supporting credentials

While a large amount of authorization on the Grid today is based solely on the requestor's identity, more sophisticated authorization systems allow the expression of attributes and other information about the requestor in policy. For example Akenti and PERMIS use attributes assertions from external sources, VOMS uses assertions of group membership and the Community Authorization Service (CAS) uses capability assertions from a VO server. We collectively refer to these assertions as *supporting credentials*.

Supporting credentials may arrive at the authorization service by a number of routes:

- Supplied by the user to the target resource, which forwards them to the authorization service;
- Gathered by the target resource and forwarded to the authorization; or
- Gathered by the authorization service.

The third case, supporting credentials gathered by the authorization service, is outside the scope of this document, but in the first two cases the supporting credentials need to accompany the request from the target resource to the authorization service.

3.1.6 Requestor contacting Authorization Service

We have mainly focused on the pull model for an authorization service as shown in Figure 1. However, some authorization services, for example the Community Authorization Service (CAS), use a push model as shown in Figure 2. This scenario is a variant of the supporting credentials scenario described in the preceding section, but is worth pointing out since the credentials issued are authorization assertions.

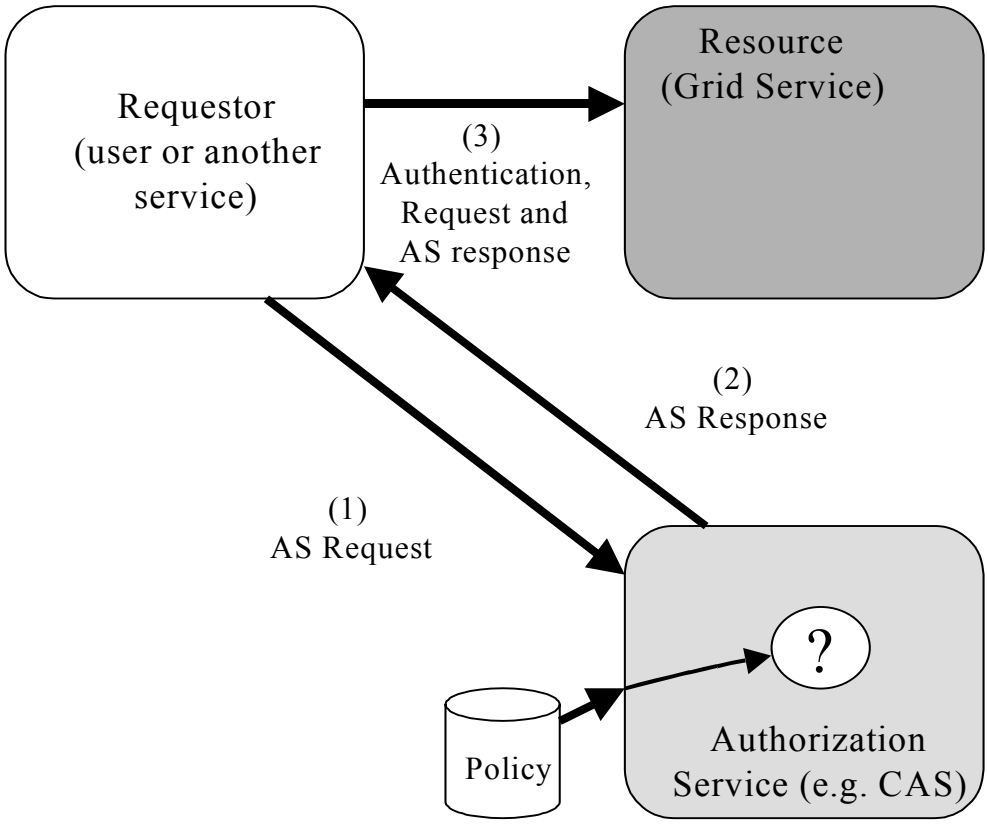

Figure 2: Authorization Service Push Model

In this scenario the requestor contacts an authorization service first and acquires a capability, which is an assertion regarding rights the requestor has on one or more resources. The requestor then presents the capability to the resource along with their request. The requestor then either evaluates the capability itself or with the help of an authorization service (possibly the same one that issued the capability).

3.1.7 Authorization of the Server by the Client

While authorization is typically thought of as a function required by services handling requests from clients, it is often also a necessary function for those clients. It is common for clients to make an authorization decision about whether a particular service is acceptable. Today client-side authorization is typically hard-coded within applications. For example in GSI and Kerberos the asserted identity of the service must contain the name of the resource on which the service is running.

However with the diversity of services on the Grid, we expect to see the use of sophisticated client-side authorization policies involving service attributes in addition identity become more common. As use scenarios become more complicated policies will also need to become more flexible to meet the needs of different users and different environments. Continuing to hard-code this policy into client will become more and more difficult and we expect to see clients making use of authorization services.

