Site access control issues
(a sneak preview of DJRA3.2)

Martijn Steenbakkers for JRA3
Universiteit van Amsterdam and NIKHEF

Outline

- Goals of the “Site access control architecture”
- What do (or should) we use today?
- What we would like to see next
- Status and future of LCAS
- Status and future of LCMAPS
  - Integration with Dynamic Account Service (DAS)
- Timeline

Goals

- **Generic access control to services at site level**
  - Authentication
  - Authorization
  - Sandboxing & legacy applications
- **Sites are in control of their resources**
- **Flexibility, scalability**
- **Centralized control**
- **Converge to one policy format**
- **Requirements from site AAA RG (incorporated in MJRA3.1 “user requirements”**
- **Requires input from MWSG, JSPG and ROC managers**

What should/could we use today?

- **Authentication: acquire ID + assertions**
  - X.509 and attribute certificates (VOMS), GSI, myproxy
- **Local Authorization**
  - For C (gatekeeper, gridftpd): LCAS
  - For Java: Authorization framework (org.glite.security.authz-framework-java)
- **Sandboxing**
  - LCMAPS
    - Provides local credentials (unix uid, gid, AFS) needed for jobs in fabric
    - Identity switching
- **Auditing**
  - Job repository
    - Central repository for Logging, Accounting, Auditing

What we would like to see next(1)

- **Authentication**
  - Any SAML assertions, either in-line or retrieved on demand
  - Use generic authN interface for myproxy??
  - Basic authN validation based on TLS handshake
  - But more complex validation pushed to authZ stage:
    - CRL checking
    - Check on authN strength (policy-OID extension)??

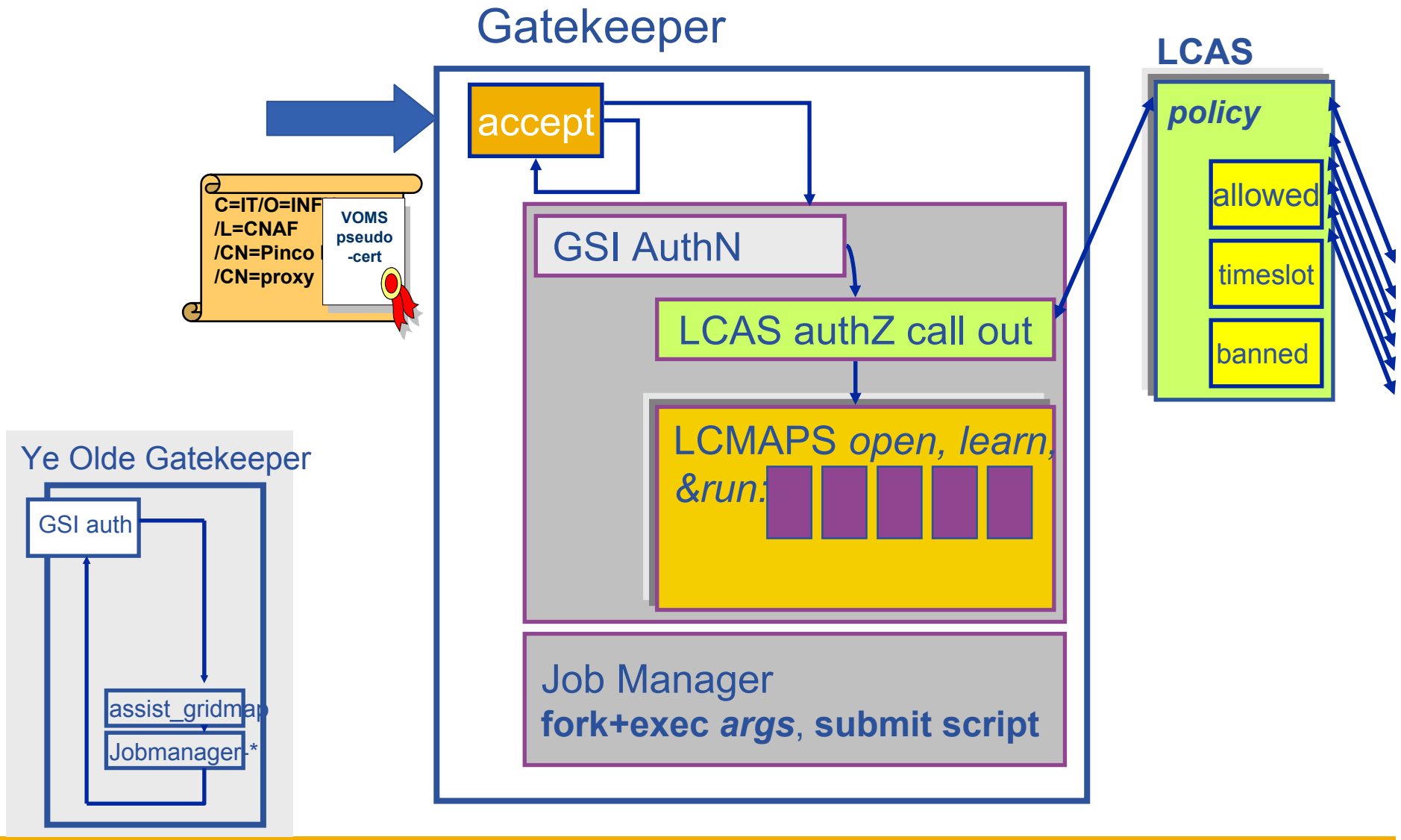
What we would like to see next(2)

