VOMS and LCMAPS
on Global Permissions and Local Credentials

David Groep & Gridification Team

partly based on CHEP2003 talk by Luca dell'Agnello et al.
(SCG, WP4, WP6)

davidg@nikhef.nl

http://hep-project-grid-scg.web.cern.ch/

Talk Outline

- Introduction
- Authorization requirements
- VO Membership Service
- Spitfire TrustManager

- Local site enforcement mechanisms (LCAS, LCMAPS)
  - LCMAPS architecture
  - Evolution Manager and the Policy Language
  - Credential Enforcement Gotchas
- Conclusions

Introduction (1)

- **EDG security infrastructure based on X.509 certificates (PKI)**
- **Authentication**
  - 16 national certification authorities
  - Policies and procedures → mutual thrust
  - Users identified by certificates signed by their national CA
- **Authorization**
  - Cannot decide Authorization for grid users only on local site basis
  - At least 2 entities involved
    - Resource Providers (e.g. Tiers in LCG framework)
    - Virtual Organizations (e.g. LHC experiments collaborations)

Introduction (2)

Authorization (cont.)

- Resource granting established by agreements VO's - RP's.
  - VO's administer user membership, roles and capabilities
  - RP's evaluate authorization granted by VO to a user and map into local credentials to access resources
    - Trust/Authorization Manager for Java (e.g. Spitfire)
    - LCAS/LCMAPS for farms
    - SlashGrid for storage (Andrew's talk)
- Need tool to manage membership for large VO's (10000 users)
  - Globus mechanism (grid-mapfile) not scalable
- VO membership service (VOMS)
  - Extends existing grid security infrastructure architecture with embedded VO affiliation assertions
  - Permits authorization control on grid services for job submission, file and database access.

Authorization requirements

- **Architecture**
  - centralized and scalable (for an Auth policy VO based)
- **Attributes support**
  - group membership (subgroup, *multiple inheritance*, ..)
  - Roles (admin, student, ..), capabilities (free form string), ..
  - Temporal bounds
- **Resource Provider**
  - keep full control on access rights
  - traceability user level (not VO level)
- **Security issues**
  - Auth Server must not be a Single point of failure
  - Auth communications must be trusted, secured and reserved

