A proposal for supporting VOMS-based ACLs in current SRMv2.2 Storage Service implementations

Flavia Donno

CERN

27 June 2007

DRAFT

This document is the result of the discussions within the WLCG Grid Storage System Deployment (GSSD) working group.

Many people have contributed to this work: Jean-Philippe Baud (CERN), Patrick Fuhrmann (DESY), Maarten Litmaath (CERN), Luca Magnoni, Riccardo Zappi (INFN-CNAF), Timur Perelmutov (FNAL), Shaun de Witt (RAL), LHC experiments representatives, WLCG site administrators.

The document has been started following the example of the VOMS-based ACL support implemented in DPM. The idea was to guarantee homogeneous functionality among all Storage Service implementations.

The experiments recognize that the functionalities offered by an implementation following the recommendations in this document are sufficient to guarantee WLCG operations.

VOMS-based ACLs in Storage Services

VOMS GROUPS, ROLES, AND ACCESS CONTROL LISTS

Every user is assigned a VOMS proxy when using the WLCG Grid. In the context of this document a simple grid proxy is equivalent to a VOMS proxy with the (single) VO being the only extra attribute (determined from a grid-mapfile). A proxy is first characterized by the Subject Distinguished Name (DN), and can have extensions that define the privileges of the user holding that proxy at a given moment. Example DN:

(1) /O=Grid/OU=cern/CN=bla/E-mail=Flavia.Donno@cern.ch

To define the privileges of a user at a given moment, groups, sub-groups, and roles can be defined. In particular a user can belong to multiple groups and sub-groups and have a number of roles at a given time (the current version of VOMS supports just a single role at most, but that limitation is due to a bug that will be fixed). Example of groups and roles:

(2) /dteam/Role=lcgadmin
/dteam/cern/Role=NULL

Once the user presents his proxy to a Grid service, this typically maps the groups, subgroups and roles to one or more GIDs (Group IDs) and the user DN to one specific UID (User ID). The privileges of the user on the resources managed by the contacted Grid service are therefore defined by the privileges of the n-tuple (UID, GIDs) in the system. LCMAPS mapping examples:

(3) “/dteam/Role=lcgadmin” .dtmsgm, “/dteam” .dteam

An Access Control List (ACL) is a list of entries defining the authorization on a given resource. ACLs can be positive, i.e. defining who is authorized to perform a certain set of operations or access a given resource, or negative, negating permission to the service.

(4) Example of ACL:

```
# file: /grid/dteam
# owner: root
# group: dteam
user::rwx
group::rwx #effective:rwx
group:dteam/Role=lcgadmin:rwx #effective:rwx
group:dteam/Role=production:rwx #effective:rwx
```

```
mask::rwx
other::r-x
default:user::rwx
default:group::rwx
default:group:dteam/Role=lcgadmin:rwx
default:group:dteam/Role=production:rwx
default:mask::rwx
default:other::r-x
```

II REQUIREMENTS

Here we list a set of requirements gathered from various discussions with the WLCG experiment representatives and site administrators. These are considered to be minimal requirements, sufficient to run the WLCG Storage Services from now till the end of 2010. After this initial period, running experience may lead to the review and reconsideration of this initial list.

1) ACLs can be set on Storage Element Files, Directories, Storage Areas and Storage Components.

> For the definition of a Storage Area and a Storage Component, please refer to the document “Storage Element Model for SRM 2.2 and GLUE schema description”, published at the following URL:
>
> http://glueschema.forge.cnaf.infn.it/Spec/V13
>
> ACLs on Files and Directories can be set up by a VO manager or a site administrator. They are meant to establish who in a VO can perform operations on a specific file or directory.
>
> ACLs on a Storage Area can be set by a VO manager, after reserving the Storage Area using a specific space token description. ACLs on a Storage Area define who can use that Storage Area.
>
> ACLs on a Storage Component are meant to address the system to use specific resources when a request for space reservation has arrived and the system needs to locate resources to satisfy the specific request finding space for a VO. Such ACLs can be set by a site administrator only.

2) Default ACLs on directories are inherited by files and sub-directories

3) Authorization on the name space is done using groups, sub-groups and roles. Storage component selection should be done using only primary groups.

4) ACLs can be positive or negative

> Positive ACLs are meant to grant access while negative ACLs specify who should be negated the authorization to the resources the ACLs apply to.

5) ACLs could be expressed by regular expressions in terms of FQAN if the wildcard semantics can be agreed on.

