Grid Authentication and Authorisation Issues

Ákos Frohner at CERN

Overview

- Setting the scene: requirements
- “Old style” authorisation: DN based gridmap-files
- Overview of the EDG components
- VO user management: VOMS
- “Login”: short lifetime proxy certificates
- Authorisation in Java web services
- Authorisation for a site: LCAS, LCMAPS

Requirements

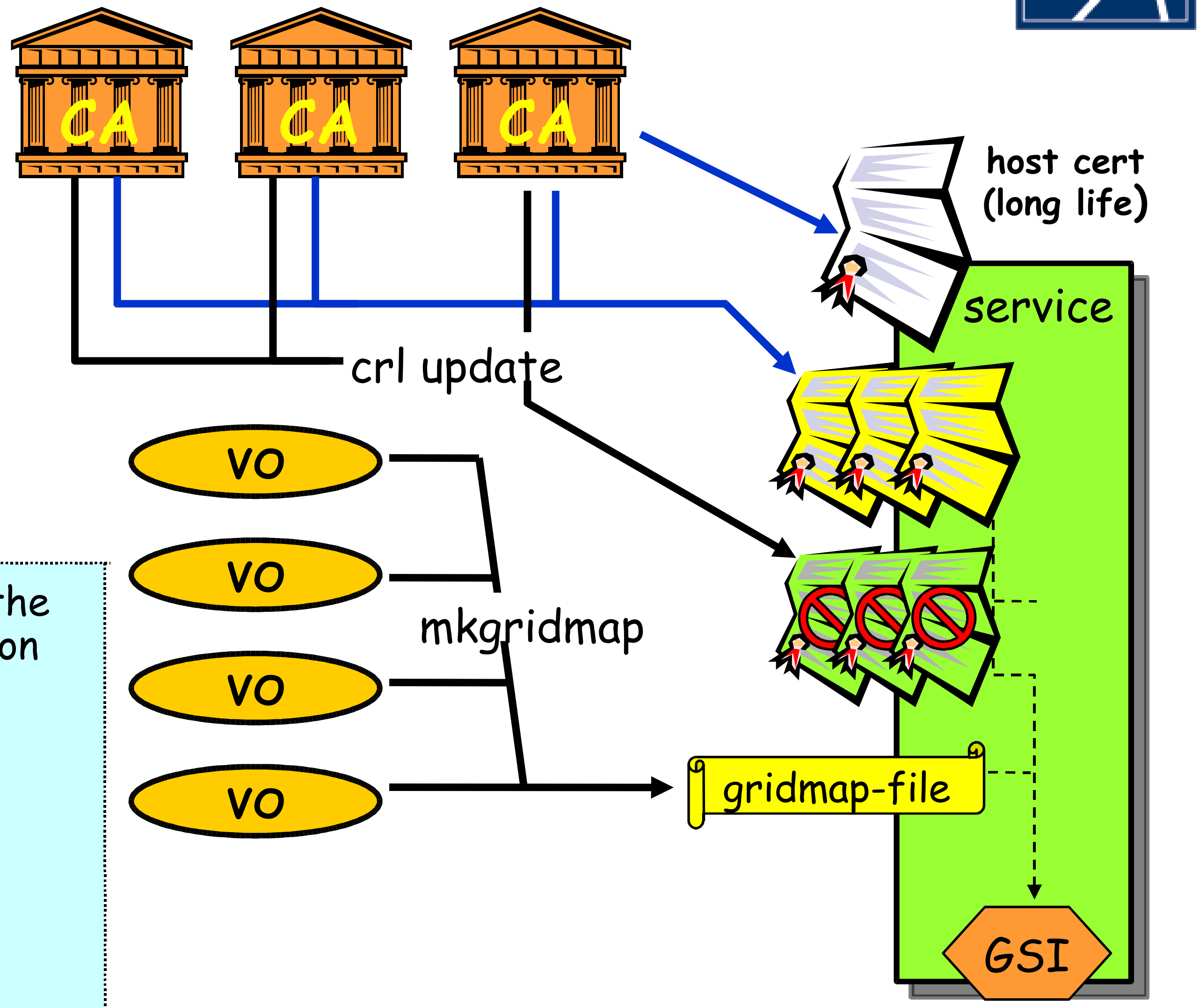
- A grid security system requires:
  - User to be authenticated by a service
  - The service to gather additional information associated with the user or the actual session (e.g. group membership, role)
  - The service to gather additional information associated with the protected service or object (e.g. file permissions)
  - The checking of any local policy applicable to the situation
  - The making of an authorization decision based on the identity of the user and the additional information
  - The Users to access resources in a global Grid environment without the need for individual accounts at various sites, while allowing resource providers to keep control over access to their resources.
- EDG gathered 112 requirements: Authentication, Authorisation, Confidentiality, Integrity, and Non-repudiation

“Old-style” Service

high frequency

low frequency

Backward compatibility on the service side: one can generate gridmap-files from the VO userlist for existing services based on GSI.