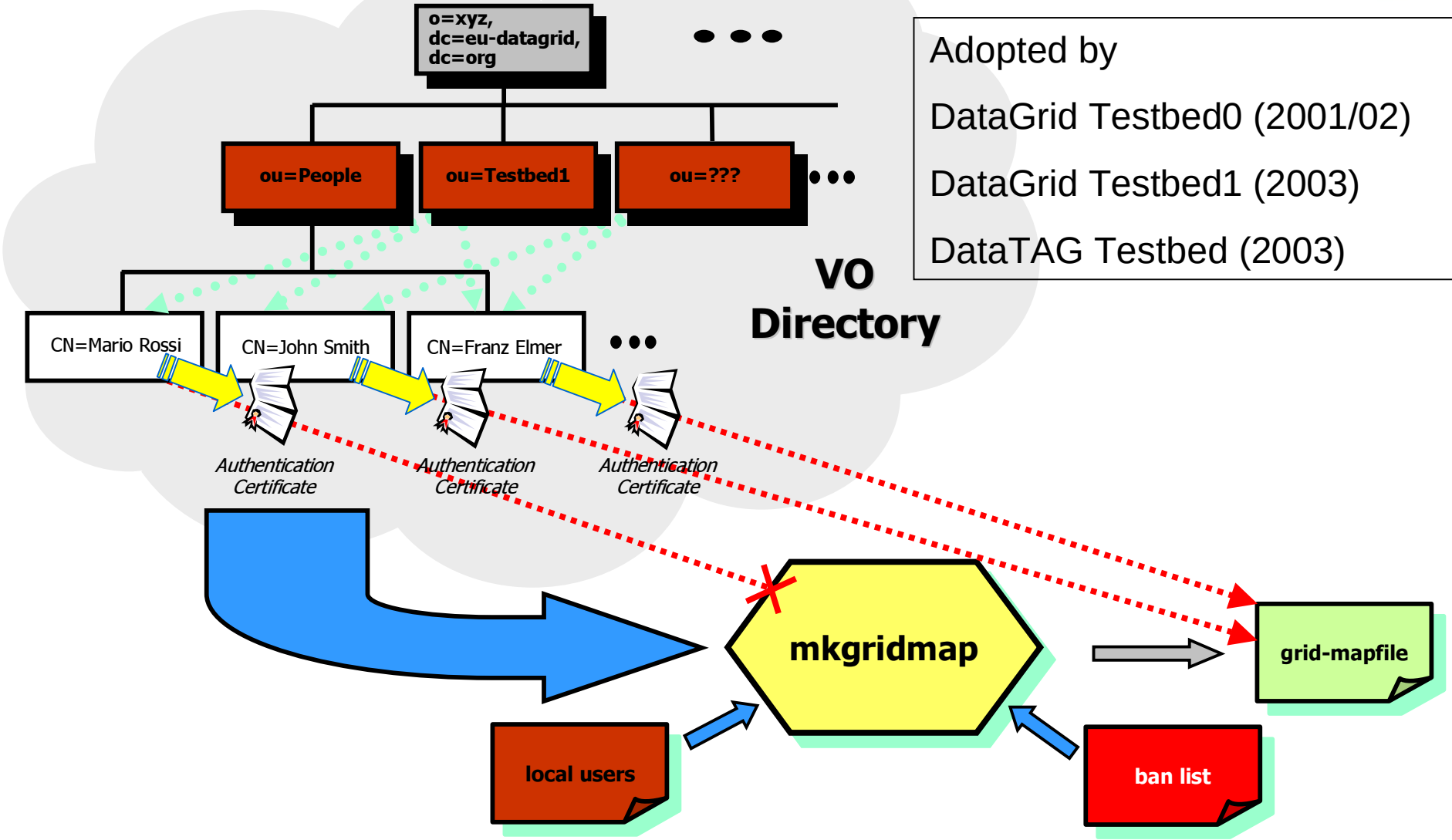
Globus Authorization Mechanism

- **grid-mapfile**
  - Grid credentials (user's Certificate) to local credentials (unix account) mapping
  - "Boolean" authorization
  - Information provided via VO-LDAP servers
  - Managed "manually" by the resource admin (via mkgridmap)

```
"/C=IT/O=INFN/L=Parma/CN=Roberto Alfieri/Email=roberto.alfieri@pr.infn.it" alfieri
"/C=IT/O=INFN/L=Parma/CN=Fabio Spataro/Email=fabio.spataro@pr.infn.it" spataro
```

- **No centralization**
- **No scalability**
- **Lack of flexibility**

VO-LDAP Architecture



Adopted by

DataGrid Testbed0 (2001/02)

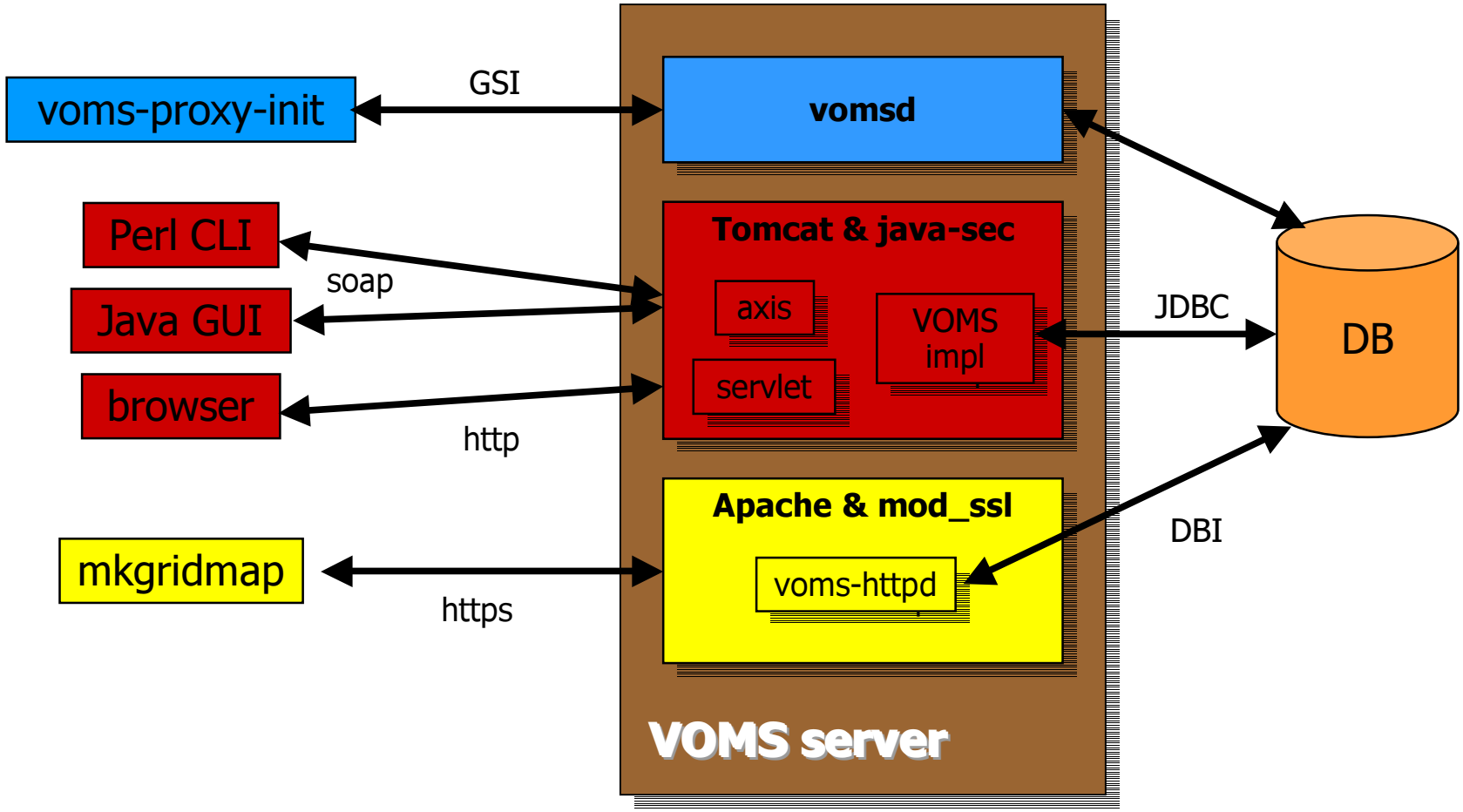
DataGrid Testbed1 (2003)

DataTAG Testbed (2003)

