Overview of the New Security Model

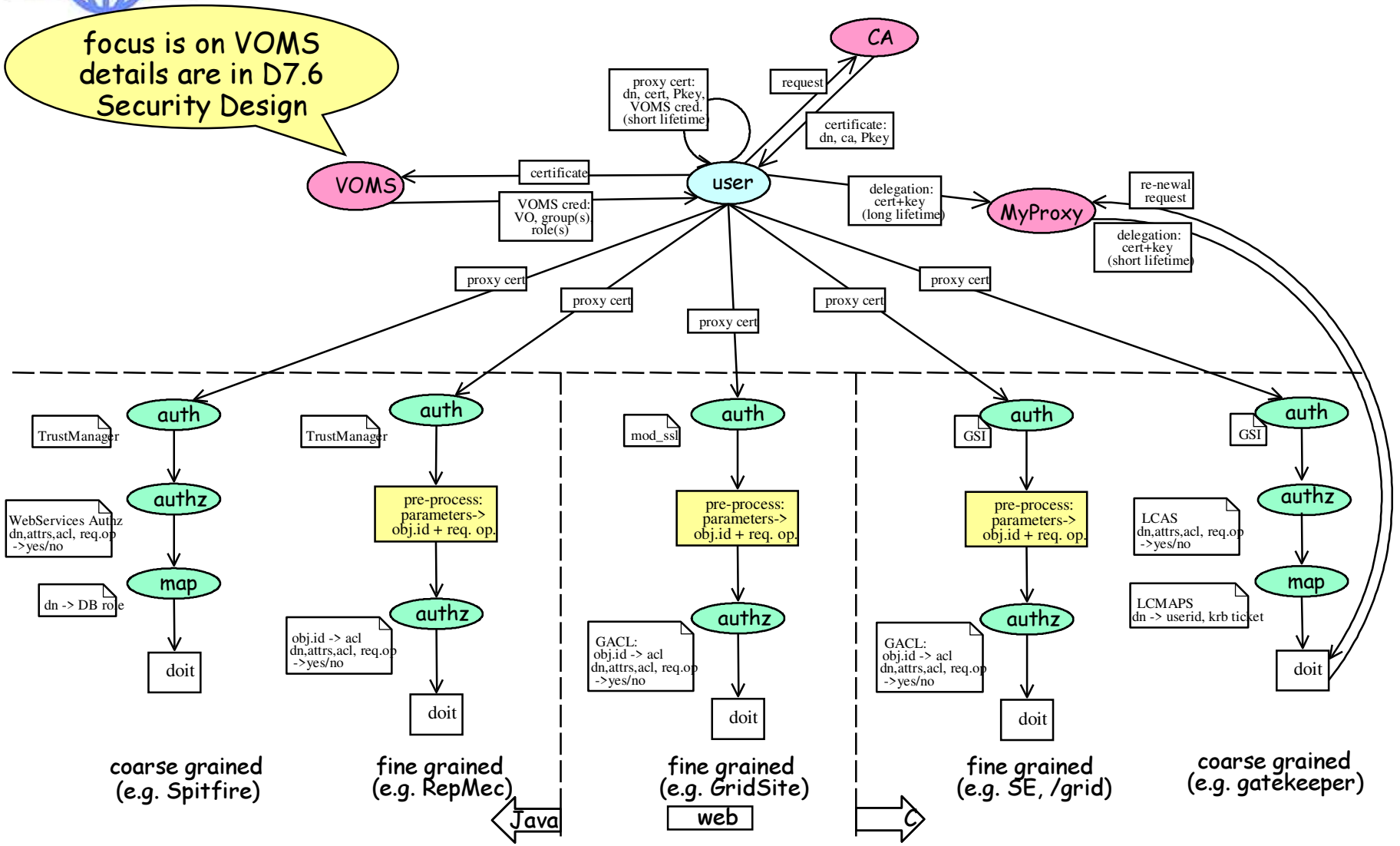
Akos Frohner (CERN)

