Open Science Grid

Document Name	**Open Science Grid Authorization Architecture - Role Based Access Control**
Version	**Draft 1**
Date last updated	20/6/2005
Authors	Gabriele Carcassi, Rob Kennedy, Ian Fisk, Markus Lorch, Timur Peremutov, Abhishek Rana, Vijay Sekhri, John Weigand, Frank Wuerthwein
OSG document #	168

Abstract:

Change History

06/17/05		I.Fisk	Pull things together

Introduction

The Open Science Grid (OSG) Role Based Access Control (RBAC) infrastructure provides a framework for role based access to resources and services. The OSG software stack, using VDT 1.3.6 [1] and above, includes an implementation [2] to administer and apply role based access control to compute and storage resources.

The current implementation provides for the following:

- Extended X509 certificates (VOMS extended certificates) to include group membership and multiple roles.
- Site-based mapping of certificates to UNIX accounts based on extended proxy certificates.
- Application of Policy through an implementation of the standard Globus Gatekeeper, GridFTP and SRM/dCache callouts.

A Virtual Organization (VO) represents a group of resources, services and users. Contracts across the groups define agreements for using the resources and the privileges that users in the VO have to access data, storage, catalogs and other services. The VO can be administered to control who has access to what, based on what the user is doing rather than who they are. The ability to trace any use of data or service back to the individual must be maintained, but this may be indirect rather than through direct use of the users DN through all events and transactions.

Allowing multiple users to be mapped to a single account for use of a resource or service provides a much easier and more scalable administrative environment than managing each users access separately on each resource. It also contributes to (but is not sufficient for) an environment to support the sharing of data, storage and other resources. At the same time it brings with it the risks associated with people being able to negatively impact each others access.

Within OSG the RBAC infrastructure supports:

- Site administrators defining local accounts based on groups and roles of the supported VOs.
- Users defining their role for any resource access.
- VO administrators defining privileges based on role and group definitions.

The OSG components provides an integrated solution to role based access control based on

- Globus gatekeeper and GridFTP server callouts [3],
- GUMS Grid User Management System [4],
- INFN VOMS VO Membership Service [5],
- the SRM/dCache storage service and gPLAZMA module [6].
- the PRIMA modules [7].

The INFN VOMS maintains a database of members and member roles for a specific VO and can issue user attributes that certify this information for authorization purposes.

GUMS maps a Grid entity to a local username (UID) at the requested resource based on the entity's X.509 name and provided attributes. Thus, GUMS provides for the site-centralized, site-consistent allocation of local user accounts. A variety of allocation algorithms are possible including the dynamic allocation from a pool of user accounts, the mapping to role-specific shared accounts, and the mapping of individual (statically allocated) accounts.

PRIMA is a dynamically loadable authorization module that replaces the grid-map file. The PRIMA Module interfaces with Globus services through the pre-ogsa authorization callout. PRIMA supports a SAML [8] based "Obligated Authorization Decision Statement" that holds at least one obligation following the XACML obligation format.

Definition of Terms

Role-based authorization: this refers to authorization decisions based on extended proxies (roles and groups in addition to DNs). A user can get a different extended proxy from the VO VOMS server according to the role he/she is to play. For example, the user may send a request as a member of the analysis group, and at another time may send a request as a production coordinator. Role-based authorization allows the site to map the user to a different account depending on his/her role.

Account pools: Each DN is assigned a unique local account from a preallocated pool of local accounts. A particular DN always gets the same local account from then on. A site can choose to have one or more pools (for example, one pool per VO, one pool per group in a VO, one pool per role in a VO, etc...)

