Von Welch, University of Chicago
Frank Siebenlist, Argonne National Laboratory
David Chadwick, University of Salford
Sam Meder, University of Chicago
Laura Pearlman, Information Sciences Institute

June, 2003

OGSA Authorization Requirements

Status of This Memo

This document has been submitted to the Global Grid Forum OGSA Security Working Group for consideration as informational document in that area of OGSA authorization.

The latest version of this document can be found at:

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

Copyright Notice

Abstract

This document details requirements for authorization of an Open Grid Services Infrastructure (OGSI) service as defined by the Global Grid Forum OGSI working group.

Contents

1. Introduction

The Open Grid Services Architecture (OGSA) [OGSA] casts resources in terms of Grid Services. The Open Grid Services Infrastructure (OGSI) working group of the Global Grid Forum is currently working to define a standard interface for a Grid Service [OGSI]. Based off of Web Services, Grid Services define operations for their invocation and means for publishing their internal state.

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]. On the abstract level both of these types of authorization services have similar semantics - they are given a description of the initiator (which might include the initiator's privileges), a description of an action being requested (including its argument), details about the target resource to be accessed, and any contextual information such as time of day, and they provide an authorization decision whether the action should be processed or rejected.

With the emergence 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 are 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 or a decision. A client may be the resource itself, an agent of the resource, or an initiator or a proxy for an initiator who passes the assertion on to the resource.

This document defines a number of use cases for authorization in OGSA covering the possible set of actions that may be attempted against an Grid Service, as well as how the different existing authorization services listed previously may be used. From these use cases it derives a set of requirements for authorization in OGSA.

Section 2 discusses Grid Services briefly and basic actions regarding them that need to be authorized. Section 3 discusses a number of Grid Service use cases from an authorization perspective. Section 4 contains a discussion of policy granularity and issues surrounding it. Section 5 contain the derived authorization requirements. The document then concludes with author information, copyright and intellectual property statements and a glossary.

2. Actions to Authorize in OGSA

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. Operations usually have a defined set of operands, and the requestor supplies these at the time of invocation.
- *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. While Service Data can be published in a Grid Service's WSDL description, Grid Service's also offer some operations to access Service Data, namely setServiceData, and findServiceData, plus the ability for clients to subscribe for notification of change of Service Data content.

We expect that much of the authorization policy on a Grid Service can be expressed in terms of an initiator'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. OGSA 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, described in RFC 2904 [RFC2904] as a *decision pull model*. In this model an initiator makes a request of a Grid Service, which contacts an authorization service for a decision as to whether it should process the request.