3.1.8 Application-specific policy

It is also expected that policies will be expressed in application-specific manners in addition those related to operations and service data. While it is expected that the mechanisms specified in this document are capable of supporting these types of policy statements, the means of expressing them is beyond the scope of this specification. It is expected methods of expressing application-specific policy will be established by convention or through other specifications for common classes of applications.

3.2 Grid Services Authorization Assertion Requirements

The use cases in the preceding section draw out a number of requirements that need to be supporting by OGSA authorization assertions:

- *Support for common OGSA actions*: Operation and service data access on Grid Services will be common. It is expected that a large amount of policy for OGSA can be written regarding requestors rights to perform these actions.
- *Conditional Replies*: Authorization decisions need to be able to express not only permit or deny, but conditional policies in situations where the authorization service may not have sufficient information to make a decision.
- *Supporting Credentials*: Queries to authorization services may need to supply assertions about the requestor necessary for the decision-making process.
- *Enumerated Rights*: In order to support push mode operation and sessions, assertions need to be made not only about a single right but a list of rights, possibly on more than one resource.

4 SAML Authorization Overview

The SAML specification [SAML] defines a number of elements for making assertions and queries regarding authentication, authorization decisions and attributes. In this section we give a brief non-normative overview of the elements related to authorization. Readers are encouraged to review the SAML specification for more details.

4.1 SAML Authorization Model

As shown in Figure 3, SAML defines a message exchange between a policy enforcement point (PEP) and a policy decision point (PDP) consisting of an AuthorizationDecisionQuery flowing from the PEP to the PDP, with an Assertion returned containing some number of AuthorizationDecisionStatements.

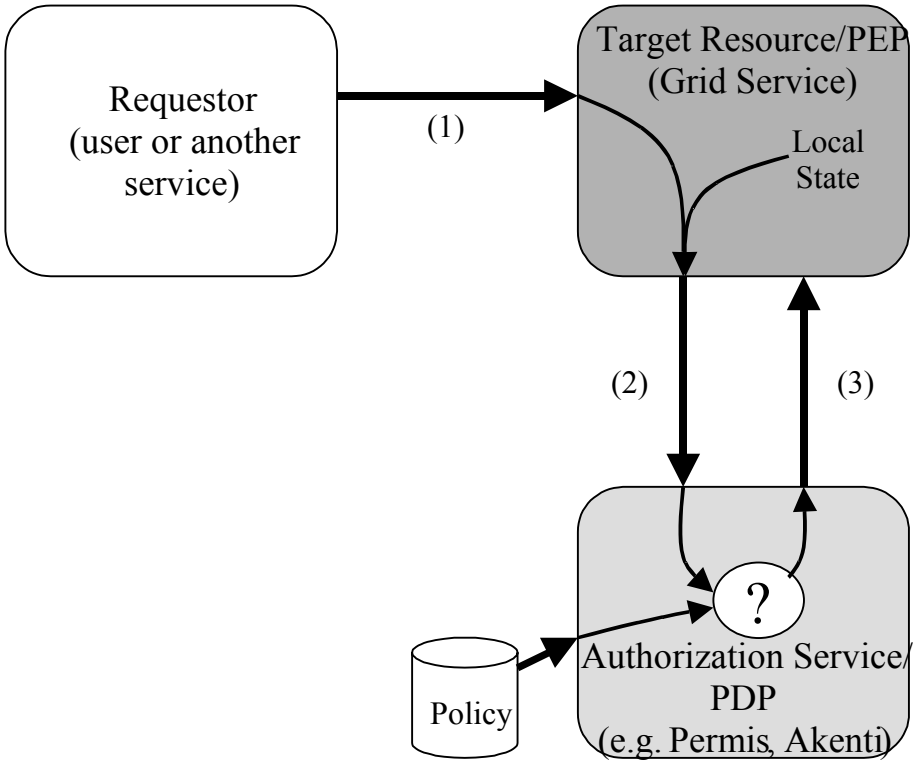

Figure 3: SAML message flow. (1) A request arrives at the target resource. (2) The Grid Service generates and sends a SAML AuthorizationDecisionQuery to an Authorization Service. (3) The service evaluates the request against policy and returns a response encoded as a SAML Assertion with one or more AuthorizationDecisionStatements.

In the following sections we describe the AuthorizationDecisionQuery, AuthorizationDecisionStatement, and Assertion elements and the elements that are use to compose these elements.

4.2 Action Element

The Action elements allows for the expression of actions that may be attempted by entities and expressed in policy. This element consists of a string and a URI defining a namespace for the action described in the string.