Authorization Process for CEs

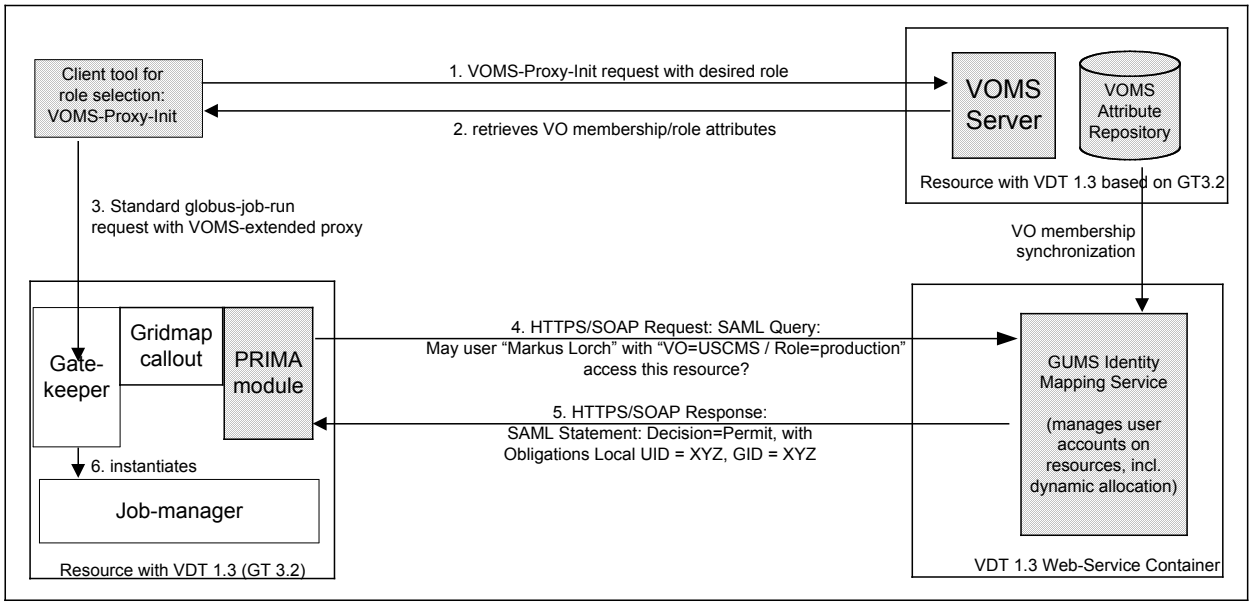


Fig. 1. Basic Privilege Project authorization architecture for compute resources

Figure 1 shows the compute resources. The numbered lines show the sequence of actions for submitting and authorizing a request for service at a compute resource.

The user requests (Step 1) the credentials that will authorize the service request to be submitted. The user can choose to either generate a generic short-lived proxy certificate using the standard "grid-proxy-init" or to use the VOMS tool "voms-proxy-init". With "voms-proxy-init" the user can customize the proxy certificate with a chosen VO and role attribute. "voms-proxy-init" will contact a VOMS server to request a trusted attribute statement from the VOMS server of the desired VO.

In response (Step 2) the VOMS server provides a signed X.509 Attribute Certificate with the requested VO-membership and desired role information if the requesting user is indeed a member of the VO and holds the desired role. The attribute information is encoded as a Fully Qualified Attribute Name (FQAN) as described in [7]. At the client side the attribute certificate is in turn embedded in the proxy credential as a certificate extension. If a user wishes to change roles a new proxy certificate must be created with the new VOMS attribute (possibly from a different VOMS server) embedded. The user can have multiple proxy certificates at any time and select the appropriate proxy certificate via an environment variable. Integration of this mechanism with credential storage solutions such as MyProxy [9] is also supported. The use of standard proxy certificates without VOMS attributes continues to be supported and provides for the backwards compatibility and ease-of-use for users with only a single (default) VO-membership and role.

Once a service request is received (Step 3) by the gatekeeper (or GridFTP server) the grid-map callout dynamically locates and invokes the PRIMA module based on information from a simple configuration file. The information provided by the grid-map callout includes the authenticated user's distinguished name (DN) as well as the security context established during authentication (which in turn holds all certificates, including the VOMS issued attribute certificate, if provided). The PRIMA module, implemented as a set of C and C++ libraries, extracts and validates the VOMS attribute certificate and parses the attribute information. To be able to verify the validity of the attribute certificates the PRIMA module must have the service certificates of all trusted VOMS servers available. However, due to the nature of GUMS (see below) verification of attribute certificates is not required if GUMS is used and can be disabled in many deployments, thus alleviating the need for trusted VOMS server certificates on the Grid resources.