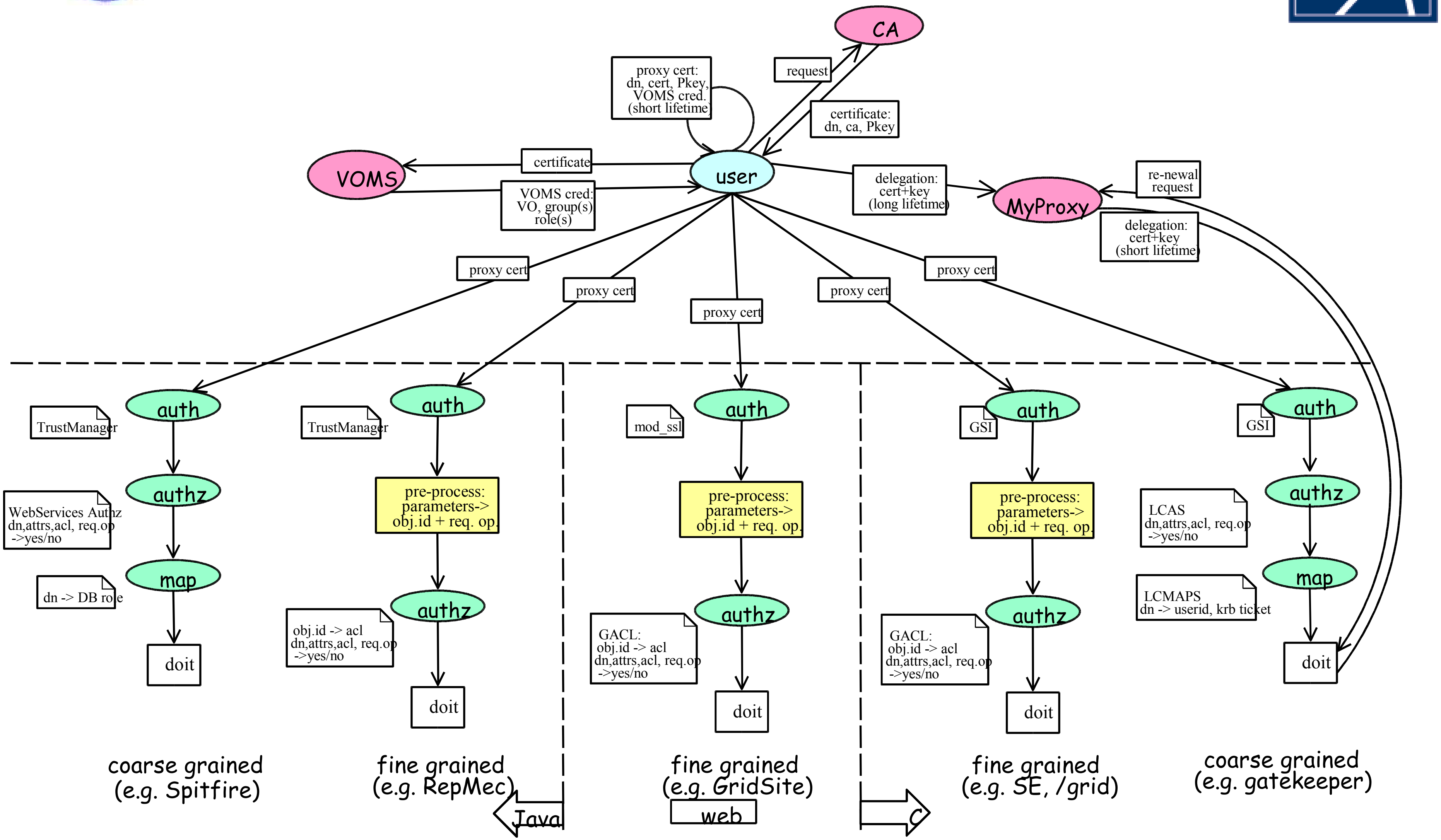
Old-style services still use the gridmap-file for authorization

- gridftp
- EDG 1.4.x services
- EDG 2.x service in compatibility mode

The Components

- GSI based or compatible authentication
- grid-mapfile or VOMS based authorization (can be both)
- policy or ACL based access control
  - coarse and fine grained solutions
  - access control description's syntax is not standard
- implemented alternatives:
  - edg-java-security for Java web services
  - GSI/LCAS/LCMAPS for native C/C++ services
  - mod_ssl/GACL for Apache based web services
  - Slashgrid for transparent filesystem ACLs and GridSite

Overview of the Components



coarse grained
(e.g. Spitfire)

fine grained
(e.g. RepMec)

fine grained
(e.g. GridSite)

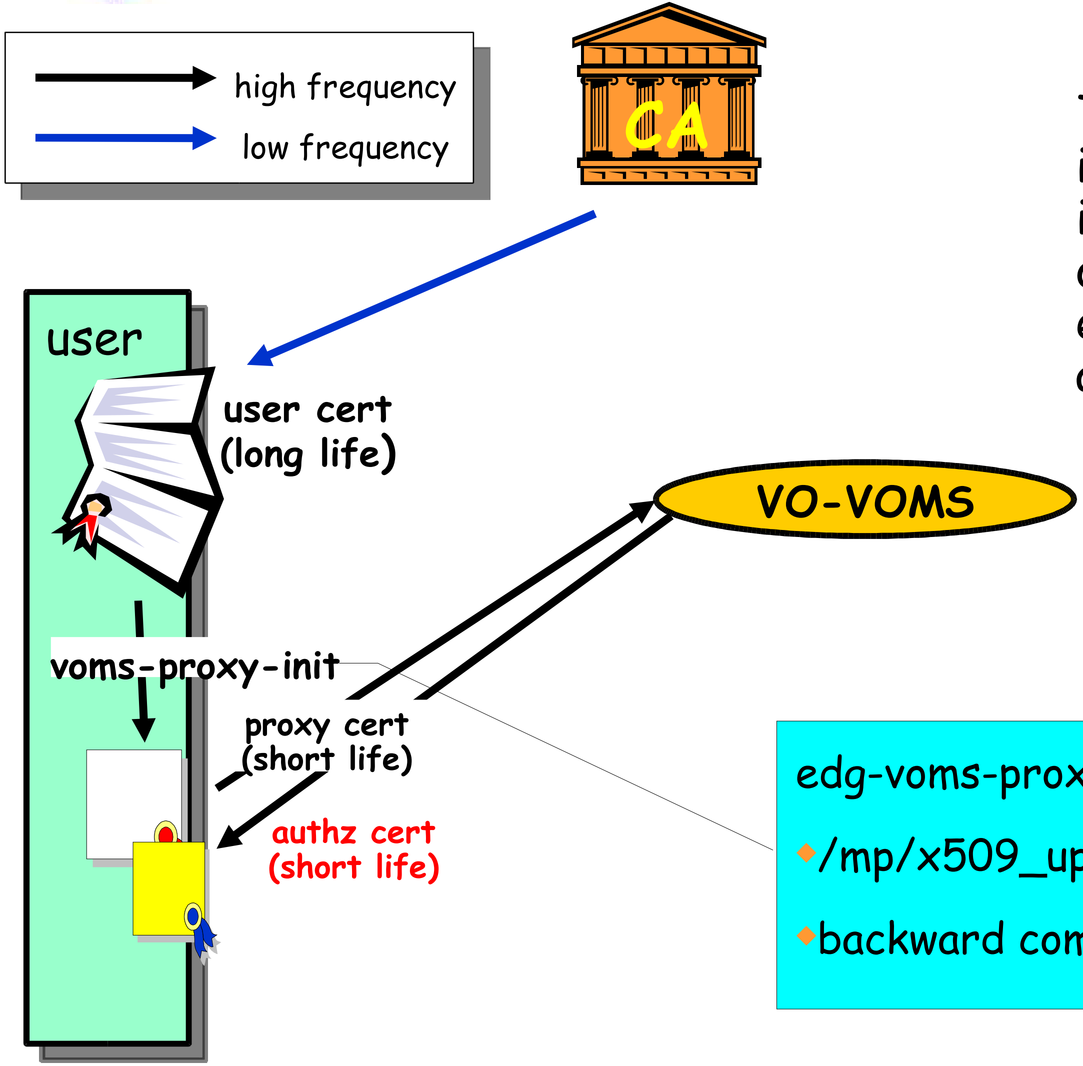
fine grained
(e.g. SE, /grid)

