Table of Contents

Job Prioritization resources

About VOMS FQANs

Some facts good to be aware of:

- The current format of the FQAN: `/dteam/certification/Role=NULL`
- The first 'tag' of the group attribute is always the VO name. In the above example it is `dteam`.
- A VOMS proxy can arrive with several FQANs , out of this the first one (the primary) FQAN could have special meaning.
- It is not possible to have FQAN with no group membership, one always belongs to the 'base' group i.e. to the VO.
- One always automatically become the member of all his/her group when creating a VOMS proxy.
- It is not possible to ask for the exclusion of one or several group from the VOMS proxy.
- One has to explicitly request the `Role=` attribute, requests without this will results `Role=NULL`
- In one FQAN only one Role is allowed, however secondary FQANs could contain additional Roles [available only in VOMS server 1.7.0 and later].
- As default, Roles are not inherited to subgroup. If one has `/Role=production` in let's say the `/dteam` group and at the same time (s)he is member also of the `/dteam/certification` group, (s)he cannot ask for a proxy having FQAN `/dteam/certification/Role=production` . The `/Role=production` has to be explicitly granted to her/him inside the subgroup , as well.

Interpretation of FQANs in LCMAPS

Versions

The glite-CE and lcg-CE are using different versions of LCMAPS.

- The glite-CE (and all othe glite services) uses `glite-security-lcmaps-1.3.6-1`,
- while lcg-CE uses `edg-lcmaps_gcc3_2_2-0.0.30-1_sl3`.

Matching mechanism 1

The matching procedure contains the following steps:

- LCMAPS treats the first (primary) FQAN specially: it *must* match a rule in the gridmapfile and it determines the UID and the primary GID.
- Subsequently all secondary FQANs are tried against the groupmapfile, to pick up as many secondary GIDs as possible.
- For each secondary FQAN the scan is stopped after the first match.
- It is not an error for some secondary FQAN not to match at all.

So, as far as job priorities are concerned, only the primary FQAN matters.

Interpretation of wildcard characters 1

In the glite version

In order to match the this FQAN

```
/atlas/Role=NULL/Capability=NULL
```

one can use wildcard characters in the following combinations. The followings will all match against the above 'target' FQAN. (Please note, that we just took the `/Capability=NULL` as an example, this attribute is deprecated!)

```
/atlas/*
/atlas/*/Role=NULL/Capability=NULL
/atlas/Role=*/Capability=NULL
/atlas/*=NULL/Capability=NULL
/atlas/Ro*/Capability=NULL
/atlas/Role/
/atlas/*=NULL/Capability=NULL
/atlas/*R*C*
```

In the edg version

TO BE COMPLETED

Interpretation of FQANs in the Match-Maker

TO BE COMPLETED

Matching mechanism 2

TO BE COMPLETED

Interpretation of wildcard characters 2

Here are some example to demonstrate how the matching works:

- Will match:

```
"/cms/Higgs*"                 "/cms/Higgsino"
"/cms/Higgs/*"                "/cms/Higgs/somesubgroup"
"/cms/Higgs/some*/Role=some*" "/cms/Higgs/somesubgroup/Role=somerole"
```

- Will NOT match:

```
 "/cms/Higgs"        "/cms/Higgsino"
 "/cms/Higgs/*"      "/cms/Higgs"
 "/cms/Higgs/some*"  "/cms/Higgs/somesubgroup/Role=somerole"
```

It means that the '*' character is interpreted separately inside the group and inside the role, since they are (should be) ortogonal to each other.

Interpretation of FQANs and mapping of users in DM

The DPM storage

DPM deals with 3 different mapping scenario:

- Request arrive to namespace with voms-proxy
- Request arrive to namespace with plain grid-proxy
- Request arrive out of namespace

Depending on the above situation different mechanism used to find the user's mapping. Request is in namespace when file path starts with `/dpm`.

1. Request arrive to namespace with voms-proxy
    - The certificate DN is matched against the virtual user database to determine the UID. If no matching found a new user is automatically created.
    - The certificate VOMS extensions are examined (all of them) and matched againts the virtual group database. Request will be granted with the matching group's membership, non matching FQANs will result automatic creation of new groups correspond to them. So there is no requirements that the primary FQAN must match.
2. Request arrive to namespace with plain grid-proxy
    - The certificate DN is matched againts the virtual user database to determine the UID. If no matching found a new user is automatically created.
    - The certificate subject DN is matched against the `/opt/lcg/etc/lcgdm-mapfile` which assigns a VO name/membership to the request. Then the request will be granted with group membership corresponds to that VO (i.e. to that string) in the virtual group database. In this case no secondary groups are possible, parsing stops at the first matching entry of `lcgdm-mapfile`.
3. Request arrive out of namespace
    - The `/etc/grid-security/grid-mapfile` is used to determine the matching. The first matching entry will be the request's UID and the unix secondary groups of this UID will be the request's groups. The request is then mapped to that unix account, that's why DPM still need pool account to be created.

DPM user mapping mechanism does not involve LCMAPs.

The LFC and DPM automatically strip off the roles and capabilities when they are NULL. The leading / is also removed. So an FQAN like

```
/atlas/usatlas/Role=NULL/Capability=NULL
```

is stored as

```
atlas/usatlas
```

And some fact to clarify the situation:

1. permissions on the name space are handled using all FQANs
2. file ownership is either inherited from the parent directory or is the primary FQAN (FQAN[0])
3. disk pool selection is done using the primary FQAN (FQAN[0])
4. accounting and quota is done using the primary FQAN (FQAN[0])

The Castor storage

In Castor there is no handling of VOMS FQANs and secondary groups, request are mapped to static accounts per VO. The developers expect to have an implementation of virtual UIDs and GIDs by the middle of 2008.