The identity-mapping and authorization service is contacted (Step 4) by the PRIMA module. Using the Security Assertion Markup Language (SAML) [0], an Authorization Decision Query is formulated which contains the authenticated user DN (SAML Subject) as well as the VOMS FQAN attribute (SAML Subject Evidence). This query is sent to GUMS via a SOAP over HTTPS connection. The PRIMA – GUMS communication is based on the interface described in [11] and is also used by the OGSA based components of the Globus Toolkit for authorization decision queries. This commonality improves

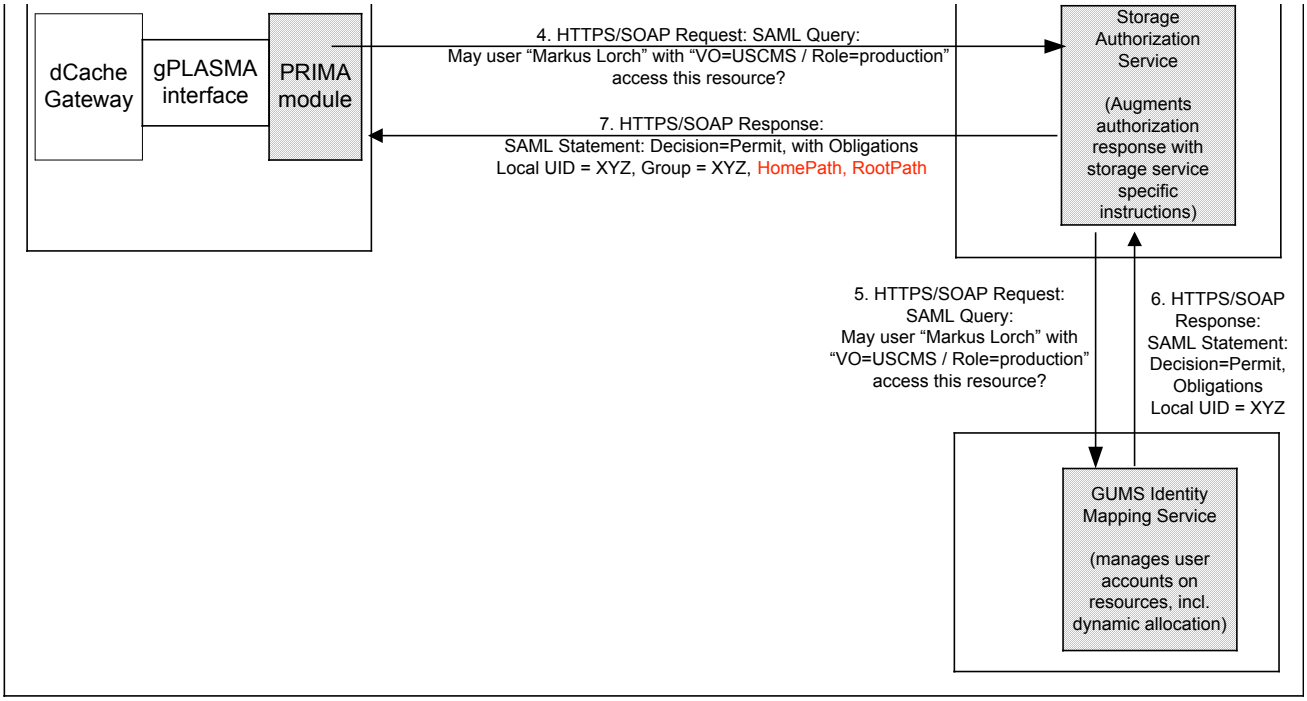
interoperability and provides for a smooth transition from the current pre-OGSA services used in the Open Science Grid to OGSA based components in the future.

GUMS, after processing the request will respond with either a SAML Authorization Decision Statement or an extended version called an Obligated Authorization Decision Statement (Step 5). The obligated statement is discussed in more detail in Section IV and enables the GUMS service to augment the basic permit/deny/indeterminate decision supported by SAML with additional decision qualifications, such as the local user account to be used for this access. The PRIMA module, upon receipt of the decision statement, will return the appropriate local user account name to the Globus gatekeeper or GridFTP server via the grid-map callout interface. The gatekeeper or GridFTP server can then continue processing the request in the same way as if a local grid-map file had been used to retrieve the local user account name.

The authorization process described above provides opportunities for backward compatibility with the existing Globus mechanisms and possible future extensions. GUMS maintains an SQL database with information about which user DNs are members of which VOs and their associated roles. This redundant storage of membership information is not necessary to reach a secure authorization decision in the described attribute push model of the privilege project infrastructure. However it enables GUMS to also operate in a legacy mode in which it creates standard grid-map files for distribution to Grid resources that do not have the PRIMA module available and thus cannot perform an online query. Furthermore it allows the PRIMA module to skip attribute verification and alleviates the need to maintain trusted attribute authority certificates (VOMS server certificates) on the Grid resources as all possible members of a VO are known to GUMS and a forged attribute cannot be used to achieve member access.