coarse grained
(e.g. gatekeeper)

VOMS: Virtual Organization Management Service

- Issues credentials to prove group/role/VO membership
  - standard RFC 3281 Attribute Certificate format
  - single string attributes – FQAN
- Core service: standalone daemon for the "login"
  - single purpose – high performance
- Administrative service: web service with API, command line and web user interface
  - for administration and registration
- Migration tools for gridmap-files and VO-LDAP servers

“Login”

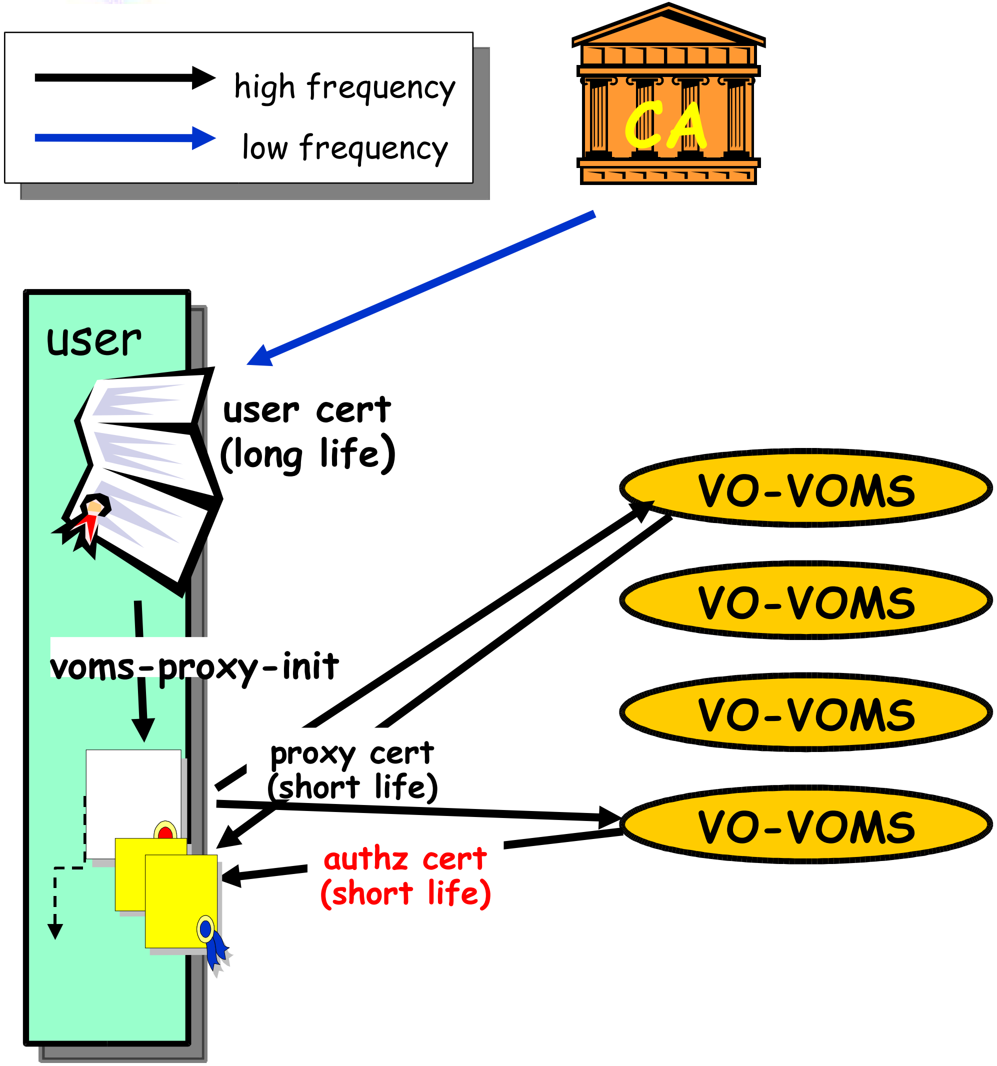


The credential created in the “login” procedure is backward compatible: one can use it with the existing services, which are based on GSI

edg-voms-proxy-init -voms iteam

- /mp/x509_up‹UID› (normal proxy location)
- backward compatible proxy format

Multi-VO “Login”



```
voms-proxy-init -voms iteam -voms wp6
```

- single proxy certificate is generated
- each VO provides a separate VOMS credential
  first one is the default VO
- each VOMS credential contains multiple group/role entries
  first one is the default group