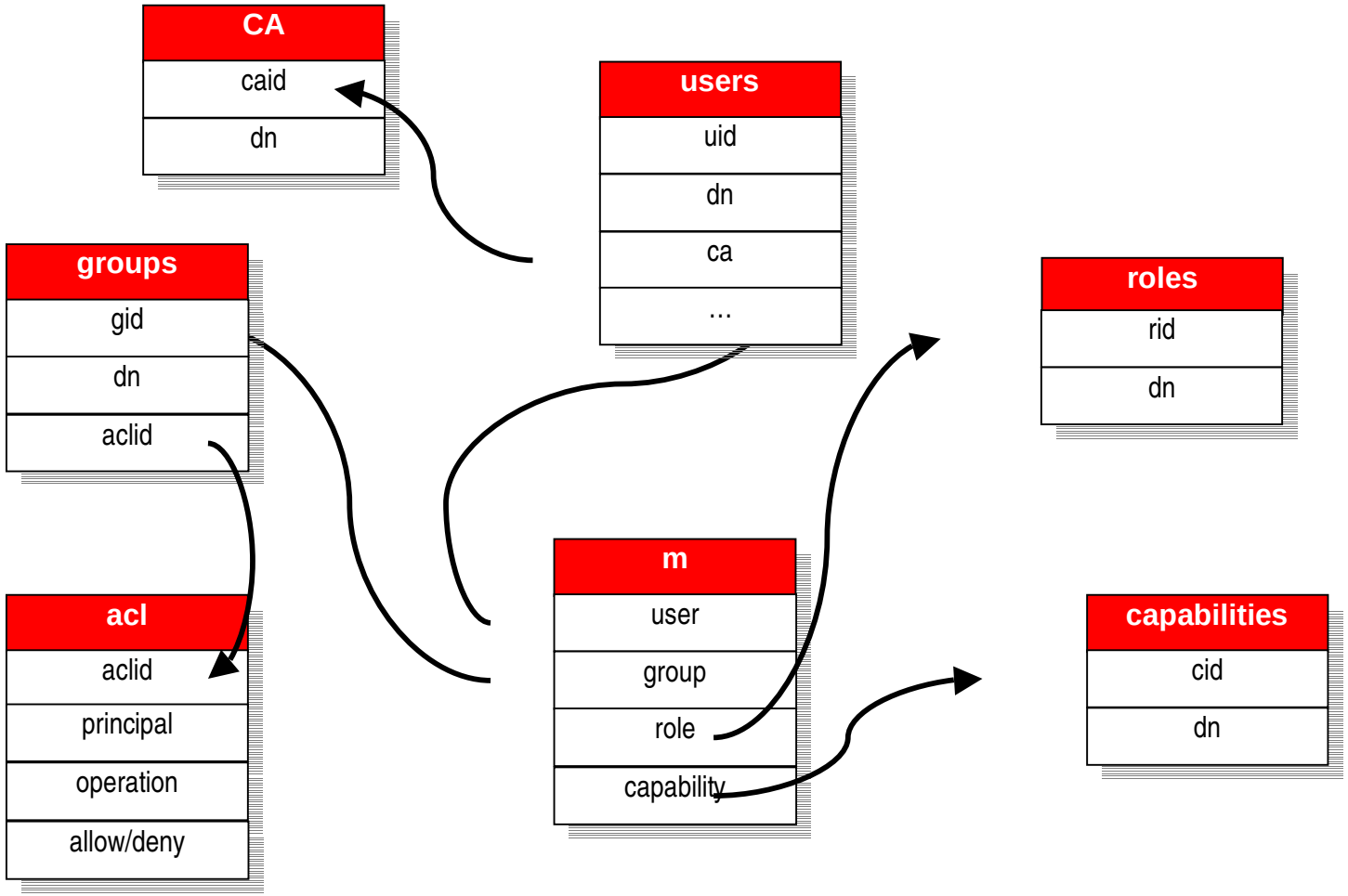
The Virtual Organization Membership Service

- **The Virtual Organization Membership Service (VOMS)**
  - Developed by European Datagrid and Datatag collaborations to solve current LDAP VO servers limitations
  - Grants authorization data to users at VO level
    - Each VO has its own VOMS
    - Support for group membership (subgroup, *multiple inheritance*, ..), "forced" groups (i.e. for negative permissions), roles (admin, student, ..) and capabilities (free form string)
  - Essentially a front-end to an RDBMS
    - User client – queries the server for authorization info
    - User server – returns authorization info to the client
    - administration client – used by VO administrators for management
    - administration server – executes client update operations on db
    - transition tool – interface to mkgridmap++ (see below)
  - All client-server communications are secured and authenticated
  - Authorization info is processed by the gatekeeper
    - full functionality of VOMS achieved via LCAS/LCMAPS plug-ins (see below)

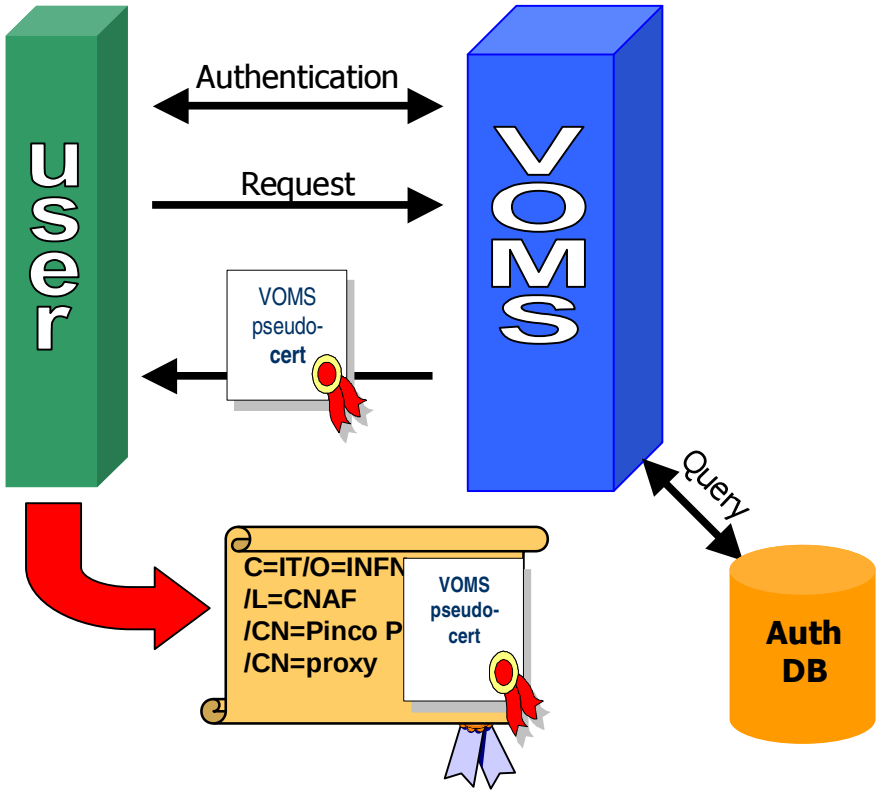
VOMS overview



DB Structure (simplified)



