Use of XACML Obligations in SAML Authorization Decision Statments

This document discusses how the Authorizaiton components developed as part of the Privilege Project for OpenScienceGrid extend SAML Authorization Decision Statements with obligation constructs. Obligations are used to provision additional decision qualifications, such as what local user account to use for the requested access, from a policy decision point to the policy enforcement point. The system utilizes the XACML obligation format.

Use of XACML Obligations in SAML Authorization Decision Statments

U.S. CMS and U.S. ATLAS Privilege Project
http://computing.fnal.gov/docs/products/voprivilege/ (
http://computing.fnal.gov/docs/products/voprivilege/)

Date: 2005-02-21, Last Updated 2005-02-23
Author: Markus Lorch, Email: mlorch at vt dot edu

The latest version of this document is available from:
https://plone3.fnal.gov/opensciencegrid/techgroups/tg-policy/vo-privilege (
https://plone3.fnal.gov/opensciencegrid/techgroups/tg-policy/vo-privilege)

Content:

This document discusses how the Authorizaiton components developed as part of the Privilege Project for OpenScienceGrid extend SAML Authorization Decision Statements with obligation constructs. Obligations are used to provision additional decision qualifications, such as what local user account to use for the requested access, from a policy decision point to the policy enforcement point. The system utilizes the XACML obligation format.

1. Introduction

The Privilege Project authorization infrastructure supplements the security infrastructure of the Globus toolkit by

1. replacing the local grid-mapfile mechanism on a grid resource with a site-centralized identity mapping and authorization server
2. allowing a large variety of flexible, VO-driven and dynamic user account mapping solutions
3. empowering users to select specific VOs, VO-subgroup, and roles individually for each service

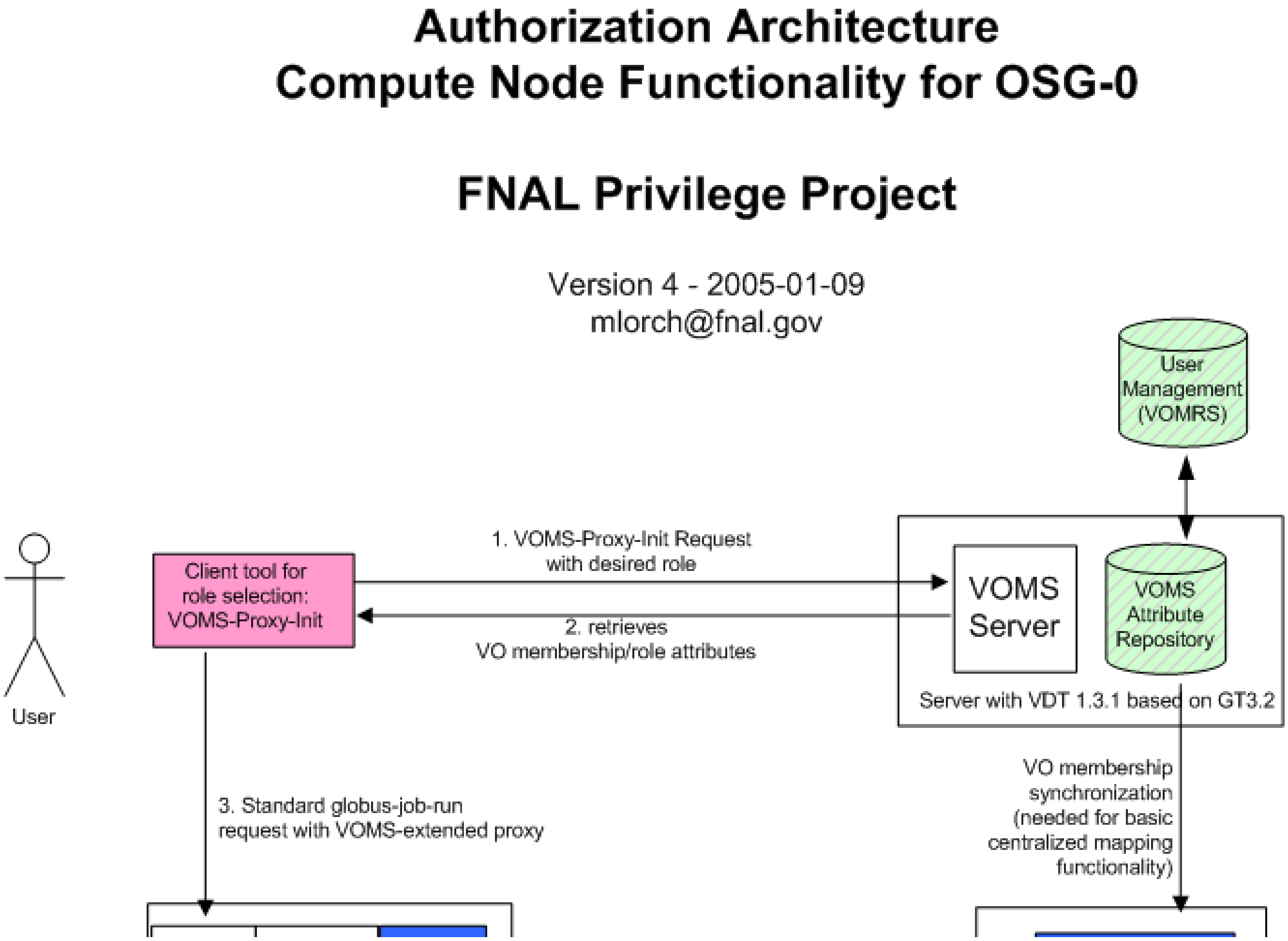
request. The user can choose from the set of entitlements bound to the user's identity by trusted attribute authorities.