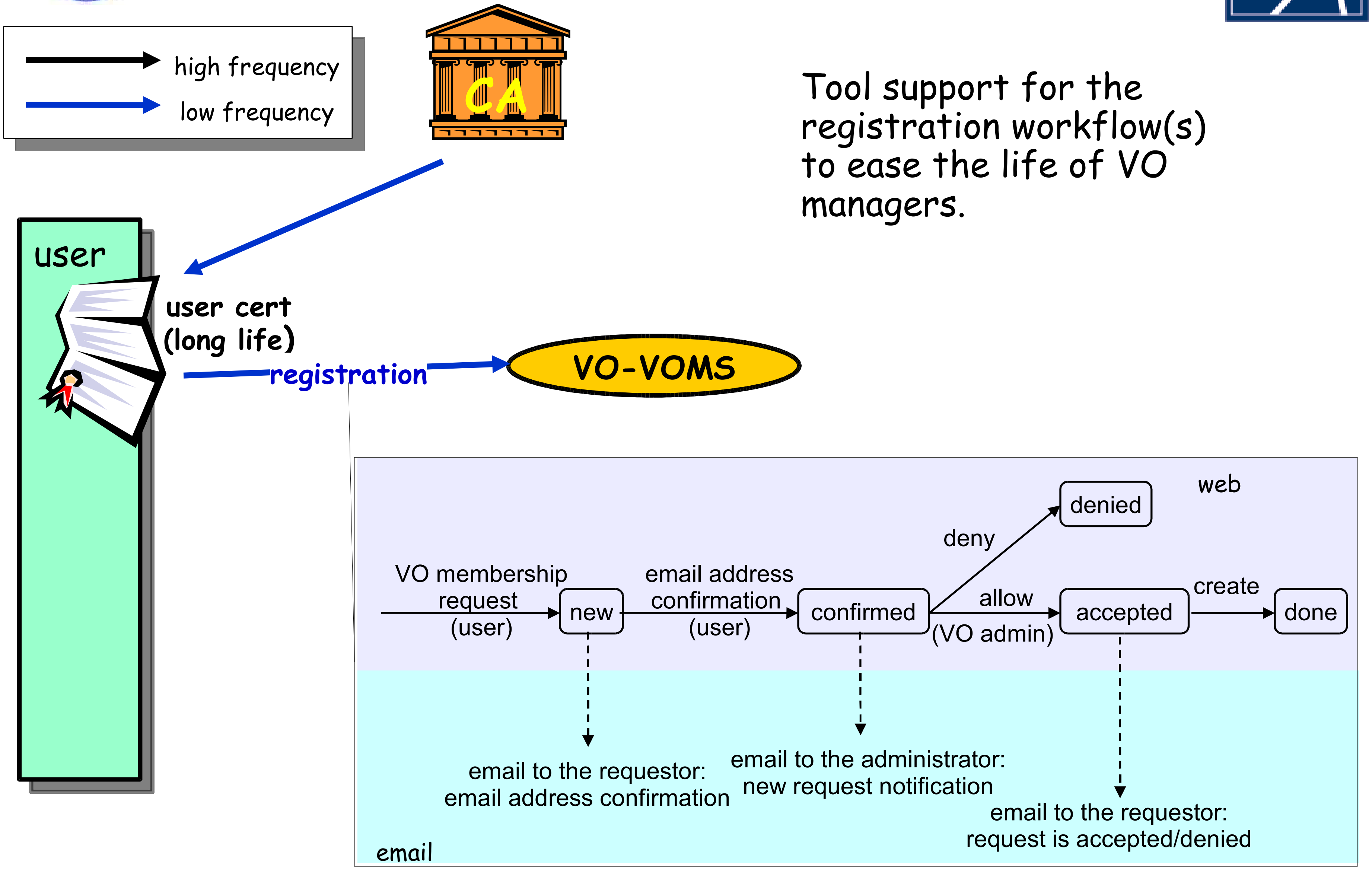
One can be member of many VOs and use their resources at the same time. The VO specific credentials are separate, but collected into the same proxy certificate.

VOMS FAQ

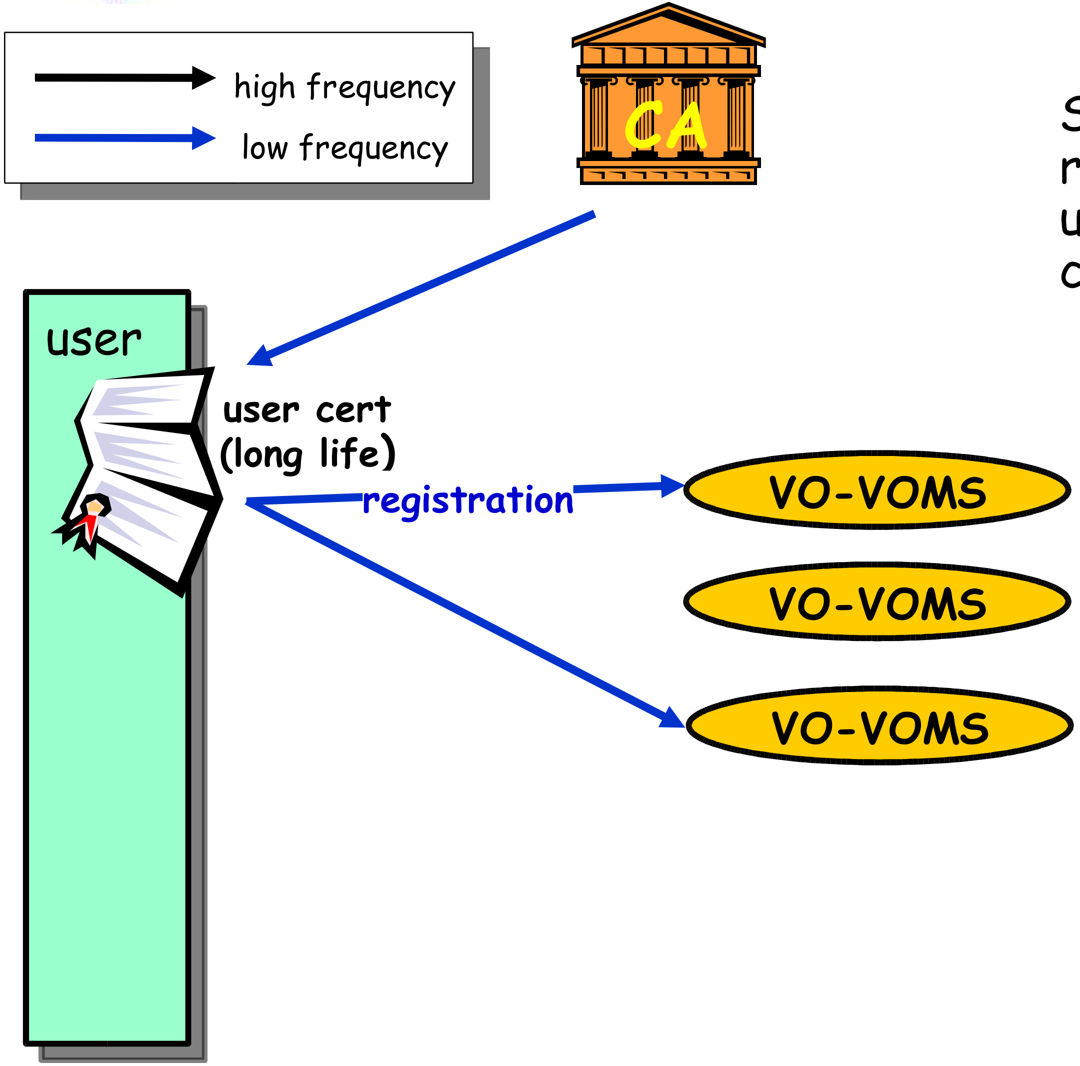
- No instant effect: the user has to “log-in”, using voms-proxy-init, to be notified of any VO change
- Delegation: a user cannot delegate her/his groups to someone else (unless s/he is a group-admin); no user groups
- Indirect effect on the policy: VOMS may name groups/roles in order to implement a policy, but it is up to the services to enforce it and up to the resource owner no to override it
- VOMS is not used to implement fine grained ACLs: it does not store file names or job ids (although it has its own ACLs for group/role administration)