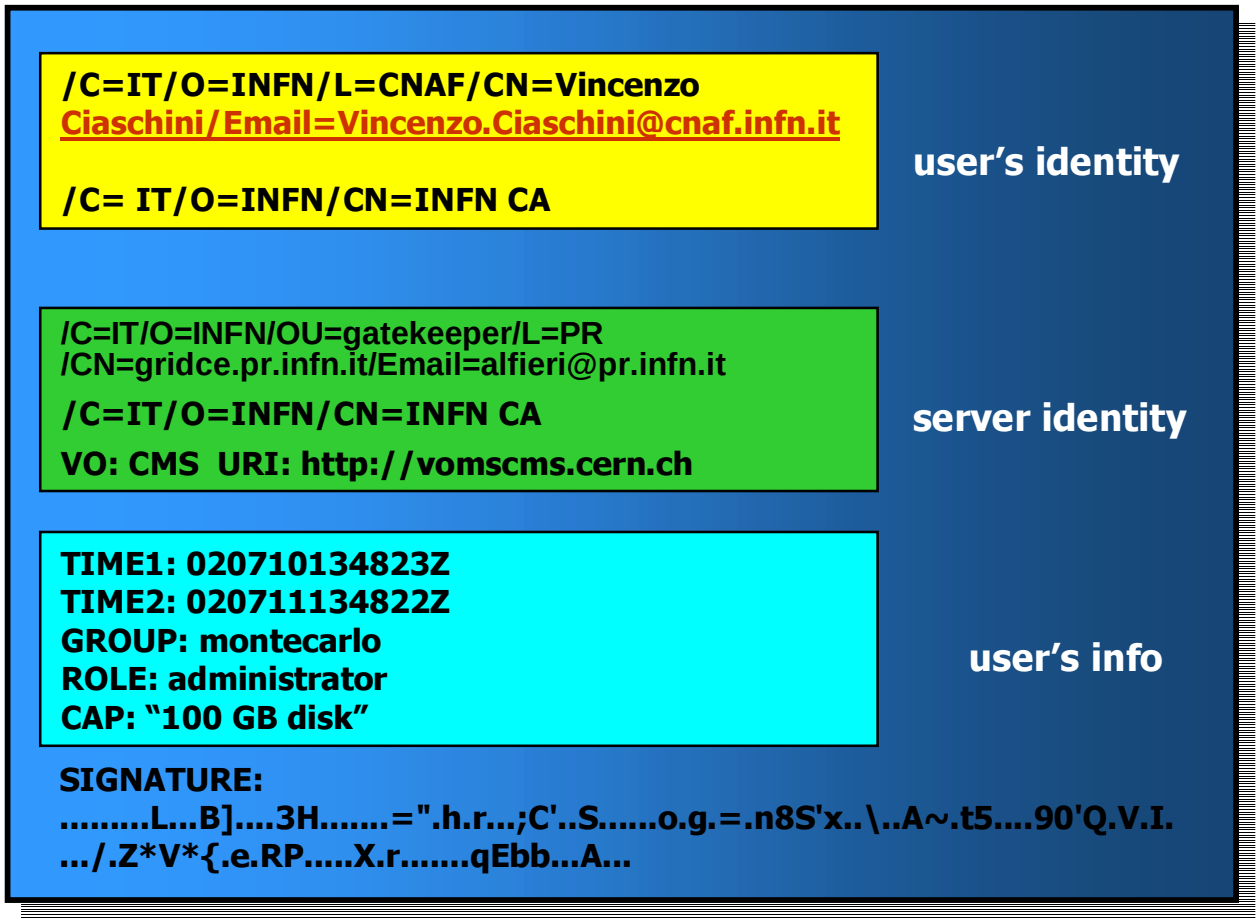
VOMS Operations



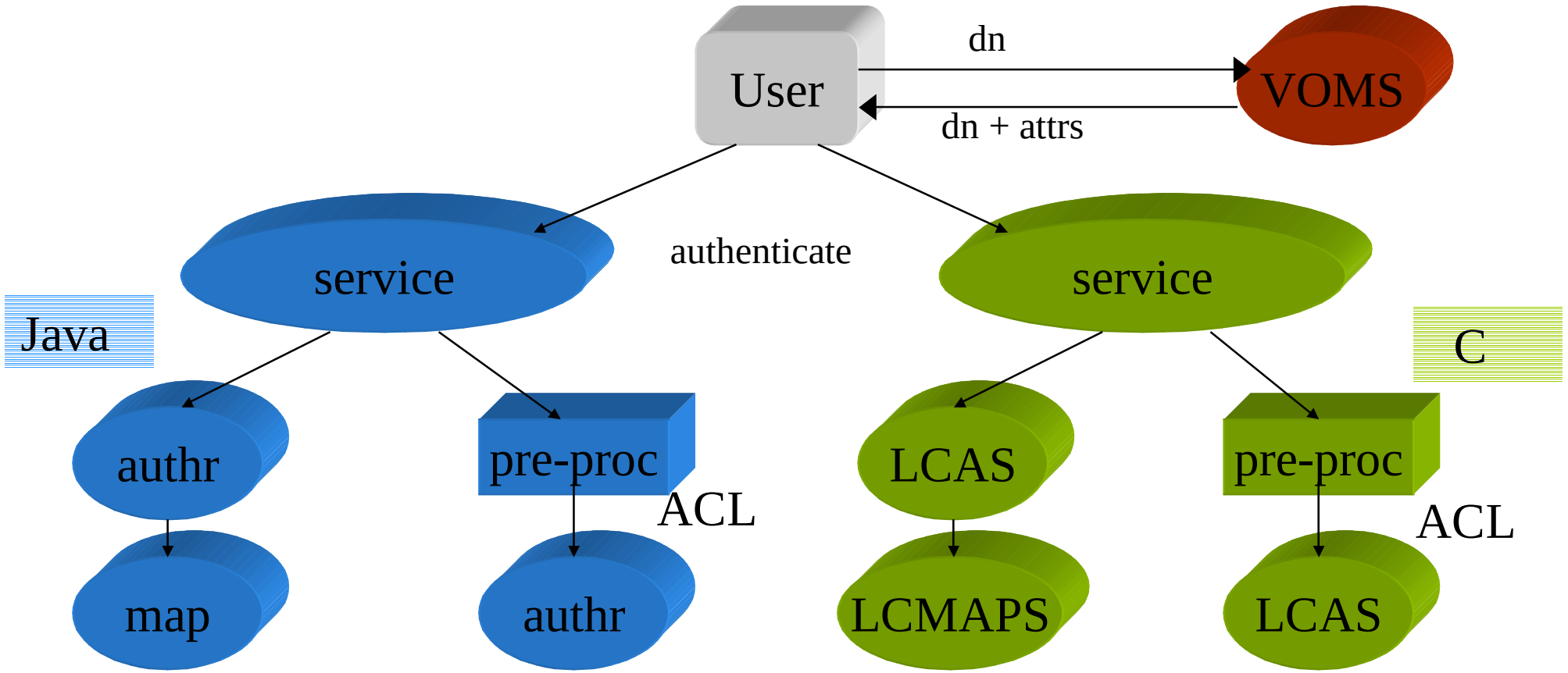
1. Mutual authentication Client-Server
   - Secure communication channel via standard Globus API
2. Client sends request to Server
3. Server checks correctness of request
4. Server sends back the required info (signed by itself) in a "Pseudo-Certificate"
5. Client checks the validity of the info received
6. Client repeats process for other VOMS's
7. Client creates proxy certificates containing all the info received into a (non critical) extension
8. Client may add user-supplied auth. info (kerberos tickets, etc…)

Pseudo-Certificate Format

- The pseudo-cert is inserted in a non-critical extension of the user's proxy
  - 1.3.6.1.4.1.8005.100.100.1
- It will become an Attribute Certificate
- One for each VOMS Server contacted