The current infrastructure could be extended to enable the setting of both UNIX uid and the primary or supplemental group. This functionality is possible in the elements of the current prototype but is not possible end-to-end. The current GridFTP call-out does not allow the group to be set and there is no robust and no intrusive way to maintain the group assignment through a batch system to a new worker node. These two limitations combined with the need to extend existing components to support group assignments lead the OSG development team to assign only UNIX IDs in this version. The requirement to be able to assign group ID could be revisited in the future.

A closely related authorization component that may be combined with this infrastructure is the Site Authorization Service (SAZ) [5]. There exists a SAZ client module that is similar to the PRIMA module in that is invoked by the Grid service and queries a service, the SAZ service, for an authorization decision. SAZ enforces the site-specific access control rules/policies such as specifying prohibited users, checking for revoked certificates, and validating the user's certificate path. The SAZ service developed at FNAL is currently relying on a proprietary protocol for communication. It is planned that future versions will implement the same SOAP/SAML protocol and authorization interface used in the PRIMA-GUMS communication.



The performance of the integrated set of services has preliminary been measured. We have measured no additional overhead in Gatekeeper due to the PRIMA to GUMS interface. Tests showed delays on the order of 0.5 seconds/request on the OSG Integration Grid. The site-centralized GUMS server can sustain more than 50 requests/second. Load-balancing and redundancy for GUMS servers is possible through the underlying Tomcat 5 web service container.