VOMS Registration

Tool support for the registration workflow(s) to ease the life of VO managers.


high frequency
low frequency
CA
user
user cert
(long life)
registration
VO-VOMS
web
denied
deny
VO membership request (user)
new
email address confirmation (user)
confirmed
allow
(VO admin)
accepted
create
done
email to the requestor: email address confirmation
email to the administrator: new request notification
email to the requestor: request is accepted/denied
email

Multi-VO Registration



Support for multi-VO registration and login using the same user certificate.

VO administration operations

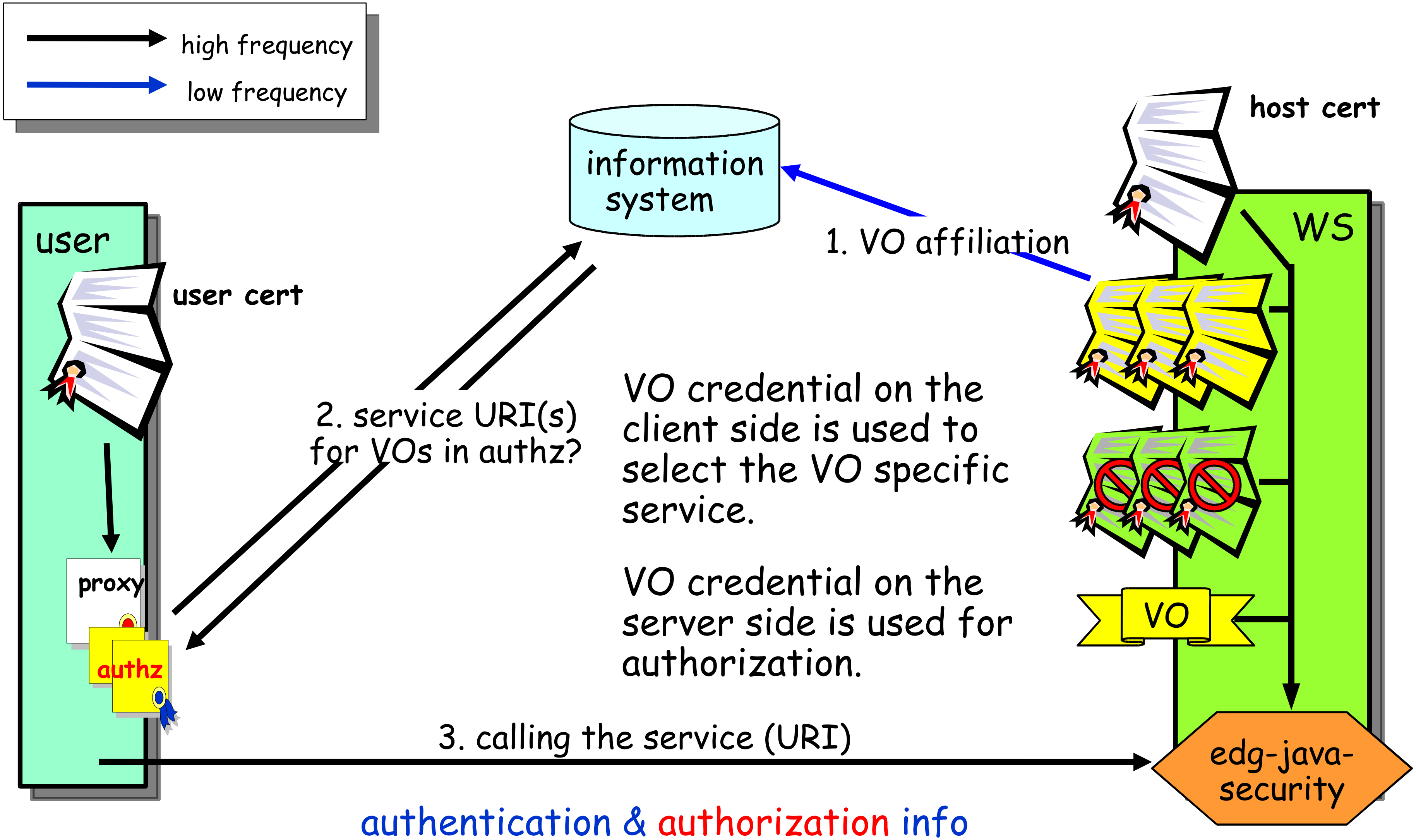
- create/delete (sub) group/role/capability
- add/remove member of g/r/c
- get/set ACLs for these operations

VO registration tasks

user requested administrative operation; e.g.:

user registration = add member

Java Web Service



VO credential on the client side is used to select the VO specific service.

