SE Security

Rémi Mollon, Ákos Frohner

EGEE'08, Istanbul, September 2008

www.eu-egee.org

gLite SE Authorization

Background: EGEE-II/MJRA1.7: gLite Authorization

- **VOMS Attributes**
- **Posix file permissions**
  - Virtual user and group ids
- **Permissions on spaces**
- **Implementations**
  - DPM/LFC
  - Castor
  - Dcache
  - StoRM

Identity

Identity and group information

- **User ID = X509 certificate DN**

  /DC=ch/DC=cern/.../CN=652521/CN=Remi Mollon

- **Groups = VOMS FQANs:**
  - VOMS group: /biogrid, /biogrid/analysis
  - VOMS role: /biogrid/H5N10/Role=production
- **VOMS generic attributes (key, value pairs) not used**

gLite SE Authorization

- VOMS Attributes
- Posix file permissions
  - Virtual user and group ids
- Permissions on spaces
- Implementations
  - DPM/LFC
  - Castor
  - Dcache
  - StoRM

Basic permissions

POSIX style file and directory permissions

- owner = DN of the creator
- group = first VOMS FQAN of the creator
- basic read/write/execute permissions for user/group/others

Exact match: any of the user's DN or VOMS FQANs has to match exactly one of the permissions on a file.

```
$ dpns-mkdir /dpm/cern.ch/home/dteam/rmollon
$ dpns-chmod 0755 /dpm/cern.ch/home/dteam/rmollon
$ dpns-getacl /dpm/cern.ch/home/dteam/rmollon
# file: /dpm/cern.ch/home/dteam/rmollon
# owner: /DC=ch/DC=cern/.../CN=Remi Mollon
# group: dteam
user::rwx
group::r-x                    #effective:r-x
other::r-x
```

Access Control List

POSIX access control list (ACL):

- Access ACLs: set permissions for other users and groups
- Default ACLs on directories: they are inherited by each entry created within.

```
$ dpns-setacl -m d:u::rwx,d:g::r-x,d:o:- /dpm/cern.ch/home/dteam/rmollon
$ dpns-setacl -m 'g:biomed:r-x,m:rwx' /dpm/cern.ch/home/dteam/rmollon
$ dpns-setacl -m \ 'u:/DC=ch/DC=cern/.../CN=Akos Frohner:rwx,m:rwx'
    /dpm/cern.ch/home/dteam/rmollon
$ dpns-getacl /dpm/cern.ch/home/dteam/rmollon
...
user:/DC=ch/DC=cern/.../CN=Akos Frohner:rwx  #effective:rwx
group:biomed:r-x          #effective:r-x
mask::rwx
other::r-x
default:user::rwx
default:group::r-x
default:other::---
```

Extra Details

- **Set-group behavior:**
  - Client's FQANs creating a new file: /dteam
  - Directory's group without 'set-gid': /dteam/sam
    New file's group: /dteam – inheriting the client's first FQAN
  - Directory's group with 'set-gid': /dteam/sam
    New file's group: /dteam/sam – inheriting the directory's group
- **Secondary groups: all VOMS attributes are considered**
  - File's permission:
    user: /DC=ch/.../CN=Remi Mollon
    group: /dteam/sam
  - Client's FQANs: /dteam, /dteam/sam, /dteam/sam/Role=...

Space protection

Space definition:

logical view of online storage area, orthogonal to the namespace, characterized by storage attributes (i.e. disk/tape, guaranteed size, owner)

Access control eventually foreseen:

- Owner – DN, ACL entities – VOMS FQANs/DNs
- Operations: release, update, read-from, write-to, stage-to, replicate-from, purge-from, modify-acl, query