WP8 Meeting

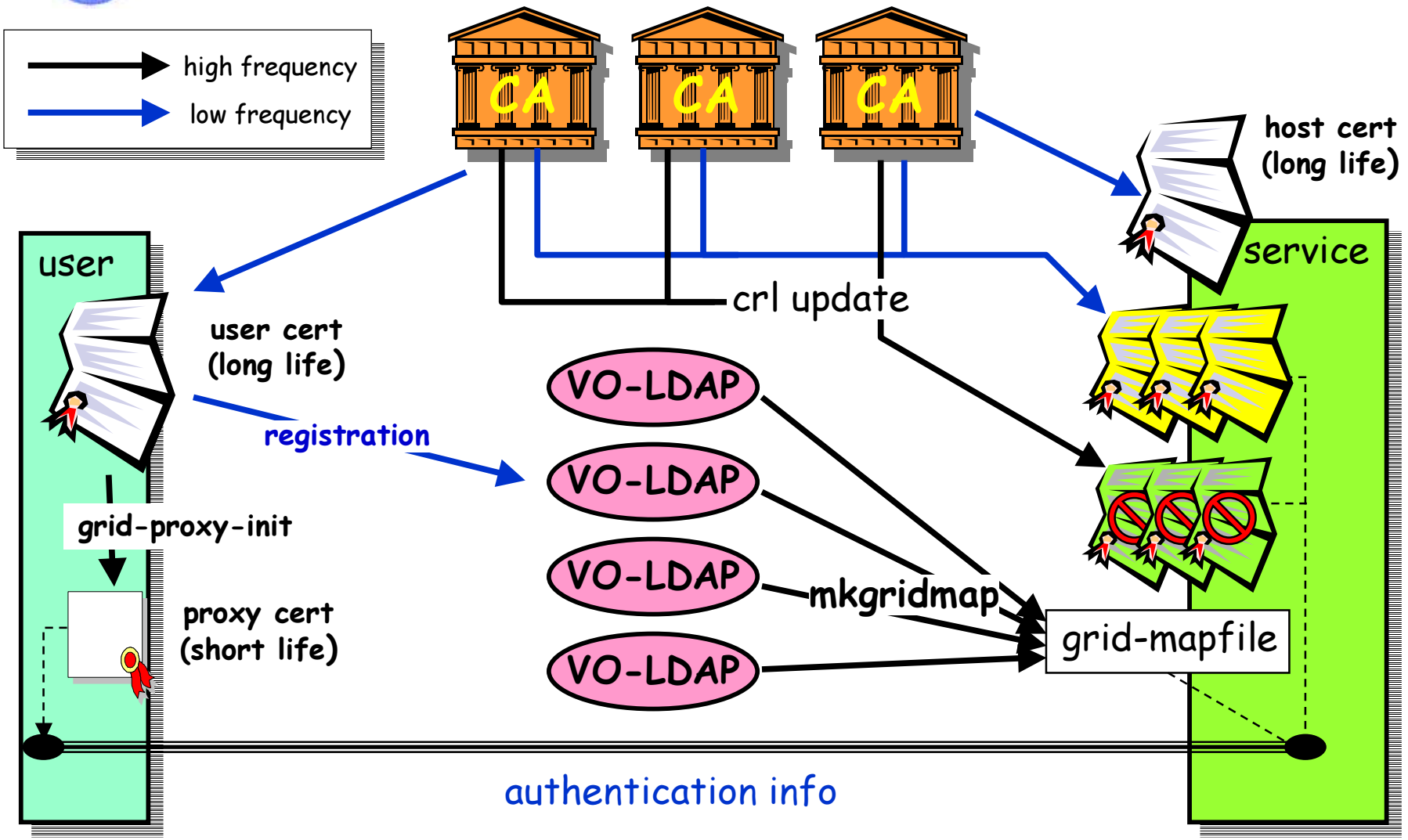
VI DataGRID Conference

Barcelone, 12-15 May 2003

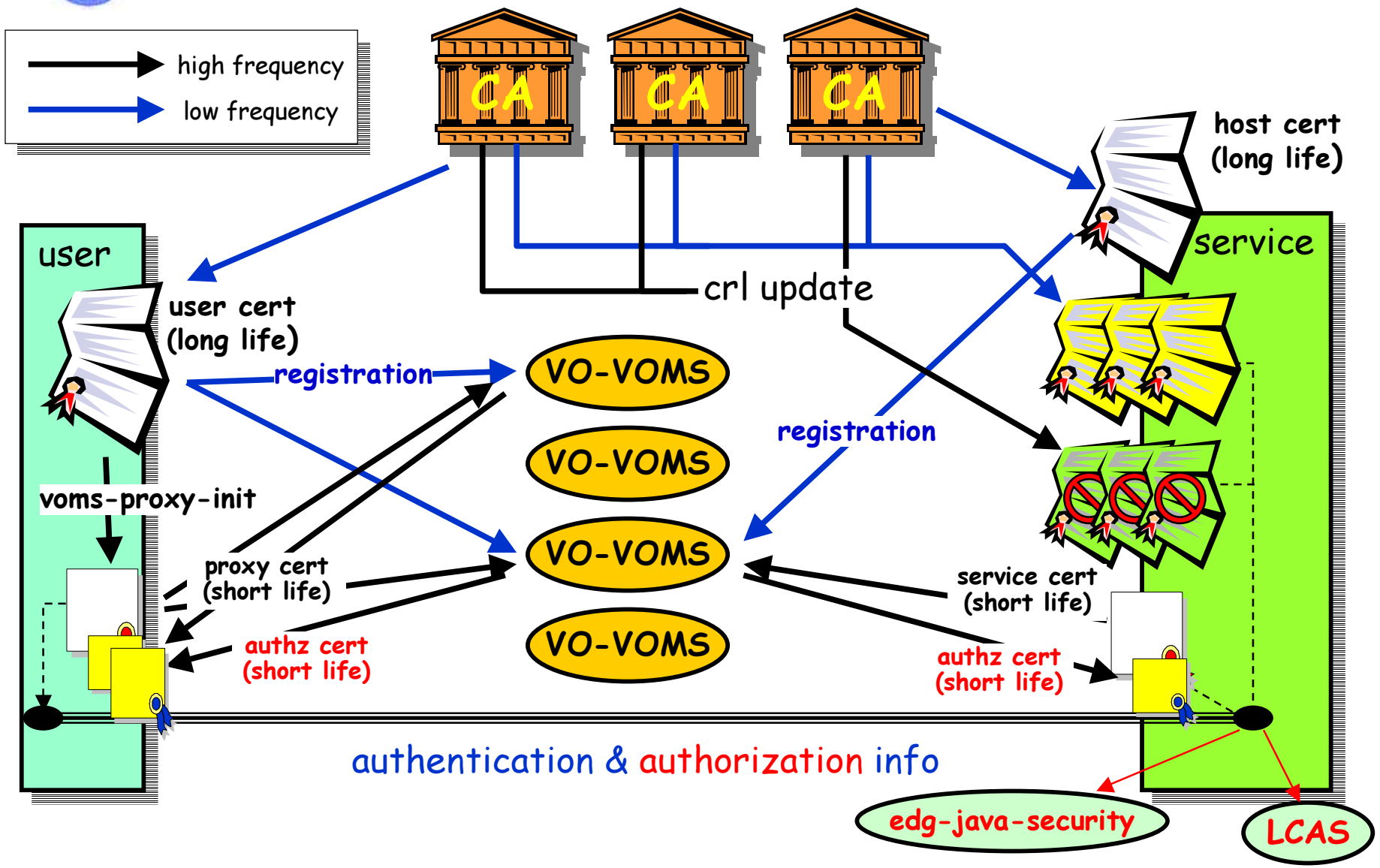
Overview


focus is on VOMS
details are in D7.6
Security Design
CA
request
certificate:
dn, ca, Pkey
proxy cert:
dn, cert, Pkey,
VOMS cred.
(short lifetime)
VOMS
certificate
VOMS cred:
VO, group(s),
role(s)
user
delegation:
cert+key
(long lifetime)
MyProxy
re-newal
request
delegation:
cert+key
(short lifetime)
proxy cert
proxy cert
proxy cert
proxy cert
proxy cert
auth
TrustManager
authz
WebServices Authz
dn,attrs,acl, req.op
->yes/no
map
dn -> DB role
doit
auth
TrustManager
pre-process:
parameters->
obj.id + req. op.
authz
obj.id -> acl
dn,attrs,acl, req.op
->yes/no
doit
auth
mod_ssl
pre-process:
parameters->
obj.id + req. op.
authz
GACL:
obj.id -> acl
dn,attrs,acl, req.op
->yes/no
doit
auth
GSI
pre-process:
parameters->
obj.id + req. op.
authz
GACL:
obj.id -> acl
dn,attrs,acl, req.op
->yes/no
doit
auth
GSI
authz
LCAS
dn,attrs,acl, req.op
->yes/no
map
LCMAPS
dn -> userid, krb ticket
doit
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)
Java
web
C