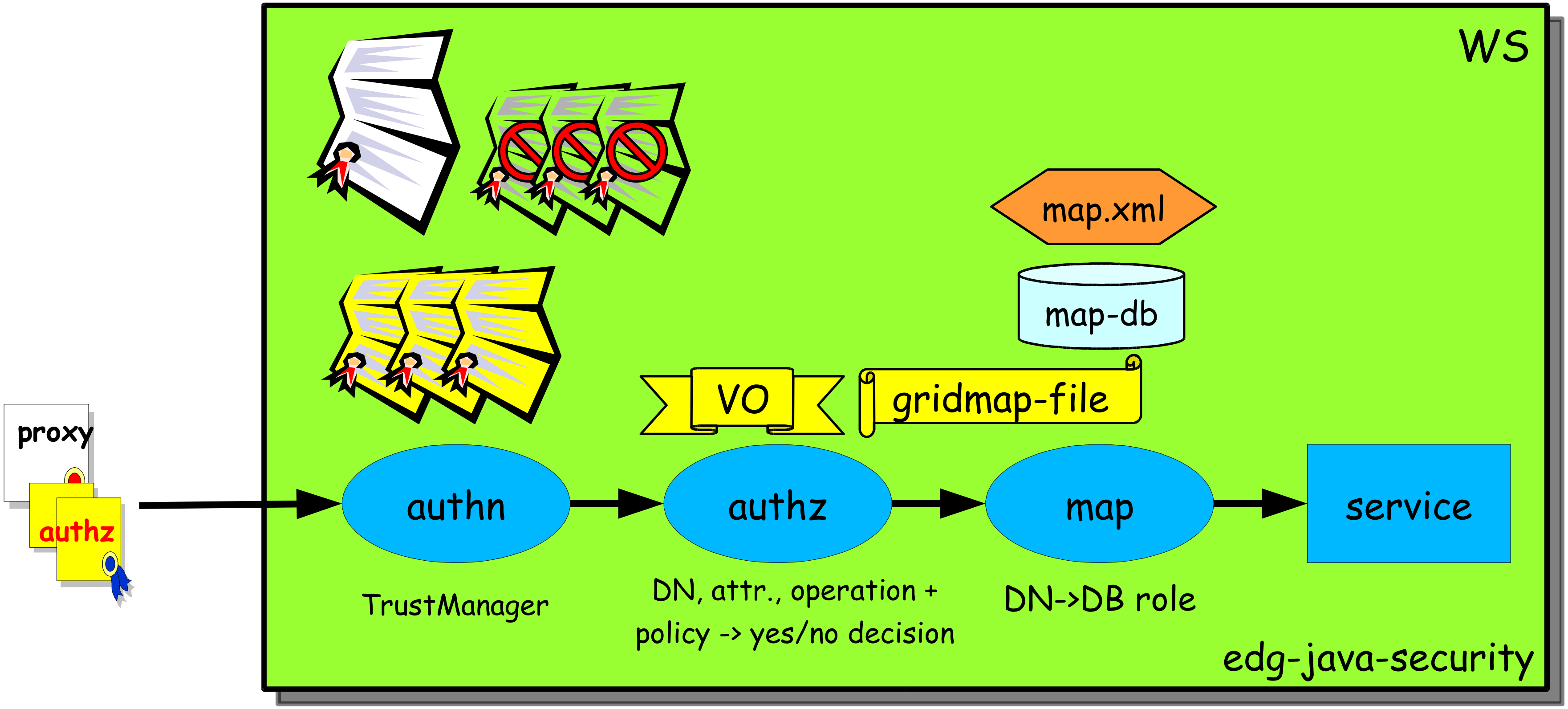
VO credential on the server side is used for authorization.