4. allowing multiple authorization policies to be combined and taken into account when a service request is evaluated

The Privilege Project authoriztion infrastructure utilizes four principal components to achieve this extension of the existing security infrastructure provided by the globus toolkit.

1. The PRIMA module, a dynamically loadable authorization module used by grid services such as the Globus gatekeeper and the GridFTP service to query the identity mapping service (GUMS, see below) by use of the SAML protocol and message format
2. The GUMS Grid User Management Service, an identity mapping service that exposes the Grid service authorization port type, accepts SAML AuthorizationDecisionQueries and responds with (extended) SAML AuthorizationDecisionStatements.
3. The PRIMA authorization service, an XACML-based policy decision point that also exposes the Grid service authorizaiton port type
4. The VOMS Virtual Organization Management Service as a trusted attribute repository

The motivations for using the privilege components and an overview over the privilege authorization infrastructure are presented at http://computing.fnal.gov/docs/products/voprivilege/documents/motivation-2005-01-14.pdf (http://computing.fnal.gov/docs/products/voprivilege/documents/motivation-2005-01-14.pdf) . Firgure 1 below provides an architectural overview that shows the privilege components needed to replace the legacy grid-map functionality. A transition path to the privilege components for the grid-map functionality is available from http://computing.fnal.gov/docs/products/voprivilege/documents/transition-to-privilege.html (http://computing.fnal.gov/docs/products/voprivilege/documents/transition-to-privilege.html) .