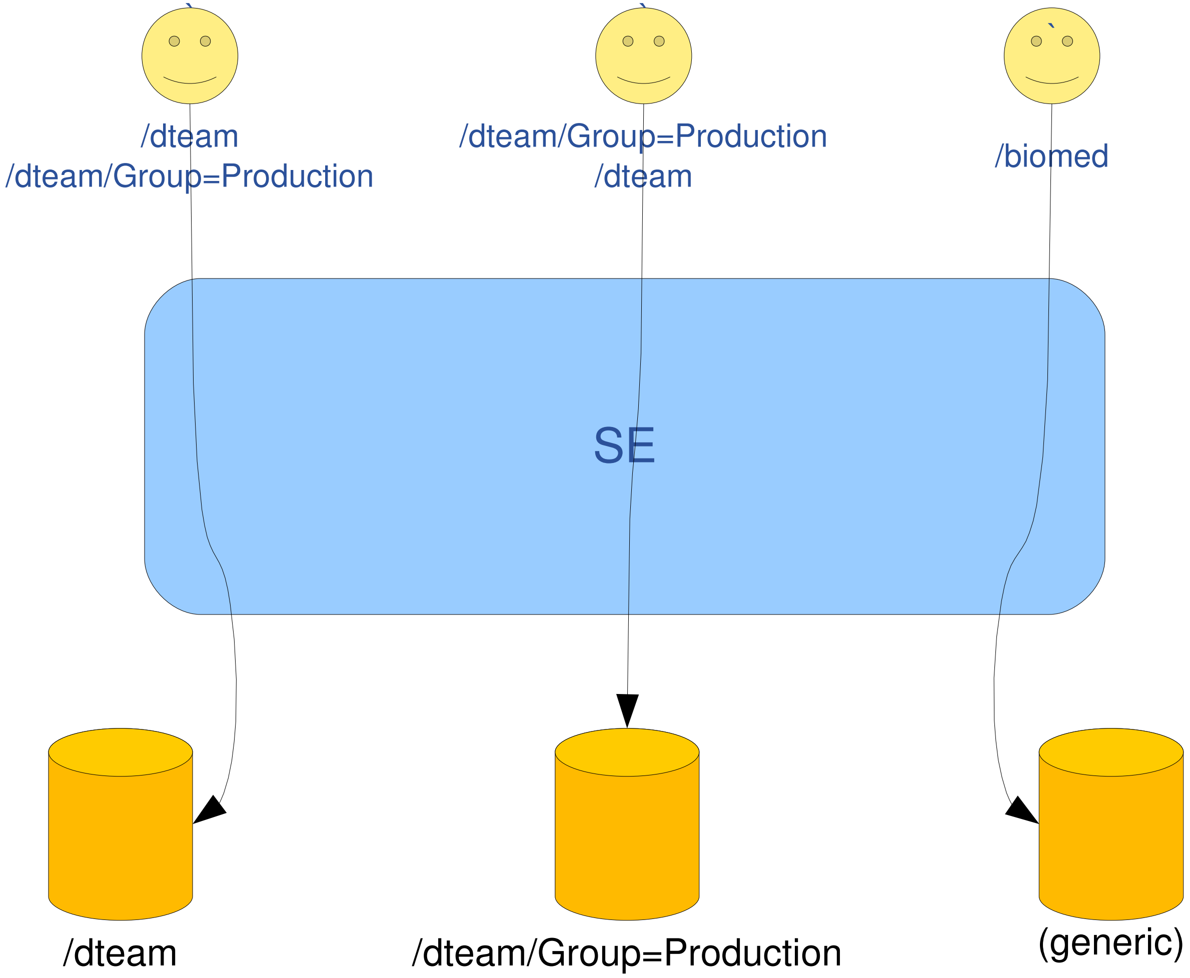
Still under discussion:

- Secondary groups (i.e. all client FQANs are used for authz.)
- Negative ACL (i.e. /dteam/sam, except /.../CN=Remi Mollon)
- Wild card matching (i.e. /dteam/prod*)

Use case: tape recall by production manager to a VO space

pool selection

Pool selection when creating a new file in the SE



- exact match based on the first FQAN
- match pool with enough space
- otherwise match with the generic pool

SE Implementations

- VOMS Attributes
- Posix file permissions
  - Virtual user and group ids
- Permissions on spaces
- Implementations
  - DPM/LFC
  - Castor
  - Dcache
  - StoRM