authentication & authorization info

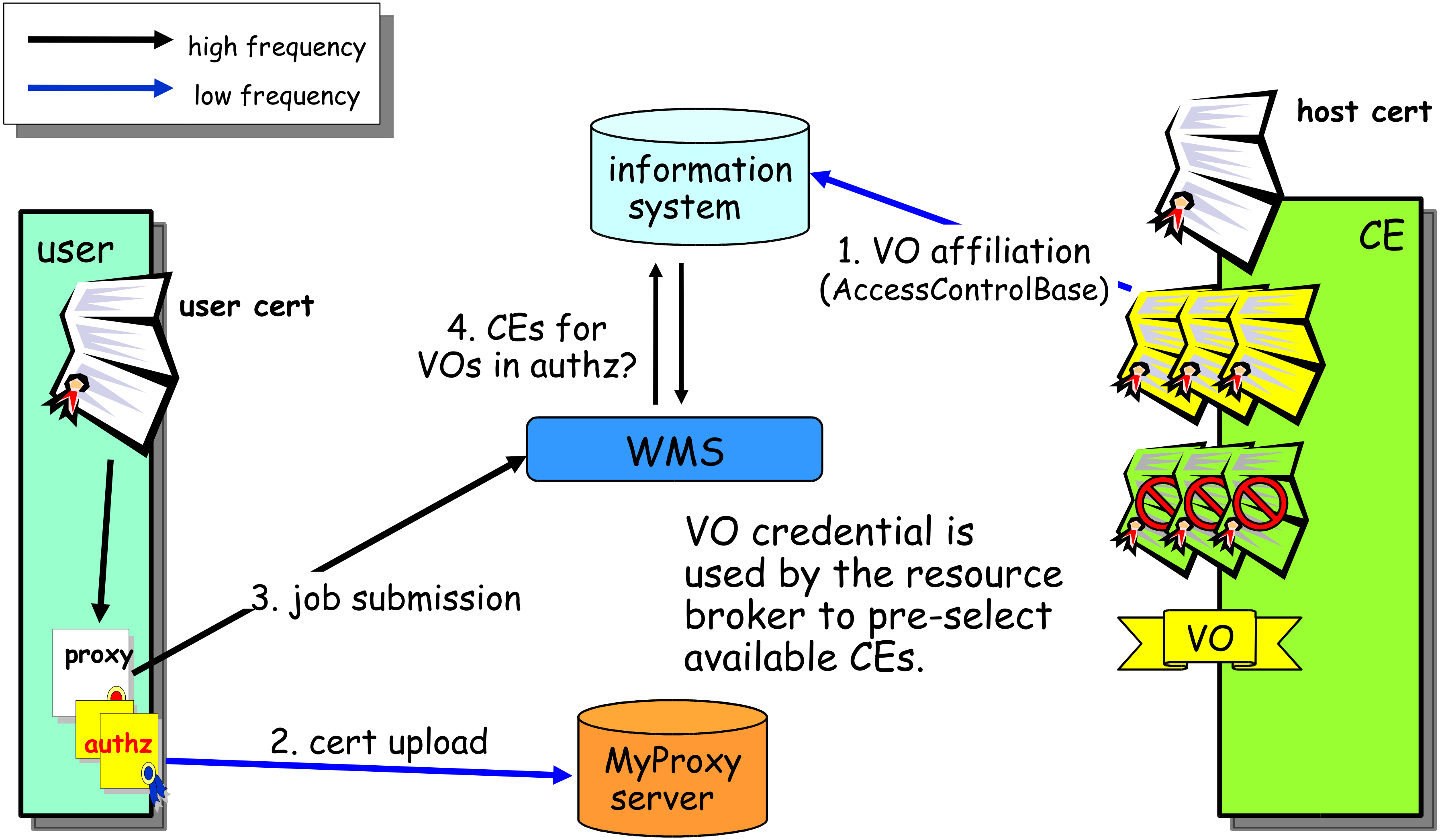
edg-java-security

- Trust manager
  - GSI compatible authentication (supporting proxy chain)
  - Adapters to HTTP and SOAP
  - Currently deployed for Tomcat4
  - VOMS credential verification
- Authorization Manager
  - Authorization and mapping for Java services
  - Plug-in framework for maps: database, XML file and for backward compatibility: gridmap-file
  - Handles VOMS attributes

Inside the Java Web Service



Job Submission



VO credential is used by the resource broker to pre-select available CEs.

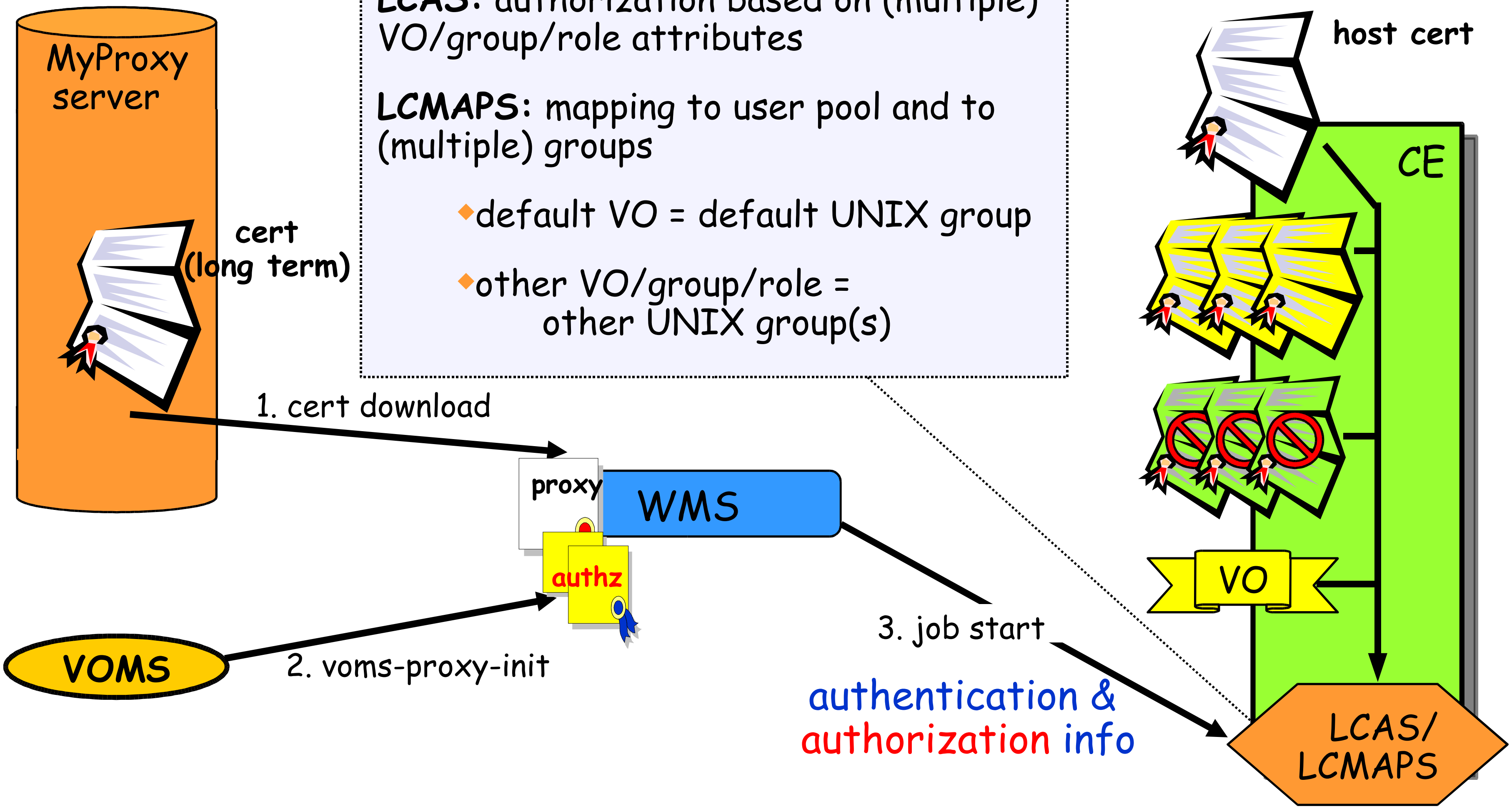
Arriving to a Computing Element

VO credential for authorization and mapping on the CE.