Authorization



Coarse-grained
e.g. Spitfire

Fine-grained
e.g. RepMeC

Coarse-grained
e.g. CE, Gatekeeper

Fine-grained
e.g. SE, /grid

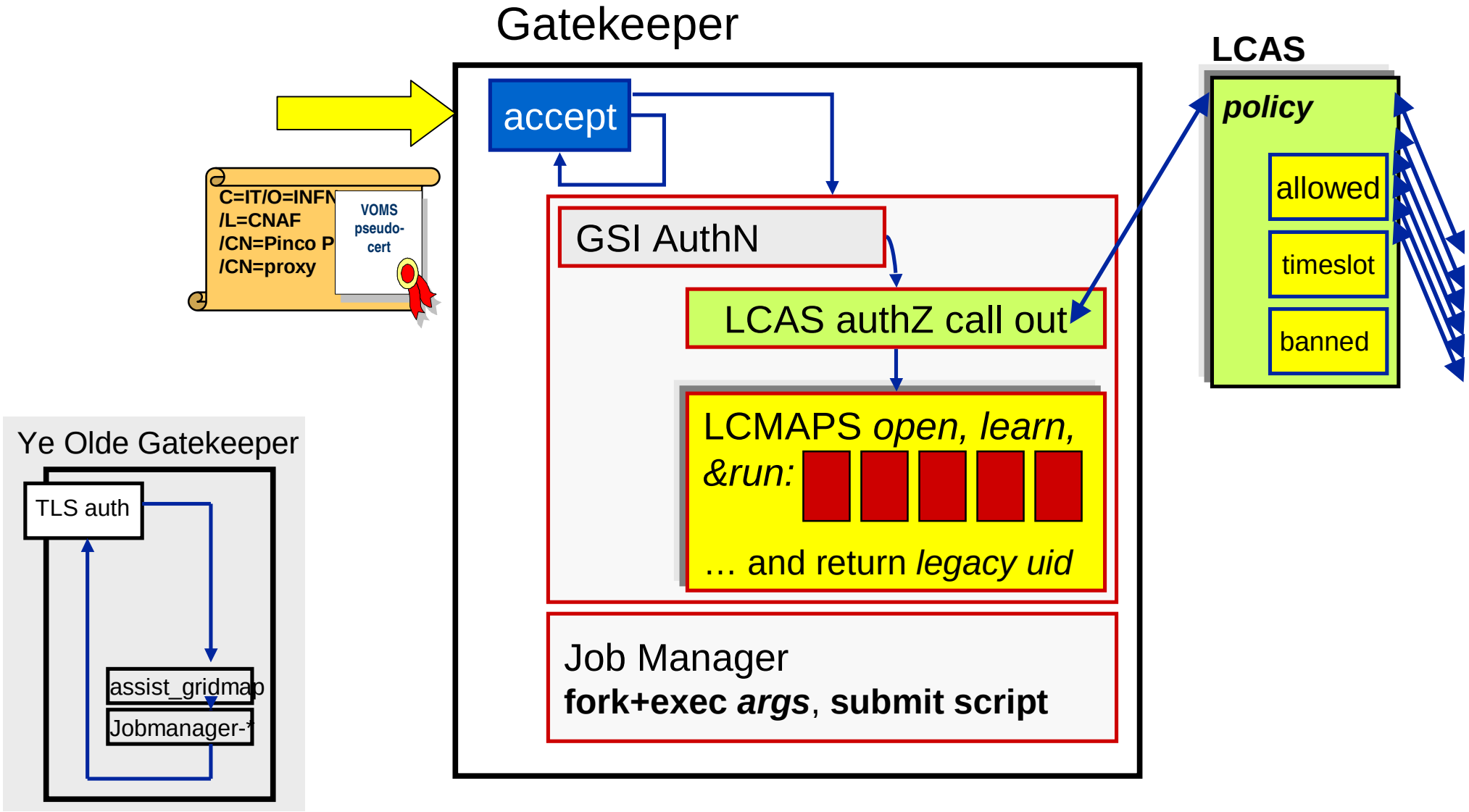
Spitfire

- **Provides uniform access to various implementations of database back ends via a grid-enabled front end**
  - SOAP interface
  - JDBC interface to RDBMS
- **TrustManager: certificate validator for Java services**
  - Permits (mutual) secure client-server authentication
  - Supports X509 certificates and CRL's
- **Support for connections via HTTP(S) using GSI certificate for authentication**
- **Role-based authorization**
  - Support for Authorization info provided by VOMS

Local Site Authorization Services

- **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 plug-ins)
    - 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
  - *Plug-in* framework, driven by comprehensive *policy language*
  - Mapping based on user identity, VO affiliation, site-local policy
  - Supports standard UNIX credentials (incl. pool accounts), AFS tokens, Krb5

EDG Gatekeeper (release 2.1)



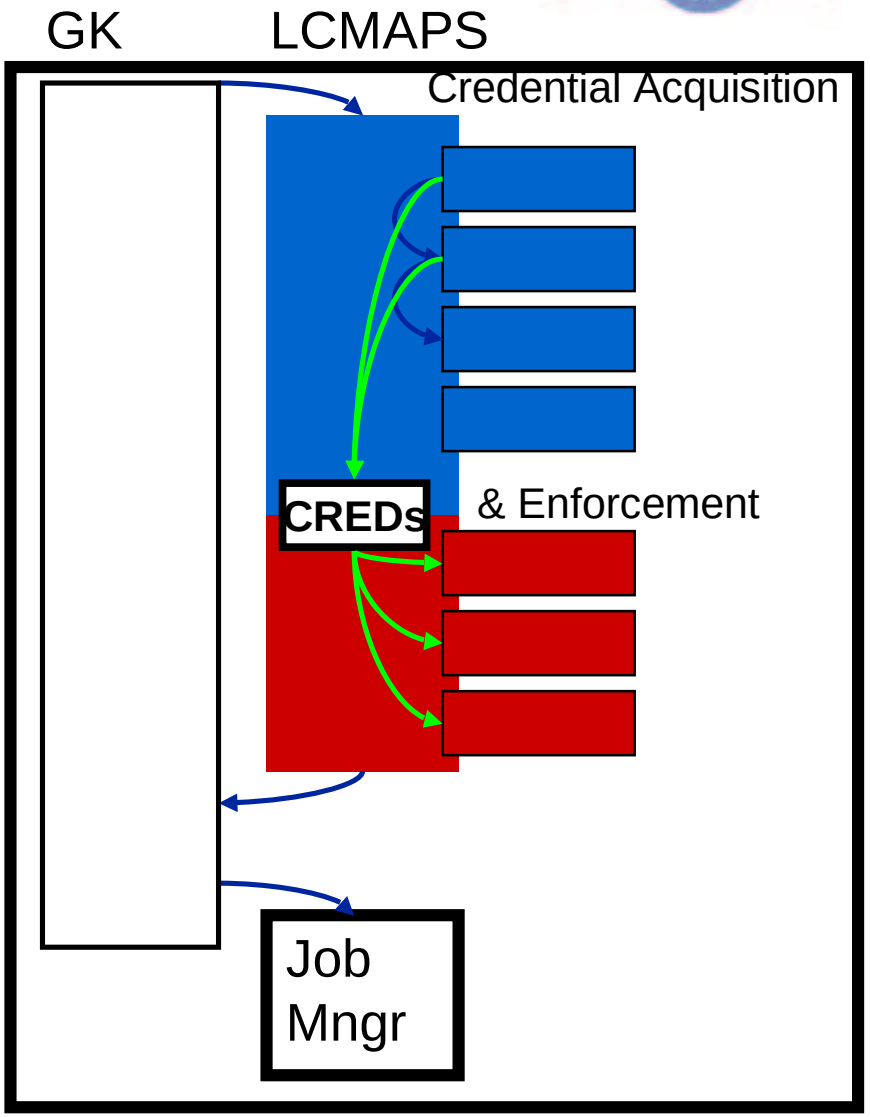
LCMAPS – requirements

- Backward compatible with existing systems (grid-mapfile, k5cert)
- Support for multiple VOs per user (and thus multiple UNIX groups)
- Mimimum system administration
  - Poolaccounts
  - Pool”groups”
  - Understandable configuration
- Extendible

- Boundary conditions
  - Has to run in privileged mode
  - Has to run in process space of incoming connection (for *fork* jobs)

LCMAPS – control flow

- User authenticates using (VOMS) proxy
- LCMAPS *library* invoked
  - Acquire all relevant credentials
  - Enforce "external" credentials
  - Enforce credentials on current process tree at the end
- Run job manager
  - Fork will be OK by default
  - Batch systems may need primary group explicitly
  - Batch systems will need updated (distributed) UNIX account info
- Order and function: policy-based