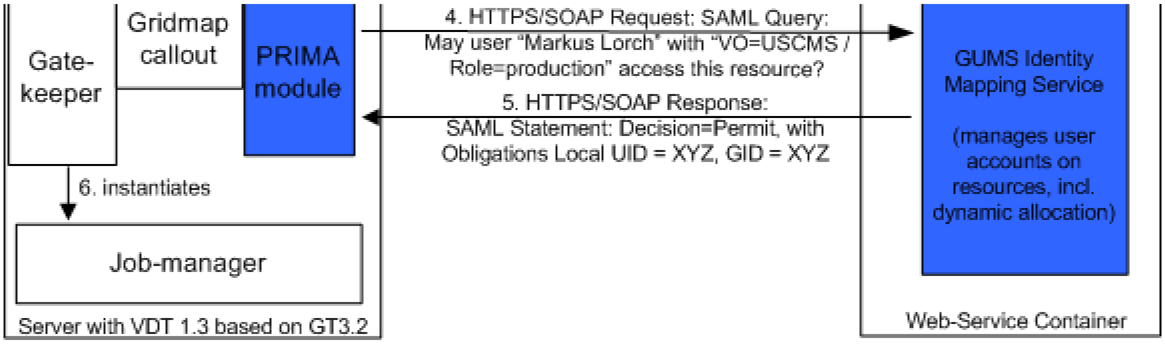

Gate-keeper
Gridmap callout
PRIMA module
4. HTTPS/SOAP Request: SAML Query: May user "Markus Lorch" with "VO=USCMS / Role=production" access this resource?
5. HTTPS/SOAP Response: SAML Statement: Decision=Permit, with Obligations Local UID = XYZ, GID = XYZ
GUMS Identity Mapping Service
(manages user accounts on resources, incl. dynamic allocation)
6. instantiates
Job-manager
Server with VDT 1.3 based on GT3.2
Web-Service Container

Figure 1 - Privilege Architecture Overview - Grid User Mapping Usecase

2. The Use of Obligations

Obligations are a set of instructions paired with an authorization decision statement or response. These instructions may be targetted at the Policy Enforcement Point and may be used to describe how a requested service, if allowed, should be confined and monitored during its execution. In this context obligations may also be used to convey additional instructions for how to treat a service request that is not authoirzed.

Obligations in authorization decision statements can be used to address one of the more subtle issues frequently encountered when fine-grained authorization decisions are made: a mismatch in the level of detail between the authorization request and the applicable policies. Policy decision points are application independent and are unable to understand or extrapolate the implications of a broad resource request. An example is the request to instantiate a user-provided service. The applicable policies and provided privileges are likely to specify in detail what a user provided service is allowed to do. But a simple permit/deny decision from the decision point cannot convey this level of detail. A positive authorization decision response thus needs to be augmented with additional decision qualifications that instruct the enforcement point how exactly the requested action should be permitted and if additional constraints should be applied. For example, the list of fine-grained access rights that specifies to what extent the user provided service is allowed to access other services and resources of the hosting environment can be provided this way. If the PEP cannot fulfill the obligations then it should not allow the access to proceed. The enforcement point, upon receiving a positive response from the decision point, instantiates a custom execution environment configured with the access rights as specified in the obligations, and starts and monitors the execution of the requested service in this environment. Obligations can also be used to pass state information e.g., about previous requests, back to the policy enforcement point or a resource monitor for accounting.

For example, if the concept of authorizaiton decision queries and authorizaiton decision statements is applied to the grid identity mapping function a conversation between a grid service and an identity mapping service may go as follows:

1. 1. Grid service queries identity mapping service:
   May subject "CN=Markus Lorch" access service "jobmanager" on Grid resource "CN=host.domain.tld" ?
2. The identity mapping service may reply with:
   Permit, subject "CN=Markus Lorch" is allowed to access service "jobmanager" on Grid resource "CN=host.domain.tld". The jobmanager service must be executed in the local user account "markus".

The while the first part of the response can be implemented using a standard SAML authorizaiton decision statement, the second part (The jobmanager service must be executed in the local user account "markus") requries the addition of obligations to the decision statement.

If an enforcement point does not understand an obligation presented to it via an authorization decision it may not provide the requested service even if the overall decision was permitt. If the obligation is applied to a deny response (currently not used in the privilege project) the enforcement point has to make a best effort to follow the instructions provided in the obligations.

3. SAML Extensions

In the Privilege Project we have opted to extend the SAML Authorization Decision Statement to create an "Obligated Authorization Decision Statement" that holds at least one obligation following the XACML Obligation Format. If the authorizaiton decision response from privilege project identity mapping service.