- **Local Authorization**
  - Common authZ framework
  - Policy evaluation engine (using XACML)
  - 'Stackable': recursive invocation
  - Policy interpretation by plug-ins
    - Proxy lifetime validation (req. saaa-rg 1.4.1.1)
  - Fit grid authZ in existing systems
    - A grid-PAM module interoperating with the authZ framework
- **Generating audit trails**
  - Site/resource-central service correlates authN/authZ data and local credential mapping

What we would like to see next(3)

- **Sandboxing/isolation for applications**
  - Hosting environment (Java)
  - Host virtualization: Zen, VMWare, UML
    - Probably wishful thinking for EGEE
    - At what level: application, VO, grid??
  - Using unix accounts, groups
    - Transparent for higher level middleware and application
    - Sudo like program takes grid credentials as input
    - A service to dynamically create and delete (pool)accounts, time management, acces control
    - A grid-mapping aware NSS module??
  - Site proxy (or its fancy new name!)
    - Dynamic connection provisioning
    - See Oscar's talk

AuthN, AuthZ in GK (Release 1)



LCAS in release 1

- **Local Centre Authorization Service (LCAS)**
- **Handles authorization requests to local fabric**
  - Authorization decisions based on *proxy user certificate* (with VOMS attributes embedded) and *job specification* (RSL)
  - Supports grid-mapfile mechanism and/or GACL (from gridsite)
- **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`)
  - Plug-in for VOMS (to process Authorization data)
    - Uses VOMS API
    - authZ policy in GACL format (or grid-mapfile)
    - Convenience tool to convert grid-mapfile into GACL format: `voms2gacl`

LCAS - lcas.gacl

```
<?xml version="1.0"?>
  <gacl version="0.0.1">
  <entry>
  <person>
  <dn>/O=dutchgrid/O=users/O=nikhef/CN=Willem van
  Leeuwen</dn>
  </person>
  <allow><read/><write/></allow>
  <deny><admin/></deny>
  </entry>

  <entry>
  <voms-cred>
  <vo>iteam</vo>
  <group>/iteam</group>
  </voms-cred>
  <allow><read/><write/></allow>
  <deny><list/><admin/></deny>
  </entry>
  </gacl>
```