For example the SAML specification defines a namespace for HTTP operations that defines actions of GET, HEAD, PUT, POST.

4.3 Resource Element

The Resource element is used to identify the target on which the policy is being asserted or requested. This element is simply a URI.

4.4 Subject and NameIdentifier Elements

The Subject element contains a NameIdentifier element as well as some elements outside the scope of this document. In SAML authorization assertions, the NameIdentifer element serves to identify the requestor of the action being authorized. The NameIdentifer element contains a string to hold an identity that has two attributes:

- The NameQualifier attribute is a string expressing the security or administrative domain that defined the name (e.g. Kerberos realm, CA name).
- The Format attribute is a URI identifying the format of the name (e.g. X509 subject name).

4.5 AuthorizationDecisionStatement Element

The AuthorizationDecisionStatement element contains statements regarding authorization policy. Each of these statements contains a *Subject* element, identifying the entity whose rights are being expressed, a *Resource* element, identifying the resource the rights apply to, and any number of *Action* elements expressing the allowed operations.

4.6 Assertion Element

The Assertion element is a signed element that can contain any number of AuthorizationDecisionsStatements. It is also capable of containing statements related to authentication and attributes, but for the purposes of this document we only consider Assertions to be containing AuthorizationDecisionStatements.

4.7 Conditions Elements

Each Assertion element can also contain any number of Conditions elements. Conditions elements are currently are specified to express restrictions on the validity time of the Assertion, however they are extendable to express arbitrary conditions on the use of the restriction.

4.8 AuthorizationDecisionQuery Element

The AuthorizationDecisionQuery element allows for the request of AuthorizationDecisionStatements. It contains a Subject, Resource, and any number of Action elements that identify the policy the requestor is interesting in seeing expressed in the returned AuthorizationDecisionStatement(s).

4.9 Evidence Elements

Evidence elements allow for queries to provide information to the PDP that may be useful for it decision-making. They also allow the PDP to express what information it used to make its decision.

Each AuthorizationDecisionStatment and AuthorizationDecisionQuery element can also contain any number of Evidence elements. Each Evidence element can contain any number of Assertions elements (or references to Assertion elements) that affect the policy decision process.

5 SAML Authorization Element Usage in OGSA

This section describes how SAML Authorization elements are used to meet OSGA requirements for authorization assertions as described in Section 3.2. It first describes the use of the AuthorizationDecisionQuery element, which is used by entities to request authorization assertions from an authorization service. This is followed by a description

of the use of the Assertion element that carries the authorization assertion from the authorization service to the resource where it is evaluated.

5.1 AuthorizationDecisionsQuery Element

The SAML AuthorizationDecisionQuery element is used by an entity to request an authorization assertion from an authorization service. This element includes the following elements:

- A *Subject* element containing a *NameIdentifier* element specifying the identity of the requestor.
- A *Resource* element specifying the resource of which the request to be authorized is being made.
- One or more *Action* elements specifying the action(s) in the request to be authorized.
- Optionally an *Evidence* element containing one or more supporting credentials about the requestor.

The following subsections describe the use of and extensions to these elements for OGSA.

5.1.1 Subject Element

The Subject and contained NameIdentifer elements are unchanged. The exact use of these elements is driven by the authentication mechanism used by the requestor. The SAML specification defines how some common identity types are asserted.

The Grid Security Infrastructure (GSI) is a common Grid authentication mechanism. that uses X.509 based identities. The SAML specification defines a URI for X.509 subject names (#X509SubjectName) that SHOULD be used for GSI authenticated identities.

5.1.2 Resource Element

The Resource element is defined as a URI. If the resource being referred to is a Grid service the resource element MUST contain the Grid Service Handle (GSH) of the service as described in [OGSI].

It is also possible that this element could contain URIs referring to things other than GSHs in an OGSA context. For example, a URI could be used to refer to a group of services. However such usage is up to convention between authorization services, policy makers and resources in a particular domain and is beyond the scope of this document.

This specification also defines a wildcard resource. This has two different meanings depending on whether it is in a query (request to an AS) or a statement (response from an AS):

- In an AuthorizationDecisionQuery, it states a desire to learn all of the requestor's rights on all the resource of which the authorization service is aware. Typically such a query will be used by a requestor who will cache the results and present them to resources later in a push model of authorization.

- In an AuthorizationDecisionResponse, it states the requestor has the given privileges on all resources that accept the authorization service as authoritative. This statement may be used when the authorization service is the authority for a group of resources with identical policy.

This wildcard URI is specified as follows:

To be specified

5.1.3 Action Elements

The Action element describes the request to be authorized. The Action element is composed of a string describing the request and a URI specifying the namespace of the action.