The XML schema for our ObligatedAuthorizationDecisionStatement is below:

```
<element name="ObligatedAuthorizationDecisionStatement"
                       type="osg-saml:ObligatedAuthorizationDecisionStatementType"/>
<complexType name="ObligatedAuthorizationDecisionStatementType">
    <complexContent>
      <extension base="saml:AuthorizationDecisionStatementType">
        <sequence>
          <element name="XACMLObligation" type="osg-saml:XACMLObligationType" maxOccurs="unbounded"/>
        </sequence>
      </extension>
    </complexContent>
</complexType>
```

The XML schema for the XACML Obligation element is:

```
<element name="XACMLObligation" type="osg-saml:XACMLObligationType"/>
<complexType name="XACMLObligationType">
  <sequence>
    <element name="AttributeAssignment" type="osg-saml:AttributeAssignmentType" />
  </sequence>
  <attribute name="FullfillOn" type="string"/>
  <attribute name="ObligationId" type="string"/>
</complexType>
```

The XML schema for the AttributeAssignment element is:

```
<element name="AttributeAssignment" type="osg-saml:AttributeAssignmentType"/>
<complexType name="AttributeAssignmentType">
  <simpleContent>
    <extension base="string">
      <attribute name="AttributeId" type="string"/>
      <attribute name="Datatype" type="string"/>
    </extension>
  </simpleContent>
</complexType>
```

The definition allows for a set of XACMLObligation elements to be added to a SAML AuthorizationDecisionStatement. Ideally this would be implemented by using the SubjectStatement extension point in SAML.

Each XACMLObligation element specifies if it is to be fullfilled on a permit or deny response. Fullfillment of an XACMLObligation translates to the application of the attribute assignment that the obligation statements conveys. A set of attribute assignments can be provided with a single obligation.

The XACMLObligation format does not describe the semantics of the attributes that are assigned and is completely independent of the application.

The use of the XACML Obligation format further more allows the seamless integration with XACML policies and policy decision functions. An XACML Obligation can simply be embedded in the applicable XACML Policy and will automatically be included in the authorization decision statement that is conveyed to the enforcement point. This allows us to write enforcement service specific policies that can provision enforcement service specific authorization information (such as the rootPath and priority obligations for storage elements explained below) without needing to have the PDP implementation be service specific.

Furthermore the use of XACML Obligations will allow us to transition seamlessly to the new XACML over SAML authorization message format in the future. (See http://www.oasis-open.org/committees/download.php/10525/XACML-2.0-SAML-PROFILE-CD-02.zip (http://www.oasis-open.org/committees/download.php/10525/XACML-2.0-SAML-PROFILE-CD-02.zip))

4. Example Obligations and Attriute Assignments

For the OpenScienceGrid a set of obligations and attribute assignments have been defined to facilitate the identity mapping scenario as well as the provisioning of storage systems with additional parameters for constrained service execution.

The following obligations all hold a single attribute assignment with data-type "http://www.w3.org/2001/XMLSchema#string":

- the **userid obligation** "opensciencegrid:authorization:UserIdObligation" holds a single string attribute "opensciencegrid:authorization:attribute:UserId" with the local user name to be used when servicing this service request
- the **groupid obligation** "opensciencegrid:authorization:GroupIdObligation" holds a single string attribute "opensciencegrid:authorization:attribute:GroupId" with the local primary group name to be used when servicing this service request
- the **supplemental group id obligation** "opensciencegrid:authorization:SupGroupIdsObligation" holds a single string attribute "opensciencegrid:authorization:attribute:SupGroupIds" with a space delimited list of local group names to be used when servicing this service request
- the **root path obligation** "opensciencegrid:authorization:RootPathIdObligation" holds a single string attribute opensciencegrid:authorization:attribute:RootPathIdAttribute" with the root path to be chrooted to before commencing with servicing the request
- the **relative home path obligation** "opensciencegrid:authorization:RelHomePathIdObligation" holds a single string attribute "opensciencegrid:authorization:attribute:RelHomePath" with the home path relative to the root path

5. Implementation Status

Within the privilege project the openSAML implementation of the SAML message format and procol has been extended (both, in java as well as in C) to accomodate the ObligatedAuthorizationDecisionStatement as well as the XACMLObligation and AttributeAssignment element. This implementation is being used in the PRIMA authorization module and the privilege project authorization service stub, which is common to the GUMS Identity Mapping Service and the PRIMA Authoirzation Service.

The PRIMA authorization module and the GUMS Identity Mapping Service have, at the time of this writing, been successfully deployed on a number of OpenScienceGrid sites used for integration testing.

These components will be available in the first production release of the OpenScienceGrid.