The dCache storage

In dCache the default still is not to handle VOMS FQANs and secondary groups, mapping request to one account per VO. However, the `vo-role-mapfile` allows selected DNs and/or selected VOMS FQANs to be mapped to selected static accounts. This new mechanism is expected to become the default for the time being. The developers expect to have an implementation of virtual UIDs and GIDs by the end of this year.

The LFC service

LFC deals with 2 different mapping scenario:

1. Requests arrives with voms-proxy
2. Requests arrives with grid-proxy

Depending on the above situation different mechanism used to find the user's mapping. Request can arrive only to namespace i.e to `/lfc`.

1. Requests arrives with voms-proxy
    - In this case LFC uses virtual group and user IDs and the mapping proceeds exactly in the same way as on case of DPM 1.)
2. Requests arrives with grid-proxy
    - In this case LFC performs the same mapping procedure as in case of DPM 2.)

Since no request can arrive out of name space, no pool accounts are needed on LFC. LFC user mapping mechanism does not involve LCMAPs.

The FTS service

There are three distinct roles on the FTS server.

1. User role which allows the user to submit to the service - upon submission the VO name is determined by the service:
    - All production FTS use a file mapping DN names to VO names. The grid mapfile mechanism
    - ...

Interpretation of wildcard characters 3

In DPM, LFC, FTS mapping mechanism the wildcard characters are not interpreted.

New proposal for FQAN matching in MW that uses VOMS

Here are the updated FQAN matching rules, with the input of today's discussion. Please comment.

FQAN composition rules:

- FQAN are composed by a group and an optional Role part.
    - Syntax is [/Role=rolename]
- always starts with a '/' and the VO name.
- By convention, the name of the first group is the name of the VO.
- Subgroups are separated from the group name with a '/' character.
- The following characters are allowed in group names: [0-9a-zA-z-_.]
- The following characters are allowed in role names: [0-9a-zA-z-_.]

FQANs should be matching against a pattern according to the following rules:

- The group and role parts of a FQAN are matched separately.
- In the absence of wildcards. The match should be done as exact strings.
- Accepted wildcards are '*' and '?'

- A Wildcard may match any character, including '/'
- /Role=NULL and /Capability=NULL MUST be eliminated before pattern matching.
- Matching is always case sensitive.

Examples:

pattern	text	match?
/atlas	/atlas	yes
	/atlas/Role=NULL	yes
	/atlas/prod	no
	/atlas/Role=sgm	no
	/atlas/prod/Role=null	no
	/atlassi	no
/atlas/Role=NULL	/atlas	yes
	/atlas/Role=NULL	yes
	/atlas/prod	no
	/atlas/Role=sgm	no
	/atlas/prod/Role=NULL	no
	/atlassi	no
/atlas/Role=*	/atlas	no
	/atlas/Role=NULL	no
	/atlas/prod	no
	/atlas/Role=sgm	yes
	/atlas/prod/Role=NULL	no
	/atlassi	no
/atlas/prod/Role=*	/atlas	no
	/atlas/Role=NULL	no
	/atlas/prod	no
	/atlas/Role=sgm	no
	/atlas/prod/Role=NULL	no
	/atlas/prod/Role=sgm	yes
	/atlassi	no
/atlas*	/atlas	yes
	/atlas/Role=NULL	yes
	/atlas/prod	yes
	/atlassi	yes
	/atlas/Role=sgm	no
	/atlas/prod/Role=NULL	yes
	/atlas/prod/Role=sgm	no
/atlas/*	/atlas	no
	/atlas/Role=NULL	no
	/atlas/prod	yes
	/atlassi	no
	/atlas/Role=sgm	no
	/atlas/prod/Role=NULL	yes
	/atlas/prod/Role=sgm	no
/atlas*/Role=sgm	/atlas	no
	/atlas/Role=NULL	no
	/atlas/prod	no
	/atlas/Role=sgm	yes
	/atlas/prod/Role=NULL	no
	/atlas/prod/Role=sgm	yes
	/atlassi/Role=sgm	yes
	/atlassi	no
/atlas/*/Role=sgm	/atlas	no
	/atlas/Role=NULL	no
	/atlas/prod	no
	/atlas/Role=sgm	no
	/atlas/prod/Role=NULL	no
	/atlas/prod/Role=sgm	yes
	/atlassi/Role=sgm	no
	/atlassi	no

Important notes to add

Note that in this last example, /atlas/*/Role=sgm can never match to /atlas//Role=sgm, since groups and roles are matched separately. Furthermore, /atlas//Role=sgm is not a valid FQAN.

Further example: If the pattern in the policy is /atlas/prod, and in the FQAN you get only /atlas, this will not be a match, since there are users in /atlas that are not members of /atlas/prod

Conversely, if the pattern in the policy is /atlas, and in the FQAN you get /atlas/prod, this will not be a match, since the policy requires explicitly the main group. However, since having /atlas/prod in the AC implies also having /atlas, there will be a match with some FQAN.

Links

- Deployment status of roll out in EGEE
- CMS requirements
- Current understanding, discussion on roles/groups

-- ClaudioGrandi - 26 Jun 2007

This topic: EGEE > EGEEgLiteJobPriorities
Topic revision: r14 - 2007-07-16 - OscarKoeroo