Future of LCAS

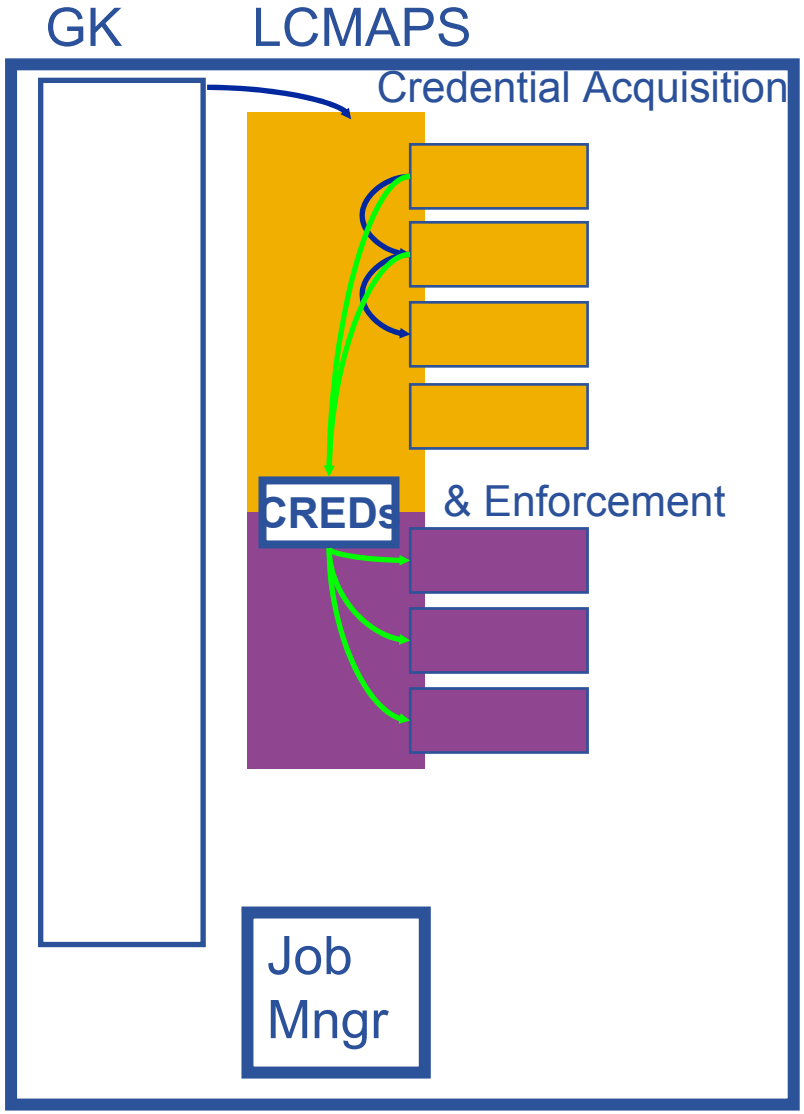
- **Interface to globus authorization call-out**
- **Merge LCAS and JAVA authZ framework into common authZ service**
  - As an intermediate step LCAS can make a call-out to the authZ framework
  - pluggable
  - Re-use of LCAS plug-ins
  - New plug-in functionality (satisfies SAAARG requirements):
    - CRL checking
    - Proxy lifetime checking
- **PAM module interface to the authZ framework**
  - Grid access to cvs, ssh

LCMAPS

- **Local Credential MAPping Service**
- **Backward compatible with existing systems (grid-mapfile, AFS)**
- **Provides local credentials needed for jobs in fabric**
  - Mapping based on user identity, VO affiliation, site-local policy
  - Supports standard UNIX credentials (incl. pool accounts), AFS tokens
  - Pool accounts, Pool groups
- **Support for multiple VOs per user (and thus multiple UNIX groups)**
- ***Plug-in* framework**
  - driven by comprehensive policy language
  - Credential **acquisition** and **enforcement** plug-ins
- **Boundary conditions**
  - Has to run in privileged mode
  - Has to run in process space of incoming connection (for *fork* jobs)