6) It should be possible for a site administrator or for a VO manager to setup “fall-back” Storage Areas. Such an option should be configurable.

> Whenever no Storage Area ACL matches the proxy of a requester, it should be possible for the system to find a “fall-back” Storage Area to accommodate the request if possible. For instance, if at a site only a Storage Area for normal ATLAS users has been set up, an ATLAS production manager should be able to use that area, if the site so desires and the VO would not be adversely affected (it may not want the user areas to get filled up accidentally with production data).

7) Management tools should be available to manipulate ACLs directly and change the internal resolution of proxies.

> This is necessary to handle for instance the case of Certification Authorities changing the DN format when certificates are renewed.
>
> Another need is to configure the Storage Element to handle correctly VO groups/sub-groups/roles (defining the privileges for a given set of groups/sub-groups/roles).

III DETAILS OF THE STORAGE SERVICE IMPLEMENTATIONS

In this section we outline the details of the VOMS-based ACL implementations available already in some of the Storage Services. In particular, we describe some of the features available in DPM, dCache and StoRM.

DPM

VOMS-based ACLs in DPM are based on virtual UIDs and GIDs which are created on the fly in the DPM database the first time the system receives a request for a new DN (in a supported VO) or a new VOMS attribute. In particular, UIDs are associated to DNs and GIDs are associated to VOMS groups/roles. Therefore, a user at a given time is associated to one virtual UID and several virtual GIDs that define the user privileges.

In DPM, VOMS ACLs can be set on directories and files. VOMS ACLs can also be set on pools (set of file systems/storage components). These latter ACLs are used by the system when allocating space to satisfy a reservation request.

At the moment DPM offers the ability to set ACLs on Storage Components and Storage Areas. A space token is assigned to one user or to one group and no further ACLs are possible on spaces. Furthermore, requirements 4 (negative ACLs),5 (regexps) and 6 (fallback) are not yet satisfied (4 and 5 would need significant effort to implement).

dCache

VOMS-based ACLs in dCache are based on virtual n-tuples of UIDs and GIDs. Each VOMS group or role can be associated to a pair of (UID, GID). Therefore, a proxy with a set of groups and roles is mapped to a set of (UID, GID) pairs which are created in advance. These n-tuples define the user privileges on a resource.

At the moment VOMS-based ACLs are not yet available in dCache. However, the Storage Element is internally enabled to handle ACLs.

VOMS ACLs can be set on directories and files. VOMS ACLs can also be set on link groups, which are the equivalent to Storage Areas and/or Components.

StoRM

StoRM allows for Just-in-Time (JIT) and for Ahead-of-time (AOT) ACLs. The JIT ACLs are automatically and dynamically created or deleted by the system when a request arrives or finishes respectively, consulting an external authorization service. AOT ACLs are created in advance or the first time a request for a given file/directory arrives.

In StoRM a Storage Area corresponds to a directory. Therefore setting ACLs on a directory implies the same ACLs are set on the corresponding Storage Area.

It is not possible to set ACLs on Storage Components in StoRM.

IV INTERFACES

It has been noted that SRM v2.2 does not offer a completely adequate interface to set VOMS-based ACLs. The concepts of files and directories are well established within SRM v2.2 and the functions srmSet/GetPermissions allow for UNIX-type privileges such as “rwx” to be set for the owner, explicitly named VOMS groups/subgroups/roles, and others. However, the concepts of Storage Areas and Storage Components, even if introduced, are not explicitly exposed by the SRM interface. Therefore, a possible interface to operate on such entities is totally missing.

It has been decided by the GSSD working group to describe a possible interface for VOMS-based ACLs, once the functionalities have been agreed. Such an interface can then be proposed to be taken into account when defining SRM v3 or v2.3, or can be defined as another standard interface. A document will follow with a first proposal of the interface.

Example:

```
dpns-mkdir /dpm/cern.ch/home/dteam/jpb

dpns-setacl -m d:u::7,d:g::7,d:o:5 /dpm/cern.ch/home/dteam/jpb

dpns-getacl /dpm/cern.ch/home/dteam/jpb

# file: /dpm/cern.ch/home/dteam/jpb

# owner: /C=CH/O=CERN/OU=GRID/CN=Jean-Philippe Baud 7183

# group: dteam

user::rwx

group::r-x              #effective:r-x

other::r-x

default:user::rwx

default:group::rwx

default:other::r-x
```