Authorization Process for Storage

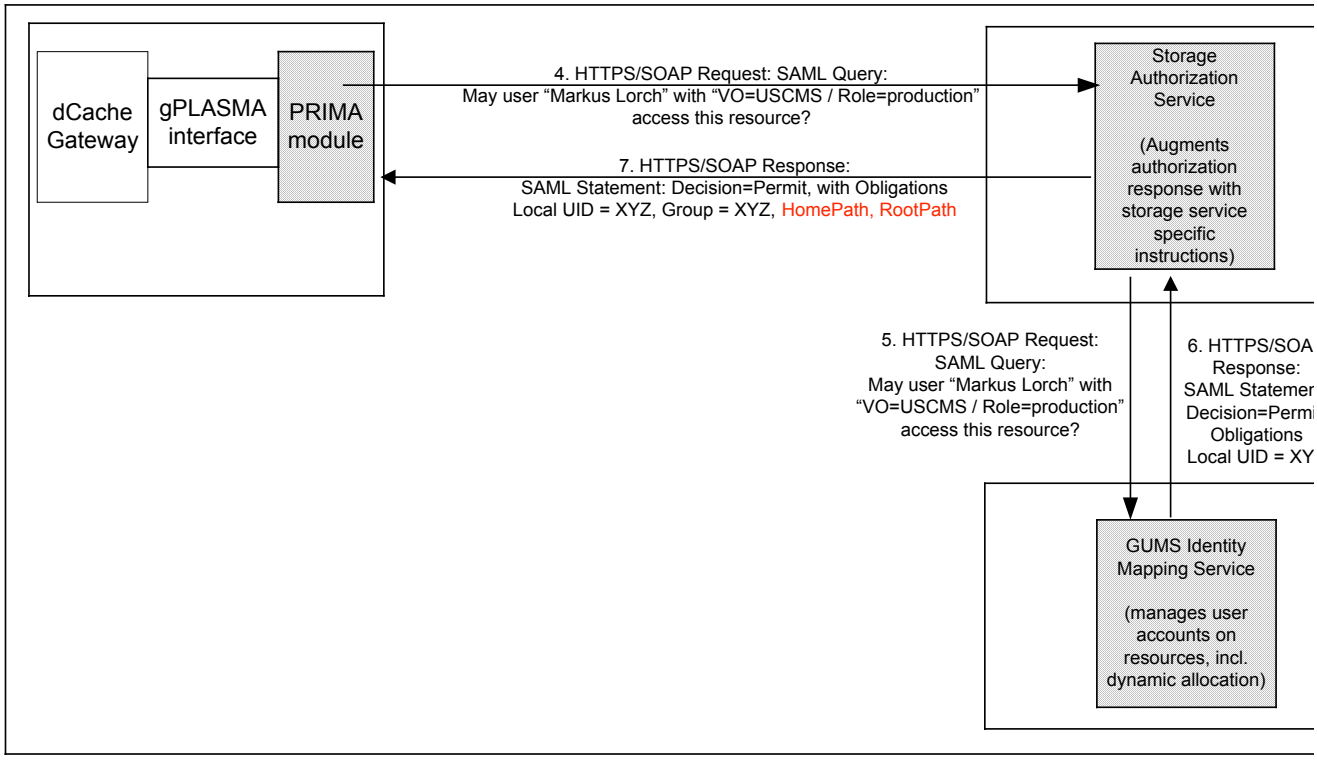


Fig. 2 Storage Authorization Architecture and Sequence

In Figure 2, Steps 1 to 3, the selection of VO and role through the creation of a proxy certificate with VOMS-issued attributes and the user initiated service request to the storage resource are identical with those in Fig. 1 and not pictured in Fig. 2. There is no semantic difference to the user between a compute and a storage service access.

When a storage service access request is received by the dCache storage system the gPLASMA interface calls the PRIMA module with user the credentials provided to the storage system during authentication. The PRIMA module extracts and optionally verifies presented attributes and formulates a SAML Authorization Decision Query. This query is sent to the Storage Authorization Service (Step 4), which exposes the same SAML authorization port type that was added to GUMS. The Storage Authorization Service, upon receipt of such a query, passes on this query to the site's GUMS server (Step 5). If the GUMS server returns a positive response (with an obligation that specifies a local user name, Step 6) then the Storage Authorization Service queries its local policy file for additional authorization decision qualifications that must be provided to the storage resource with the decision. If GUMS returns a negative response then no further processing on by the Storage Authorization Service is required. Finally the augmented SAML response, including additional obligations for the storage resource, is provided back to the PRIMA module of the dCache service (Step 7).

Using the privilege project components in OSG

The following steps happen in the OSG implementation of the CE:

- Get a VOMS extended proxy with the attributes you wish to use
- Submit a request to run a job.
- Gatekeeper PRIMA modules parse the proxy attributes and contact the GUMS service for uid mapping
- GUMS service checks its database and configuration to map a local user id for that DN + attributes and returns the answer to the gatekeeper.
- The gatekeeper performs the job as the assigned user.

An example GUMS configuration:

```
<groupMappings>
    <groupMapping name='ivdgl' accountingVo='ivdgl' accountingDesc='iVDgL'>
      <userGroup className='gov.bnl.gums.VOMSGroup'
url='https://grid03.uits.indiana.edu:8443/edg-voms-admin/vos/services/VOMSAdmin'
            persistenceFactory='mysql' name='ivdgl' voGroup="/vos/ivdgl"
            sslCertfile='/etc/grid-security/hostcert.pem'
```

```
        sslKey='/etc/grid-security/hostkey.pem' ignoreFQAN="true"/>

    <accountMapping className='gov.bnl.gums.GroupAccountMapper'
            groupName='ivdgl' />
    </groupMapping>
```

References

[1] Virtual Data Toolkit http://www.cs.wisc.edu/vdt//index.html

[2] VO Privilege Joint Project
http://computing.fnal.gov/docs/products/voprivilege/index.html

[3] I. Foster and C. Kesselman, "Globus: A Toolkit-Based Grid Architecture" in "The Grid, Blueprint for a Future Computing Infrastructure", I. Foster, and C. Kesselman, Editors, Morgan Kaufmann, San Francisco, 1999, pp. 259-27 http://www.globus.org/

[4] G, Carcassi "The Grid User Management System". http://grid.racf.bnl.gov/GUMS

[5] Alfieri et al. "VOMS: an Authorization System for Virtual Organizations" 1st European Across Grids Conference, Santiago de Compostela, Feb. 13-14, 2003

[6] A. Rana et al "gPLASMA" http://osg-docdb.opensciencegrid.org/0001/000108/001/plugable-authorization-module-info-fermilab.doc

[7] M. Lorch, D. Kafura "The PRIMA Grid Authorization System", Int. Journal of Grid Computing (2004) 2: pp. 279–298.

[8] P. Hallam-Baker, E. Maler, et al, "Assertions and Protocol for the OASIS Security Assertion Markup Language (SAML), Oasis Standard, November 5th, 2002

[9]J. Novotny, S. Tuecke, and V. Welch, "An Online Credential Repository for the Grid: MyProxy", Proceedings of the Tenth International Symposium on High Performance Distributed Computing (HPDC-10), IEEE Press, 2001.