User's Authorization in EDG 1.4.x



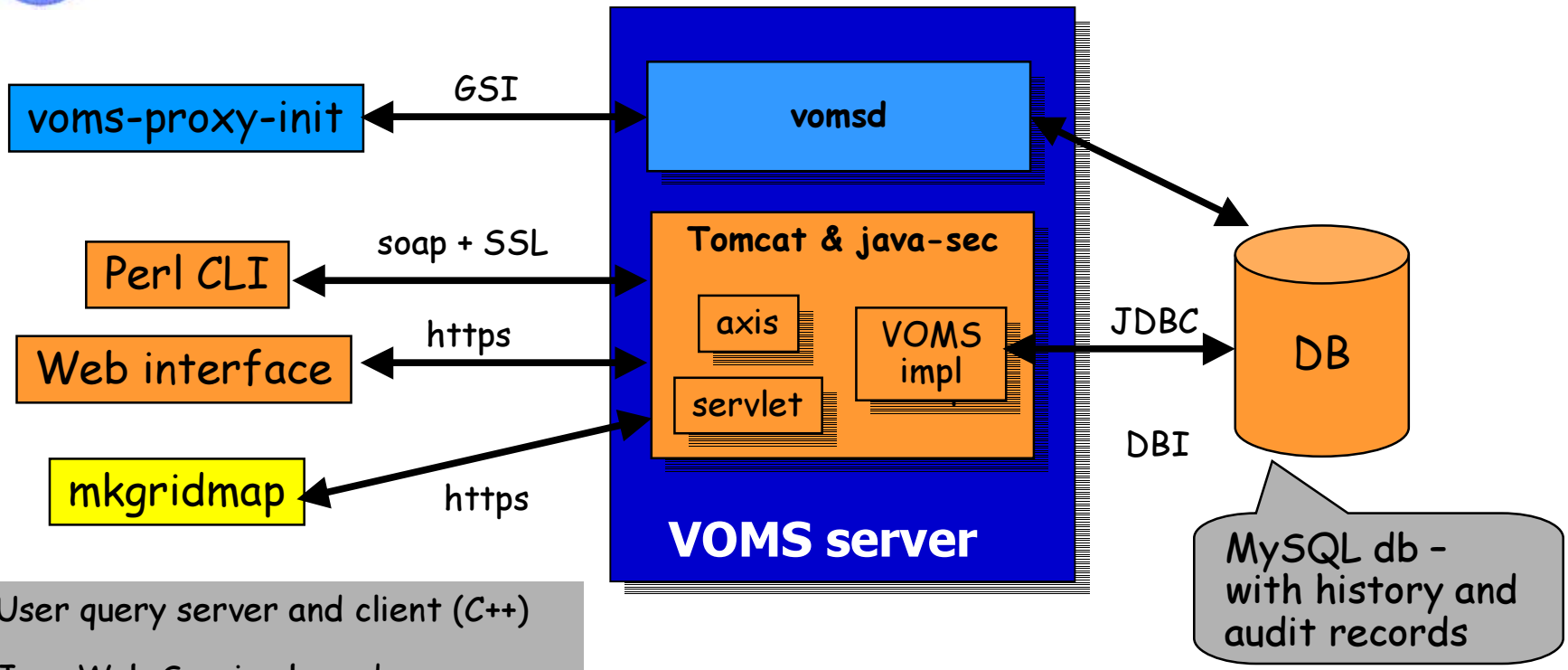
User's Authorization in EDG 2.x



VOMS Overview

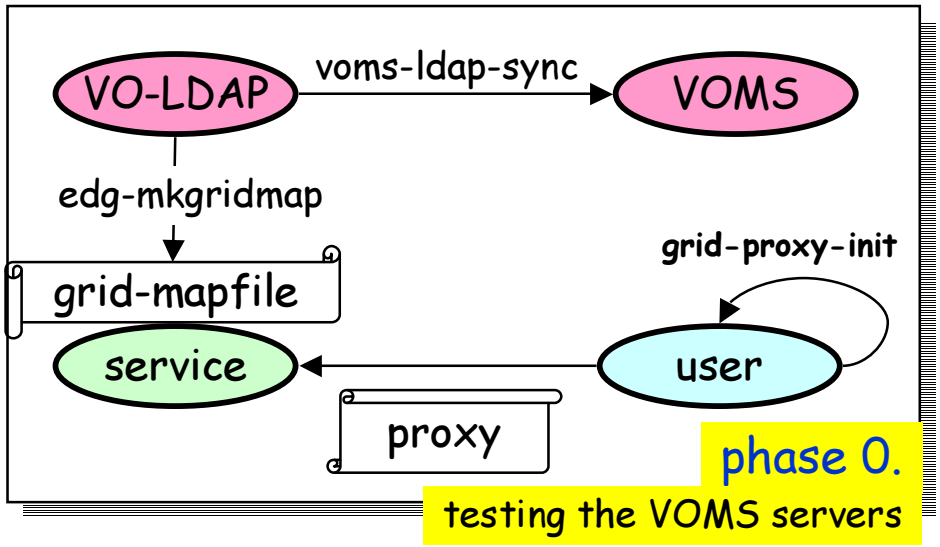
- Provides info about the user's relationship with his VO('s)
  - groups, roles (admin, student, ...), capabilities (free form string), temporal bounds
- Features
  - **single login:** *voms-proxy-init* only at the beginning of the session (replaces *grid-proxy-init*);
  - **expiration time:** the authorization information is only valid for a limited period of time (possibly different from the proxy certificate itself);
  - **backward compatibility**: the extra VO related information is in the user's proxy certificate, which can be still used with non VOMS-aware services;
  - **multiple VO's:** the user may authenticate himself with multiple VO's and create an aggregate proxy certificate;
  - **security**: all client-server communications are secured and authenticated.