DPM/LFC

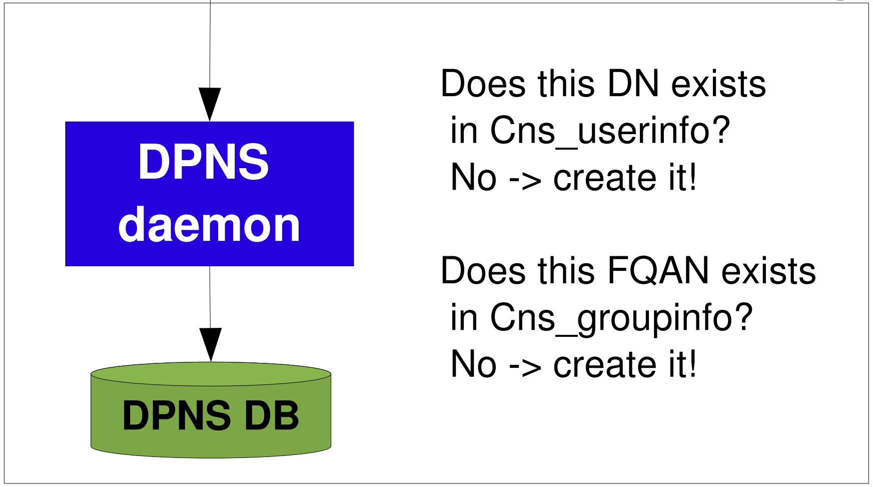
(see the examples before)

- X509 (or Kerberos 5) based authentication
- Support for secondary groups on files
- gridmap-file: if the client does not have VOMS AC, the VO/group is determined via an SE specific gridmap-file
- Space permission:
  - write permission for a single group (ie. VOMS FQAN)
  - list of groups, with secondary group support in the next release

DPM/LFC: virtual uid and gid

DN: /DC=ch/DC=cern/.../CN=Remi Mollon

```
$ voms-proxy-init -voms dteam
$ dpns-ls /dpm/cern.ch/home/dteam/rmollon
drwxr-xr-x   0 101  103  0 ... /dpm/cern.ch/home/dteam/rmollon
```



Does this DN exists
in Cns_userinfo?
No -> create it!

Does this FQAN exists
in Cns_groupinfo?
No -> create it!