LCMAPS – plugin introspect

- Framework is “resistent” to new module functionality and v.v.
- Invocation and arguments list for modules discovered via the ”introspection API”
  - Information in (VOMS) proxy cert access by symbolic names
  - Argument description by name, type, range, modifiability
  - Credential acquisition in named and typed lists
- Various modules can support different interfaces
- Modules from multiple generation can be “mixed”
- An “old” framework will work with “bleeding-edge” modules
- See apidoc for more details…

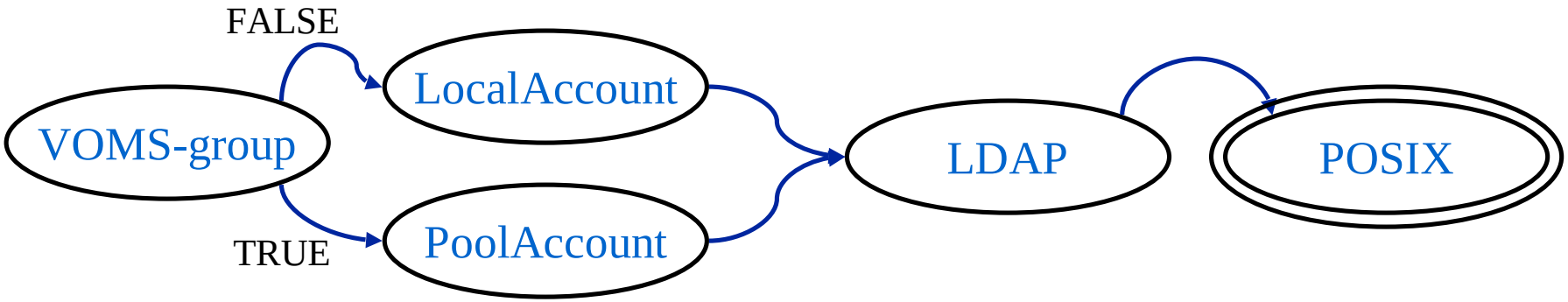
LCMAPS – modules

- Modules represent *atomic* functionality

- **VOMS** from role info and local mapfile assign gid (A)
- **PoolAccounts** from username assign unique uid (A)
- **PoolGroups** from (VOMS) groupname assign unique gid (A)
- **LocalAccount** from username assign local existing unique uid (A)
- **AFS/Krb5** get token based on user DN info (A)
- **POSIX *process*** setuid() and setegid() (E)
- **POSIX LDAP** update distributed user database (E)
- **Krb5** run job via k5cert (E)
- **…**

LCMAPS – policy evaluation

- State machine approach (superset of boolean expressions)