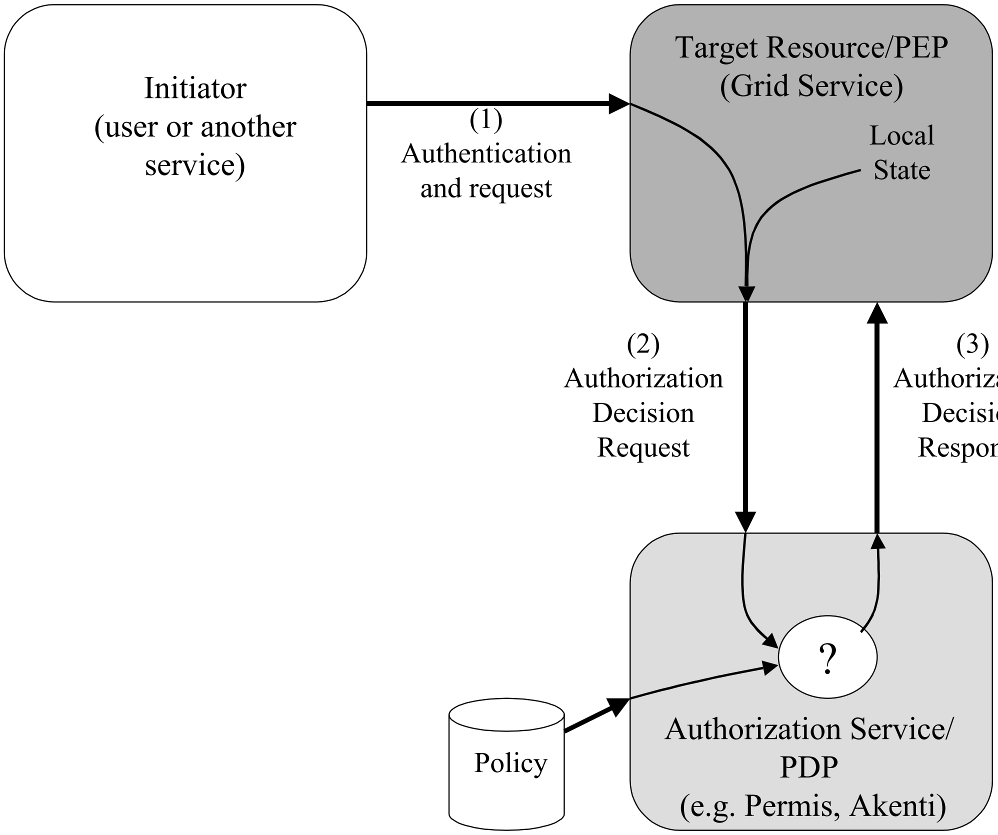

Figure 1: Authorization Decision Pull Model. Initiator, at left, makes request of target resource. Resource queries authorization service to determine if it should process the request.

3.1 Operation Invocation

The initiator attempts to invoke an operation on a target resource. The resource authenticates the initiator 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 (its name and operands), any relevant environmental parameters, 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.

3.2 Service Data Access

The initiator 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 initiator's identity, a specification of the SDE, the nature of the request (e.g. read/write, and if write, optionally the new value), any relevant contextual information, 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.

3.3 Supporting credentials

While a large amount of authorization on the Grid today is based solely on the initiator's identity, more sophisticated authorization systems allow the expression of attributes and other information about the initiator 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 or privileges*.

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 service; or
- Gathered by the authorization service.

In the first two cases the supporting credentials need to accompany the request from the target resource to the authorization service. This is the *credential push model* of RFC 3281 [RFC3281]. In the third case the authorization service may know *a priori* where to gather the supporting credentials, or the target resource may provide pointers to these locations in the authorization request. This is the *credential pull model* of RFC 3281.

3.4 Initiator contacting the Authorization Service

In addition to the decision pull model, the *decision push model,* as shown in Figure 2, is used by some authorization services, for example the Community Authorization Service (CAS). 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.

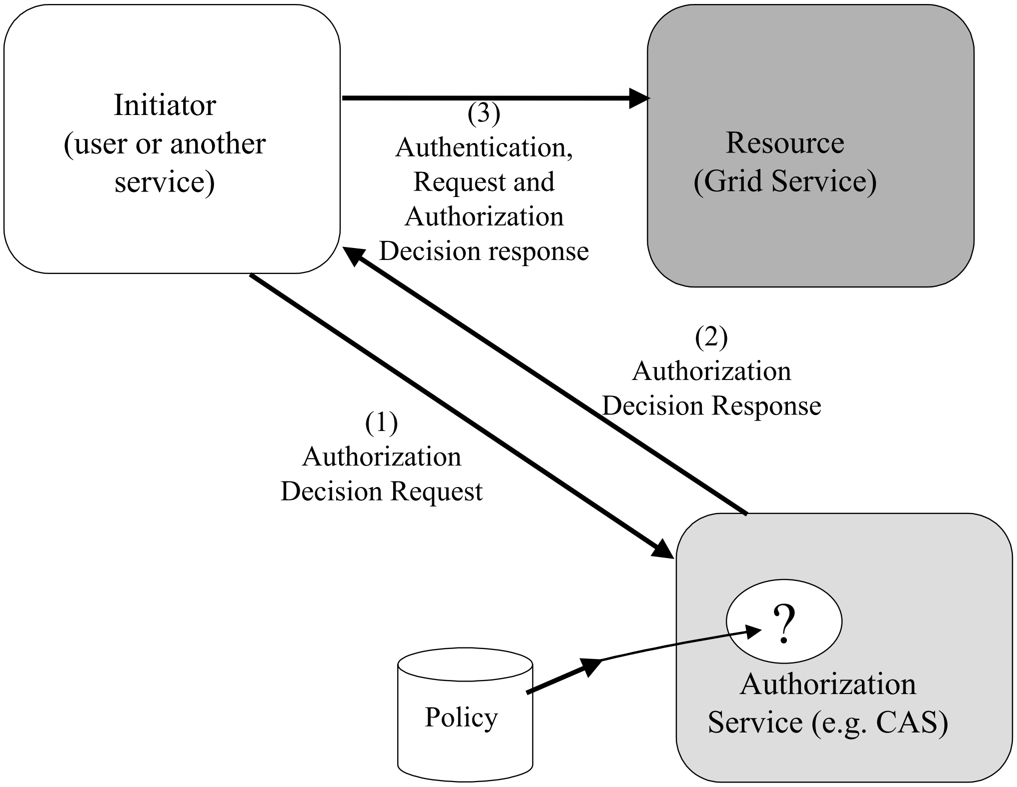

Figure 2: Authorization Service Decision Push Model

In this scenario the initiator contacts an authorization service first and acquires a capability, which is an assertion regarding rights the initiator has on one or more resources. The initiator then presents the capability to the resource along with their request. The resource then either evaluates the capability itself or could elicit the help of an authorization service as in Figure 1 (possibly the same one that issued the capability).