DPNS DB

Cns_userinfo
101 /DC=ch/DC=cern/../CN=Remi Mollon

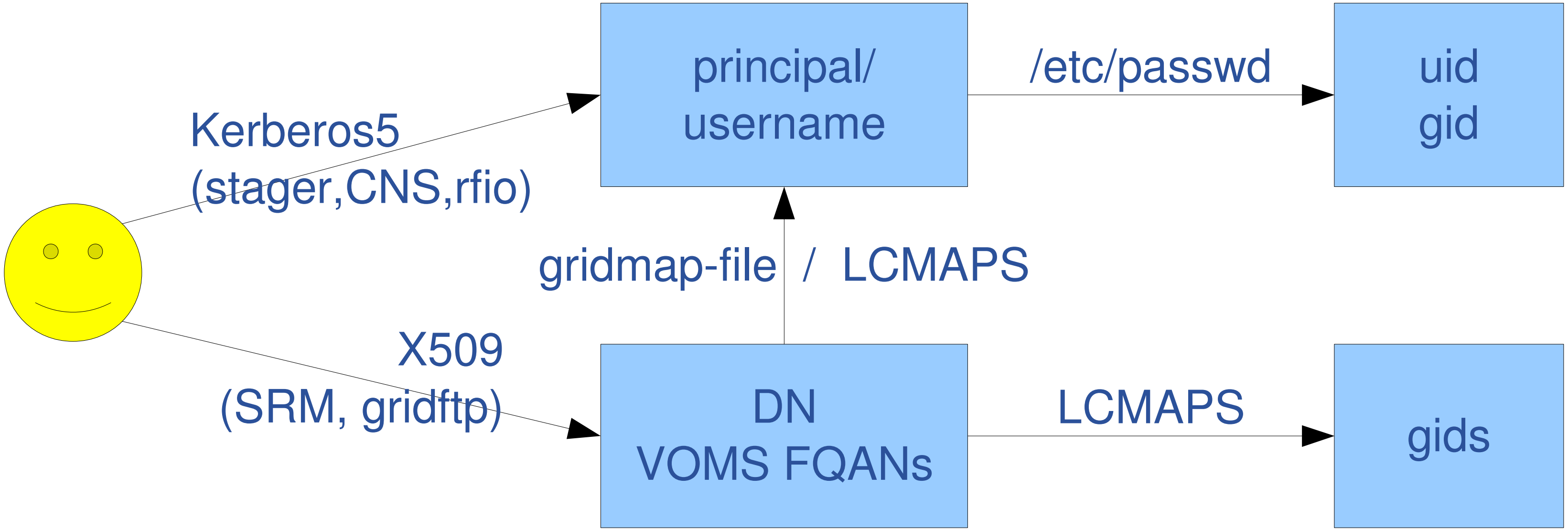
Cns_groupinfo
103 dteam

- no need to create pool accounts
- no need to change the /etc/passwd file
- faster check on ACL than with string/pattern matching on DN/FQAN
- Mapping multiple DNs (Krb5 principals) into the same uid

Castor specialties

- **Client authn/authz is primary goal**
  - back-end services are in a controlled environment, so authn/authz of administrative actions comes later
- **X509 or Kerberos5**
  - every CERN user has Kerberos principal
  - speed of Kerberos5 is better than X509
- **Virtual UID/GID – not yet**
  - stager scheduler requires real uid/gid
  - every internal user is already in the CERN user DB
- **Secondary groups – not yet**
  - passing secondary group information needs lot of changes
  - How to add secondary group information in Kerberos?

Castor Authentication



- **stager, CNS and rfio currently uses uid/gid authn**
  - first goal is to improve this authentication
- **SRM and GridFTP use X509 with pool accounts**
  - effective permissions are at group level
  - goal is to map individual DNs into individual uids
  - shortcut: CERN DN contains the username

Castor Authorization

- **Name Service**
  - current authorization is by Posix uid/gid numbers
  - mapping from Kerberos and X509 to uid/gid(s) solves the problem
  - non-CERN users are problematic...
- **Stager and SRM**
  - checks in the name service the file permissions
  - stores the uid/gid(s) with the request
- **I/O protocols (rfio, gridftp)**
  - one-time services are started for each request
  - requests are granted with a one-time token
  - the authenticated and mapped uid/gid is compared with the one in the request too
- **xrootd**
  - authz. in the redirector, granted as a one-time token

StoRM

- **X509 based authentication**
  - Mapping DN and VOMS FQANs to uid/gids via LCMAPS
  - Uses system uid/gid
- **File permissions using the underlying file system**
  - Just-in-time: temporary ACL for the time of the access (SRM request)
  - Ahead-of-time: ACL in the file system according to the authorization policy, when the file is created
  - Any local file system with ACL support
- **Can apply a set of ACL entries on new files**
  - Authorization policy is configurable at system level
- **Space permissions**
  - Per VO access for a storage area
  - Planned: flexible per user/group permissions

dCache

- **Supported authentication methods:**
  - X509
  - Kerberos 5
  - SAML based grid VO role mapping
  - GUMS
- **VOMS support considering the first FQAN**
- **Implementation via virtual uid/gid**

Work in progress (for the 1.9.x series):

- **NFS 4.1 style ACL**
  - includes set-group directory and default ACL
- **Permission on spaces**
- **Secondary groups**

Further reading

EGEE-II/MJRA1.7: gLite Authorization
https://edms.cern.ch/document/887174/1

- **VOMS Attributes**
- **Posix file permissions**
  - Virtual user and group ids
- **Permissions on spaces**
- **Implementations**
  - DPM/LFC
  - Castor
  - Dcache
  - StoRM