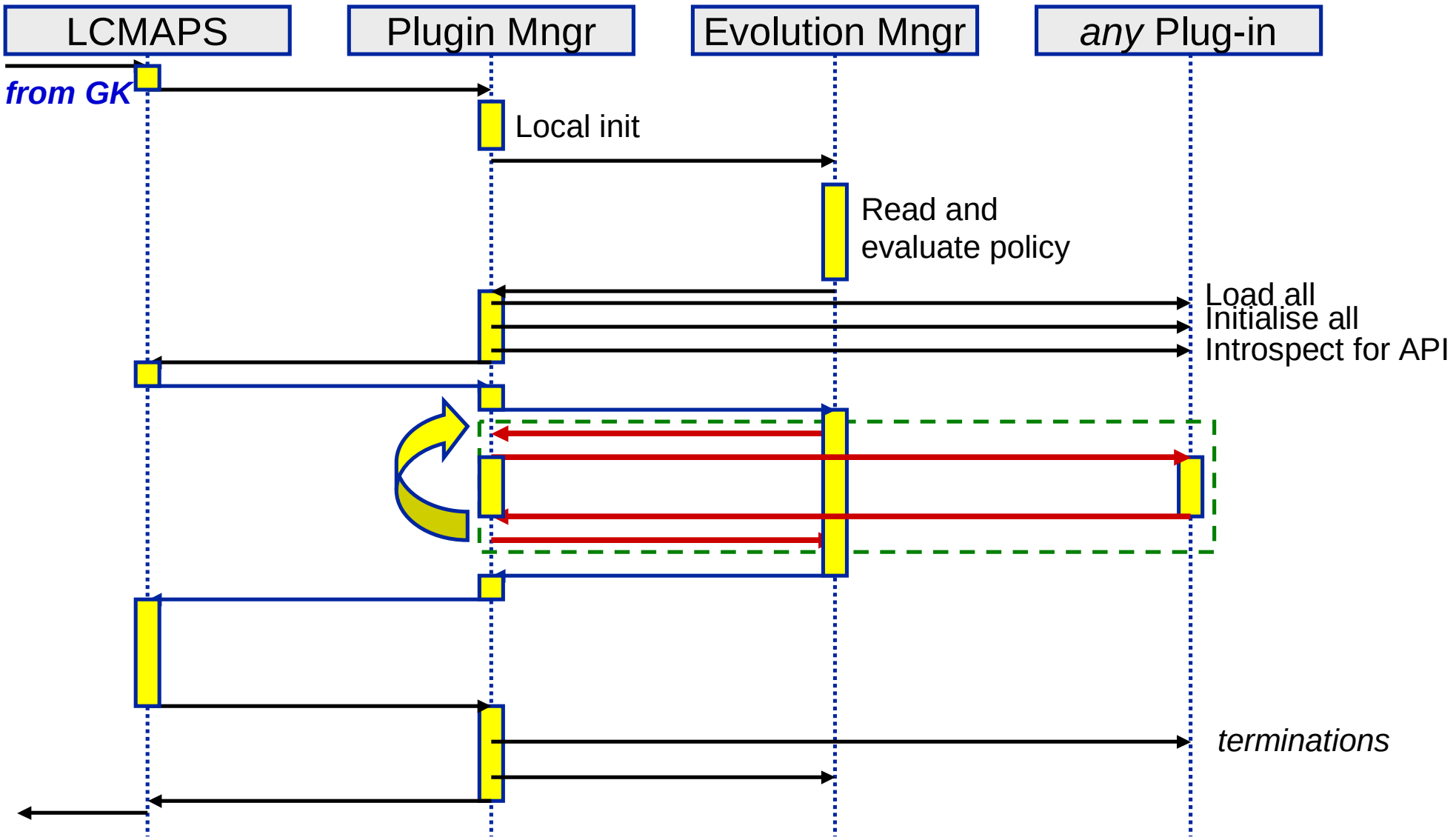


- Policy description file:

```
path = /opt/edg/lib/lcmaps/modules

localaccount ="lcmaps_localaccount.mod \
                    -gridmapfile /etc/grid-security/grid-mapfile"
poolaccount = "lcmaps_poolaccount.mod -gridmapfile /etc/grid-security/grid-mapfile"
posix_enf =   "lcmaps_posix.mod -maxuid 1 -maxpgid 1 -maxsgid 32"
voms =        "lcmaps_voms.mod -vomsdir /etc/grid-security/certificates \
                    -certdir /etc/grid-security/certificates"
standard:
voms -> poolaccount | localaccount
localaccount -> posix_enf
poolaccount -> posix_enf
```

LCMAPS – invocation and running



LCMAPS – enabling new functionality

- Local UNIX groups based on VOMS group membership and roles
- More than one VO/group per grid user
- No pre-allocation of pool accounts to specific groups
- New mechanisms:
  - groups-on-demand
  - Central user directories (nss_LDAP, pam-ldap)
- Why do we (still) need LCAS:
  - Centralized decisions on authorized users (like at FNAL)
  - Coordinated access control across multiple CEs
  - (and save on expensive account allocation mechanisms in LCMAPS)

Status and Future Works

LCAS was in release 1.4.x and is currently used

VOMS release delayed till after 2.0.0
Unit deployment **VOMS** (Client/server, Admin, mkgridmap++) in Feb. '03

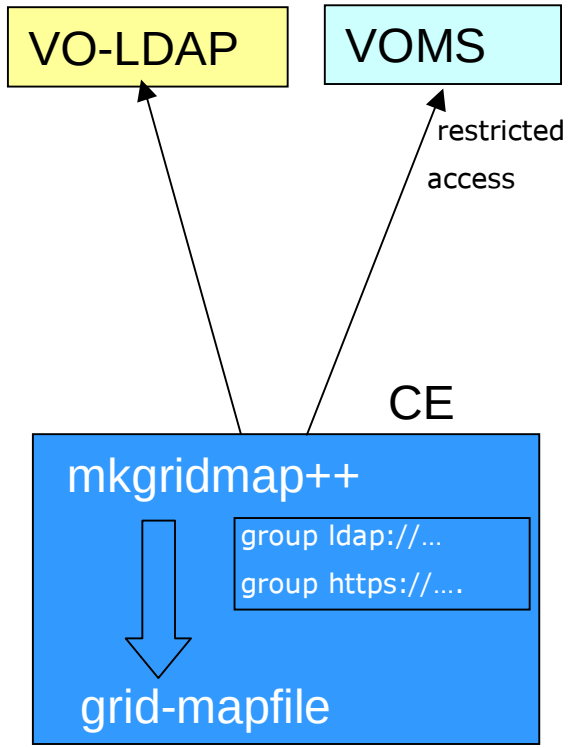
LCMAPS release foreseen for $DATE (see status talk ☺)

Work in progress

- **VOMS**
  - Certificates will be substituted by true Attribute Certificates (RFC3281)
  - Support for time cyclic/bound permissions and roles
  - Database Replication
- **LCAS/LCMAPS**