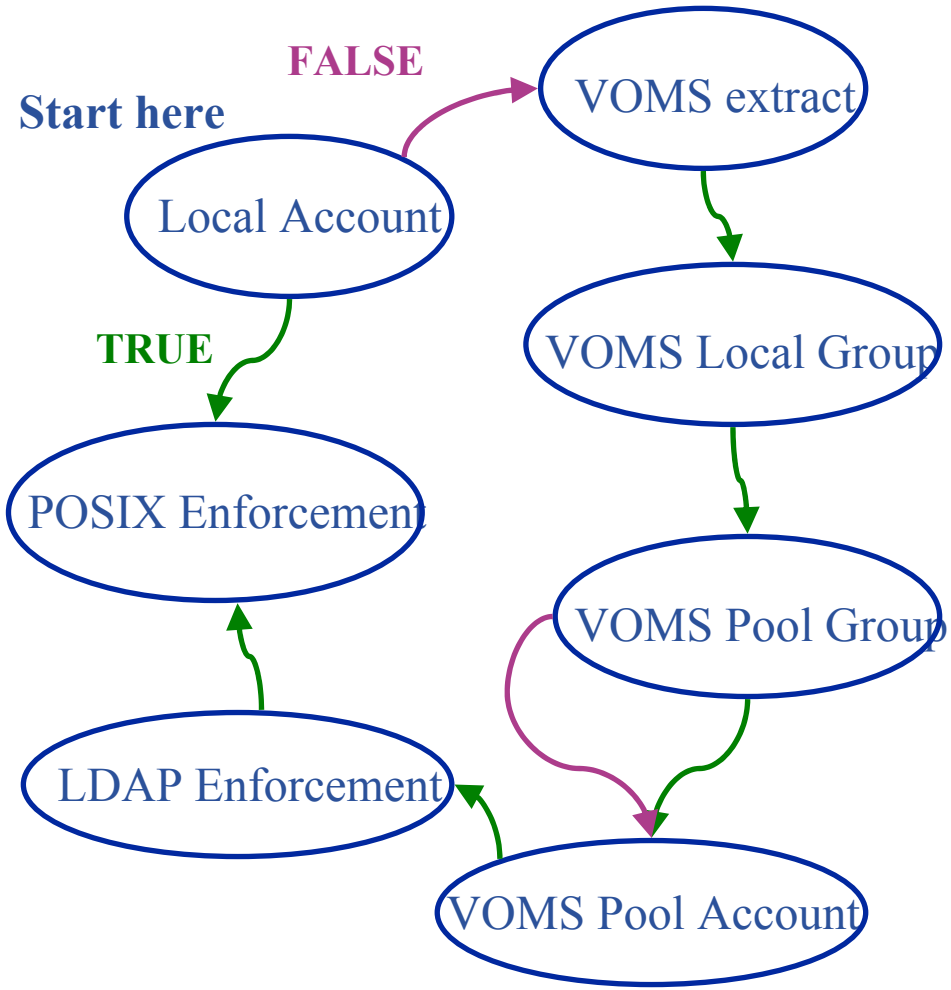
LCMAPS – control flow

- **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**
  - Batch systems will need updated (distributed) UNIX account info
- **Order and function: policy-based**
- **`groupmapfile` for VOMS group-mapping**



LCMAPS – Policy Description Language

State machine approach:




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

# Plugin definitions:
localaccount      =
   "lcmaps_localaccount.mod"
   "-gridmapfile /etc/grid-security/grid-mapfile"
[…]
vomslocalgroup    =
   "lcmaps_voms_localgroup.mod"
   "-groupmapfile /etc/grid-security/groupmapfile"
vomspoolaccount   =
   "lcmaps_voms_poolaccount.mod"
   "-gridmapfile /etc/grid-security/grid-mapfile"
   "-gridmapdir /etc/grid-security/gridmapdir"
[…]

# Policies:
vomspolicy:
localaccount -> posix_enf | vomsextract
vomsextract -> vomslocalgroup
vomslocalgroup -> vomspoolgroup
vomspoolgroup -> vomspoolaccount | vomspoolaccount
vomspoolaccount -> ldap_enf
ldap_enf -> posix_enf
```

LCMAPS – VOMS groupmapfile

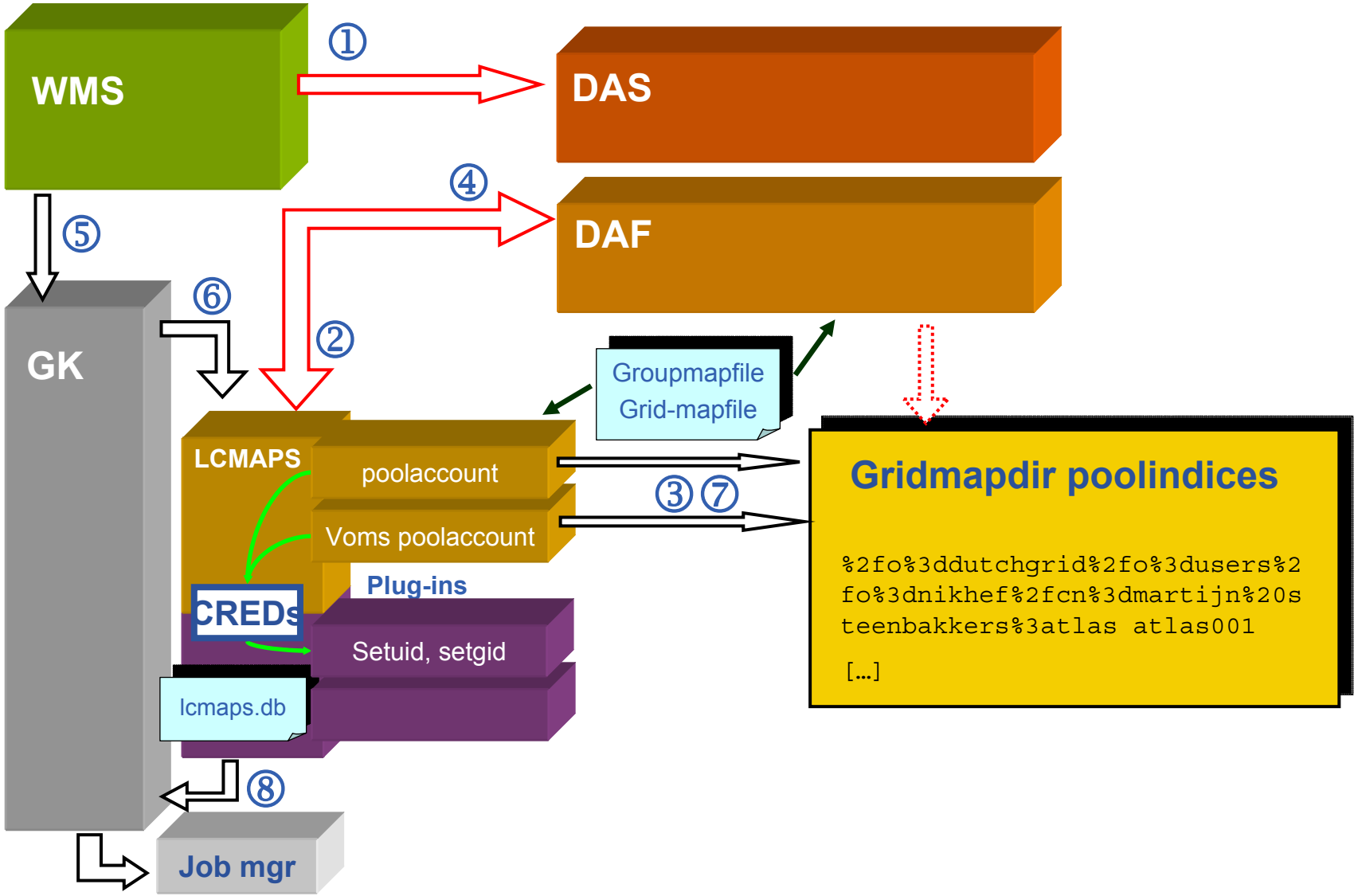
```
# Example groupmapfile:
# Users with this exact VO-group info
# will be added to the local group "fredje"
"/VO=fred/GROUP=fred/ROLE=husband" fredje