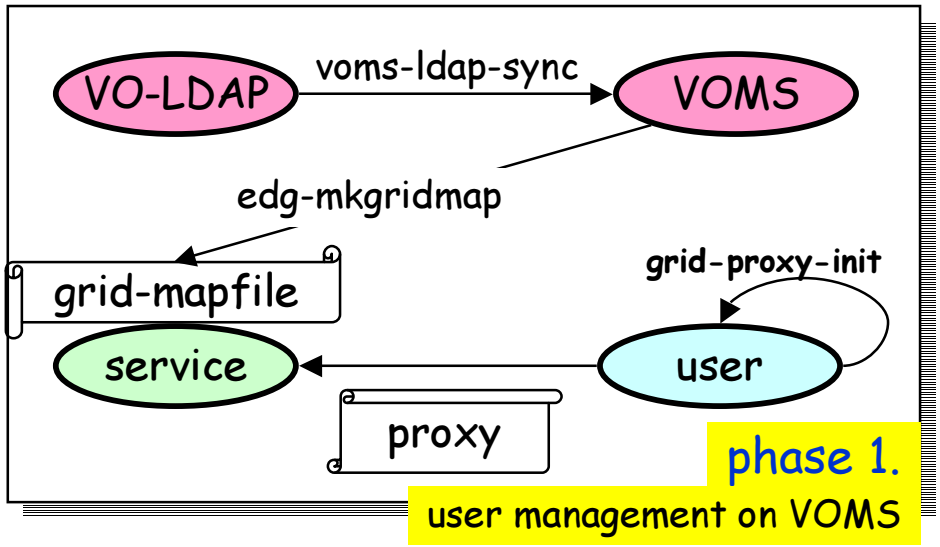
VOMS Architecture

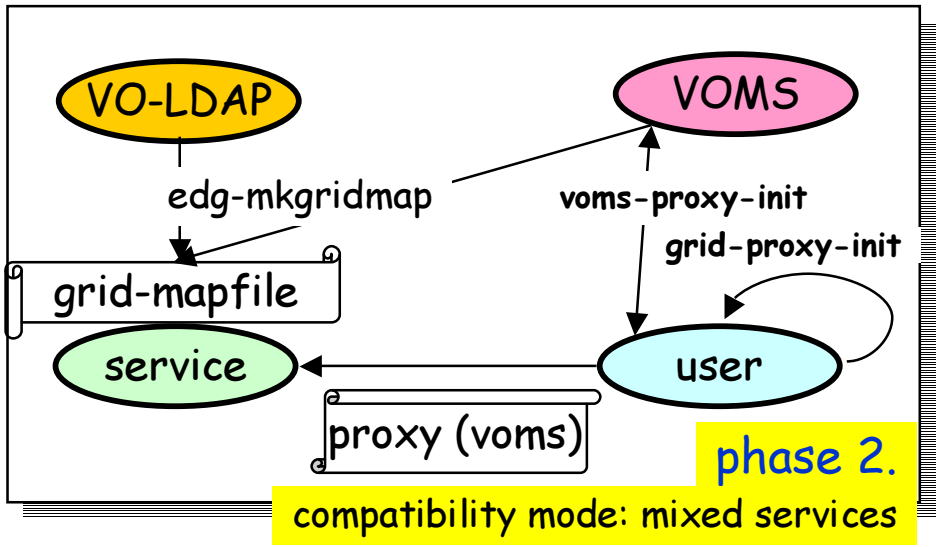


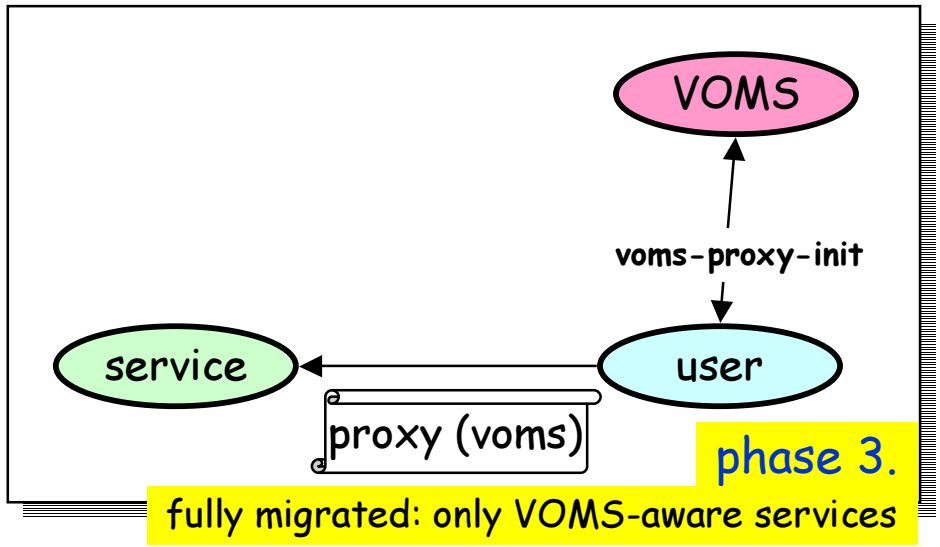
- User query server and client (C++)
- Java Web Service based administration interface
  - Perl client (batch processing)
  - Web browser client (generic administrative tasks)
- Web server interface for mkgridmap

Migration to VOMS









Auth/Authz in Services

- 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
  - (Slahgrid for transparent filesystem ACLs)

Local Site Authorization

- 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*)
    - 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

edg-java-security

- Trust manager
  - GSI compatible authentication
  - Adapters to HTTP and SOAP
  - Currently deployed for Tomcat4
- 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

TODO

- Test the pieces in the Testbeds
- Implement the missing pieces and Discarding the unused
- Common syntax and semantics for access control configurations
- Substitution of VOMS certificates by Attribute Certificates (RFC3281)
- Support for time cyclic/bound permissions and roles
- Database replication

- Use the security model -> get real life use cases