Overview

- Trust Manager
- Authorization
- Life as a separate module
- VOMS

The things, which are done and plans.

Trust Manager

- Using /etc/grid-security/certificates -> should be default
- CRL handling -> should be /etc/grid-security/certificates
- Using service certificate – default location?
  /opt/edg/<service>.pem|key
- Configuration in Java classes -> should be XSLT
- Generalization for every SSL socket
  both on server and client side
- Moved to BouncyCastle (no CoG dependency)
- Plans on full certificate chain checking by hacking the BouncyCastle trustmanager

Authorization

- Server side mappings
  - File mapper (e.g. gridmap-file)
  - XML configuration file
  - Mapping by regular expressions
  - Mapping in a DB table
  - TODO: catching VOMS roles/groups
- Client side: gathering the information
  - Put role attribute into SOAP header (James)
  - Put roles in an attribute certificate (VOMS – tbl.)

How-to/doc for service developers (Wiki-HIP)
Tomcat4/Connector configuration

Life as a separate module

- Renaming misery… but hopefully finished
- Updated to the common “tools” build system
- Questions:
  - client/server/service specific jar files?
    Where to put the glue code?
  - How many RPMs we need … if we need at all?
  - Configuration: server.xml from scratch or configure script?

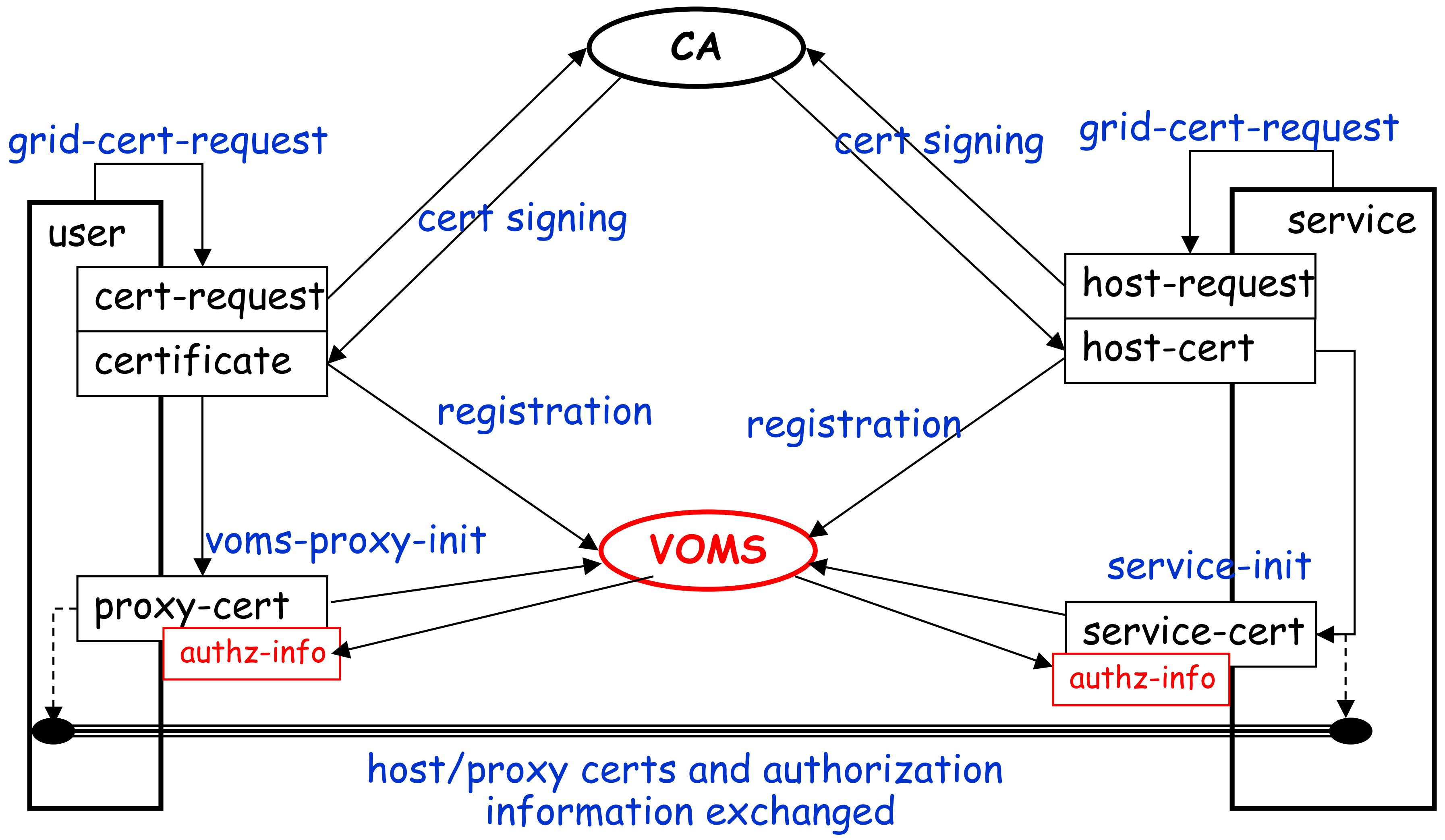
VOMS

- Big news: merge with WP6 efforts
  common point: the MySQL database
- Our part: Java
  - Administrative interfaces
  - Authorization plugin for Java services
- WP6 part: C
  - voms-proxy-init and coresponding service in C
  - Authorization plugin for C service = LCAS
- Test/Development solution: emulate the VO/role attribute passing through SOAP headers – not secure, but will look like the same for the services

VOMS idea

1. User asks VOMS for attributs (VO/group/role)
   voms-proxy-init –vo CMS –role admin
2. VOMS looks up the attributes and packs it into an Attribute Certificate (AC): {user, voms, attributes, voms-signature}
3. User generates a proxy certificate and puts the AC as an optional extension into this
4. User contacts a service with this prepared proxy
5. Service authenticates the user in the "old way"
6. Service extracts the AC from the proxy -> roles
7. Service does its authorization based on the roles

VOMS big-picture


CA
grid-cert-request
cert signing
cert signing
grid-cert-request
user
cert-request
certificate
service
host-request
host-cert
registration
registration
voms-proxy-init
VOMS
service-init
proxy-cert
authz-info
service-cert
authz-info
host/proxy certs and authorization information exchanged