LCAS: authorization based on (multiple) VO/group/role attributes

LCMAPS: mapping to user pool and to (multiple) groups

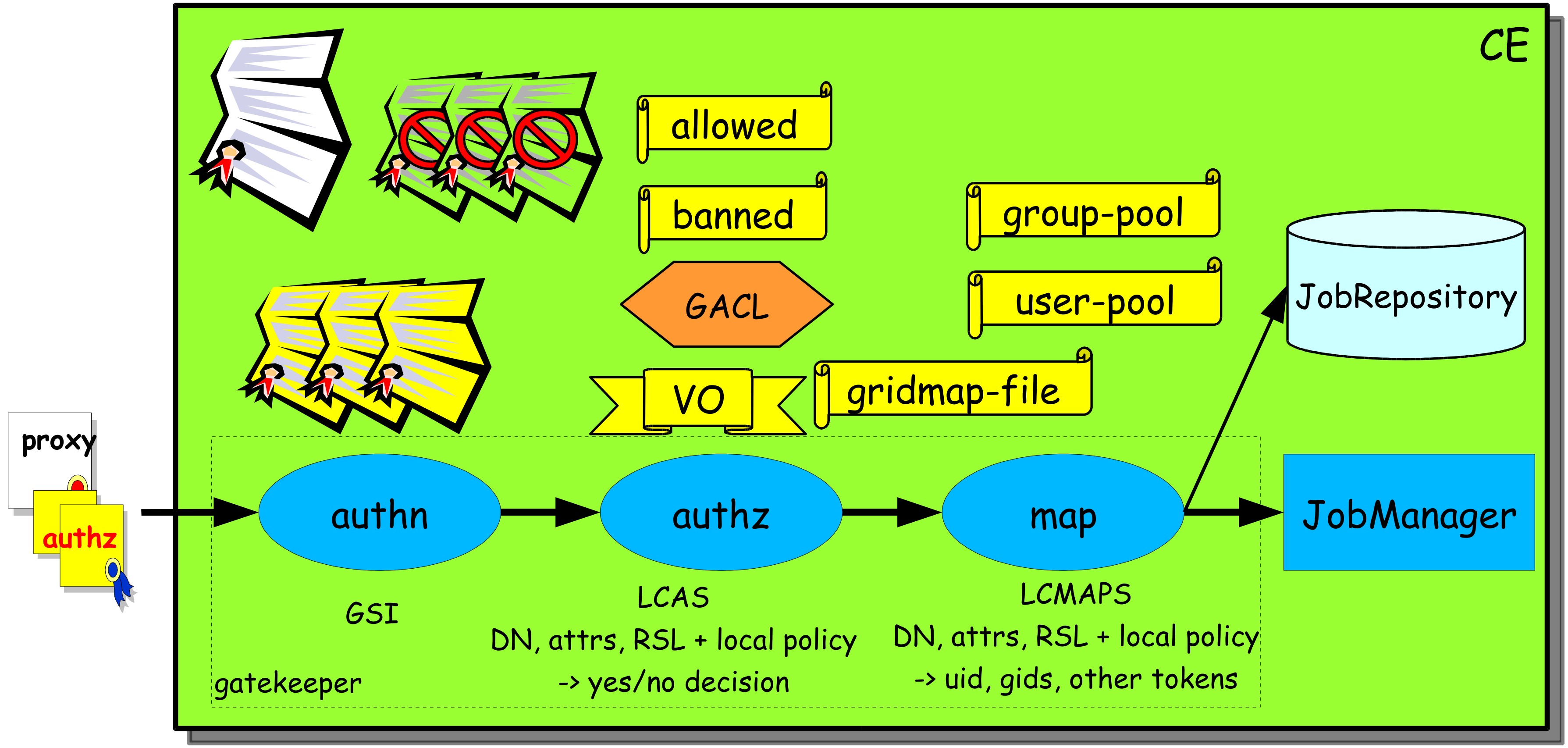
- default VO = default UNIX group
- other VO/group/role =
  other UNIX group(s)



LCAS and LCMAPS

- Local Centre Authorization Service (**LCAS**)
  - Handles authorization requests to local fabric
    - authorization decisions based on proxy user certificate and job specification;
    - supports grid-mapfile mechanism.
  - Plug-in framework (hooks for external authorization plugins)
    - allowed users (grid-mapfile or allowed_users.db), banned users (ban_users.db), available timeslots (timeslots.db), GACL
    - plugin for VOMS (to process authorization data)
- Local Credential Mapping Service (**LCMAPS**)
  - provides local credentials needed for jobs in fabric
  - mapping based on user identity, VO affiliation, local site policy
  - plug-ins for local systems (Kerberos/AFS, LDAP nss)

Inside a Computing Element



Summary of Issues Resolved

- Compatibility with existing systems
  - Tomcat - edg-java-security
  - Apache – gridsite
  - gridmap-file – LCAS and LCMAPS
- Credential Mapping
  - implementation on computing element
- Credential Renewal
  - for long running jobs
- Delegation
  - absent from standard HTTPS

More Information

- European DataGrid Project Security Coordination Group
  http://cern.ch/hep-project-grid-scg
- LCAS/LCMAPS homepage
  http://www.dutchgrid.nl/DataGrid/wp4/lcas/
- Java Security
  http://cern.ch/grid-data-management/security/
- GridSite
  http://www.gridpp.ac.uk/gridsite/
- VOMS
  http://grid-auth.infn.it/
  http://cern.ch/edg-wp2/security/voms