  - Framework ready, evolution manager ready, doc & apidoc available
  - Completed plug-ins: localaccount, poolaccount, POSIX
  - In development (various stages): VOMS, AFS/Krb5, PoolGroups, LDAP

mkgridmap++

- **Need for a tool for the transition to LCAS/LCMAPS mechanism**
- **VOMS and VO-LDAP can and MUST coexist**
  - VOMS can also be used for grid-mapfile generation.
  - New directive in the config file
- **New feature**
  - Authenticated access to VOMS (*not LDAP*) servers based on https protocol to restrict the clients allowed to download the list of the VO members



More Informations

EDG Security Coordination Group

Web site **http://hep-project-gris-scg.web.cern.ch/**

VOMS

Web site **http://grid-auth.infn.it/**

CVS site **http://cvs.infn.it/cgi-bin/cvsweb.cgi/Auth/**

Developers' mailing list **sec-grid@infn.it**

LCAS-LCMAPS

Web site **http://www.dutchgrid.nl/DataGrid/wp4/**

CVS site **http://datagrid.in2p3.fr/cgi-bin/cvsweb.cgi/fabric_mgt/gridification/lcas/**

http://datagrid.in2p3.fr/cgi-bin/cvsweb.cgi/fabric_mgt/gridification/lcmaps/

Maillist **hep-proj-grid-fabric-gridify@cern.ch**

Spitfire

Web site **http://spitfire.web.cern.ch/Spitfire/**

Related Works

- **CAS (Globus Team)**
  - Proxy generated by CAS server, not by user (difficult traceability)
  - Proxy not backward compatible
  - Attributes are permissions (resources access controlled by VO)

- **Permis (Salford Univ., England)**
  - AC's stored in a repository at the local site
  - Good policy engine
  - VOMS complementary (flexible VOMS AC + PERMIS pol. engine)

- **Akenti (US Gov.)**
  - Target Web sites, not easy migration in a VO environment