3.5 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. In the future clients may have privacy policies that determine what information about themselves should be released to servers.

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 to identity ones, 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 clients will become more and more difficult and we expect to see clients making use of authorization services, particularly to protect their privacy.

3.6 Session-based Authorization

Initiators may have long-term sessions with a resource in which they invoke a number of operations over a period of time. Validating the privileges that have been granted to an initiator may be a time consuming process, whereas making a decision about an operation may be relatively fast once an initiator's privileges are known to be valid. Two optimizations are provided for this scenario. Resources that want to optimize message overhead, and that are prepared to perform some decision evaluation themselves, may request a single assertion from the authorization service that contains all the rights of the initiator on the resource. Resources that wish to optimize performance of the authorization service, may request multi-step authorization. In this case, their first request to the authorization service is to validate the privileges of the initiator, and their subsequent requests are to have decisions made about each operation.

3.7 Application-specific policy

It is also expected that services will want to enforce application-specific policy with more complex logic than simple operation or Service Data access.

4. Policy Granularity

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 operands.

Examples of fine-grained authorization policy include:

- The initiator 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 initiator supplies.
- The initiator is querying for a large number of SDEs using a filter and the findServiceData operation. The user may only be authorized to a subset of the set of SDEs defined by the filter. The fine-grained authorization is held within access control lists or other mechanisms inside the SDEs themselves. The policy in the authorization service is more coarse grained and only says whether the initiator is allowed to perform certain operations or not.
- The initiator is attempting to set an SDE to a given value and the policy depends on the value.
- The initiator is attempting to invoke an operation, but the policy has time or other constraints in it, that limit when or how the initiator may invoke the operation - e.g., only between the hours of 8am and 5pm, or only up to a maximum of six times a day.

If a Grid Service is using an external authorization service it may not know the granularity of the policy and hence the level of detail it needs to supply regarding attempted actions.

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

Instead, if the resource has provided too little information, 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. If the resource is unable or unwilling to evaluate any conditions that are returned to it by the authorization service, it always has the option of making a new decision request and sending more information in the request, or of simply denying access to the initiator.

5. 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 initiators 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 initiator necessary for the decision-making process. Alternatively, the authorization service may know how to retrieve the supporting credentials itself, in which case the initiator may need to provide no information or simply a pointer to where the information can be obtained.
- *Enumerated Rights*: In order to support decision 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.
- *Session-based Authorization*: When initiators perform a series of operations on the target, the authorization decision-making should be made as efficient as possible, so that quick decisions can be made. Information that is common to each authorization request, such as the initiator’s details, should only need to be sent once to the authorization service.

6. Security Considerations

This entire document pertains to security in the form of authorization in OGSA. This document is focused on requirements and does not describe any particular mechanism and hence generate any security considerations itself.

Author Information

Von Welch
Univseríty 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

David Chadwick
Information Systems Institute
University of Salford
d.w.Chadwick@salford.ac.uk

Glossary

The following terms are abbreviations are used in this document.

ACI – Access Control Information (from ISO 10181-3). Any information used for access control purposes, including contextual information.

ADF – Access control Decision Function (from ISO 10181-3). A specialized function that makes access control decisions by applying access control policy rules to an access request, ADI (of initiators, targets, access requests, or that retained from prior decisions), and the context in which the access request is made.

ADI – Access control Decision Information (from ISO 10181-3). The portion (possibly all) of the ACI made available to the ADF in making a particular access control decision.

AEF – Access control Enforcement Function (from ISO 10181-3). A specialized function that is part of the access path between an initiator and a target on each access request and enforces the decision made by the ADF.

Client – the entity making a decision request to the ADF (it could be the target, the initiator, or a proxy acting on behalf of the initiator)

Contextual information – Information about or derived from the context in which an access request is made (e.g. time of day).

Environmental parameters – same as contextual information.

Initiator – An entity (e.g. human user or computer-based entity) that attempts to access other entities (from ISO 10181-3).

PDP – same as ADF

PEP – same as AEF

Privilege – An attribute or property assigned to an entity by an authority

Target – An entity, usually a resource, to which access may be attempted (from ISO 10181-3).

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.

Full Copyright Notice

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.

[OGSA] 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.

[OGSI] Tuecke, S., et. al., Open Grid Service Infrastructure, Version 1.0 (work in progress), April 5 2003.

[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).

[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.

[RFC3281] Farrell, S., Housley, R. “An Internet Attribute Certificate Profile for Authorization”, RFC 3281, May 2002.

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

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