This specification also defines the following namespaces:

- URI TBD
  This namespace is used to define an operation invocation on the specified Resource by the specified Subject. The action string should contain the namespace and name of the operation being invoked.

- URI TBD
  This namespace is used to define the reading of a ServiceDataElement. The action string should contain XXX some identifier of the SDE
- URI TBD
  This namespace is used to define the modification of a ServiceDataElement. The action string should contain XXX some identifier of the SDE.

ISSUE: Should portType be included here? If so the portType that defined the operation or the portType that the service is implementing. If it is included - how?

This specification also defines a wildcard action. This action has two different meanings depending on whether it is in a query or an assertion:

- In an AuthorizationDecisionQuery, it states a desire to learn all of the requestors rights on the specified resource. Typically this will be used by a policy enforcement point on the resource that will cache the results, set up a session with the requestor and do further policy processing without the authorization service.
- In an AuthorizationDecisionStatement, it states the requestor has all privileges on the resource. This will often be the case where the requestor is the policy authority for the resource in question.

This wildcard action is specified as follows:

URI TBD

5.1.4 Evidence Elements

The Evidence element can contain supporting credentials regarding the requestor. This specification makes no constraint on the use of this element. It is expected that specifications for different types of supporting credentials will be developed.

5.2 Assertion Element

The SAML Assertion element is used by one entity to assert the capabilities of another. While an Assertion element can contain a variety of SAML statements, for purposes of this document we consider only AuthorizationDecisionStatments.

When returned by an authorization service to an entity, the Assertion element will be enveloped in a SAML Response element as described in the SAML specification.

The Assertion element includes the following elements:

- A *Conditions* element specifying the conditions for use of the assertion.
- An *Advice* element specifying advice for use of the element.
- Any number of *AuthorizationDecisionsStatements* specifying capabilities.
- An optional *Signature* element allowing the Assertion to be verified.

The following subsections describe the use and extensions to these elements for OGSA.

5.2.1 Conditions Element

This specification places no constrains on the Conditions element. However implementations are advised to be conservative in their use of this element and only include it when they are confident it will be understood.

The Conditions element contains any number of Condition elements (note difference in plurality between element names). Condition elements serve as an abstract element for extension. It is envisioned that future specification will be able to extend the Condition element to return fine-grained policies for parameters on operation invocation and service data access.

5.2.2 Advice Element

This specification places no constrains on the Advice element. However implementations are advised to be conservative in their use of this element and only include it when they are confident it will be understood.

5.2.3 AuthorizationDecisionStatement Element

The AuthorizationDecisionStatement element contains the same elements as the AuthorizationDecisionQuery. These elements are discussed in Section 4.1.

5.2.4 Signature Element

This specification places no constrains on the Signature elements. Implementations SHOULD sign assertions when they do not have an authenticated connection to the evaluator of the assertion.

6 Copyright Notice

7 Intellectual Property Statement

The GGF takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the GGF Secretariat.

The GGF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights which may cover technology that may be required to practice this recommendation. Please address the information to the GGF Executive Director.

8 Author Info

Von Welch
Univserity of Chicago
welch@mcs.anl.gov

Frank Siebenlist
Argonne National Laboratory
franks@mcs.anl.gov

Sam Meder
University of Chicago
meder@mcs.anl.gov

Laura Pearlman
Information Sciences Institute
University of Southern California
laura@isi.edu

References

[Akenti] Thompson, M., et al., "Certificate-based Access Control for Widely Distributed Resources," in Proc. 8th Usenix Security Symposium. 1999.

[CAS] Pearlman, L., V. Welch, I. Foster, C. Kesselman, S. Tuecke, "A Community Authorization Service for Group Collaboration," Proceedings of the IEEE 3rd International Workshop on Policies for Distributed Systems and Networks, 2002.

[PERMIS] Chadwick, D.W., O.Otenko, " The PERMIS X.509 Role Based Privilege Management Infrastructure", Proceedings of 7th ACM Symoisium on Access Control Models and Technologies (SACMAT 2002).

[VOMS] "VOMS Architecture v1.1," http://grid-auth.infn.it/docs/VOMS-v1_1.pdf, Febrary 2003.

[RFC2904] Vollbrecht, J., et al, " AAA Authorization Framework," RFC 2904, August 2000.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels," BCP 14, RFC 2119, March 1997.

[OGSI] Foster, I., C. Kesselman, J. Nick, S. Tuecke, "The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration," Open Grid Service Infrastructure WG, Global Grid Forum, June 22, 2002.

[Roadmap] Siebenlist, F., et al, "OGSA Security Roadmap," OGSA Security WG, Global Grid Forum, July, 2002.