# All users from VO wilma will be added to the allocated poolgroup
# "pool[1-9]*"
"/VO=wilma/GROUP=*" .pool
```

- **FQAN not supported yet, but will be (a trivial change)**

DAS and LCMAPS

- **Dynamic account service is part of GT4 (Kate Keahey et al.)**
  - DAS: *Account mgmt interface*
  - DAF: *Creation of accounts*
- **Provides lifetime management**
- **Access control**
  - Currently based on DN
  - Will provide ACLs on VOMS attributes (based on call-out ?)
- **Support of poolaccounts**
  - Clean-up of poolaccounts
  - Use LCMAPS to manage gridmapdir (poolindex)
  - Interface to LCMAPS being discussed
    - Currently directly accessing gridmapdir, not consistent with LCMAPS
  - How to integrate DAS (GT4/WSRF) with gLite (GT2)?

integration of the DAS



LCMAPS future

- **Use a standard credential mapping call-out interface**
  - Being defined in collaboration with globus
- **Replace gatekeeper by a lightweight sudo program**
  - Call-out to authZ FW
  - Use LCMAPS
  - CGI-bin interface to insert into apache server (gridsite)
  - CLI to be used for perl, java
- **NSS module??**
  - Use the JobRepository to look up the grid mapping
  - Example:

```
$ ls -l file_from_atlas
-rw-r--r--    1 /O=dutchgrid/O=users/O=nikhef/CN=Martijn Steenbakkers
   /ATLAS/user/Role=Admin            1 Nov 13 17:22 file_from_atlas
```

Job Repository

- **What?**
  - JR is a Relational Database
  - The data consist of user info. with X509 certs, Job info., VOMS info., Credential info. and the links between these types of info. for every Job
- **Why?**
  - Central repository, Logging, Accounting, Auditing
- **Where?**
  - CE – Plug-in for LCMAPS
  - CE - Various scripts controlled by the Job Manager
  - The database has to be installed close to (or on) the CE.

timeline

Yah!

- **LCAS**
  - Globus callout: 15 December
  - Proxy lifetime checking: this year?
  - Merge with authZ framework: Summer 2005
  - PAM module: ??
- **LCMAPS**
  - Update installation guide + examples: 22 November
    - http://www.nikhef.nl/grid/lcaslcmaps
  - DAS integration: 3 December
    - Depends on what we decide
  - Sudo function: april 2005
  - NSS module: ??
- **Generic authN method Myproxy: ??**
  - contacts with myproxy developers have to be established