Grid-based access control for Unix environments, filesystems and websites

Andrew McNab
(University of Manchester, GridPP and EU DataGrid WP6)

mcnab@hep.man.ac.uk

http://www.hep.man.ac.uk/~mcnab/grid/access-chep2003.ppt

Talk Outline

- Account management issues
- Pool Accounts
- SlashGrid filesystems
- GridSite
- Grid ACL's
- Grid ACL vs VOMS (or CAS)
- Future developments

Authors

The EU DataGrid Security Co-ordination Group,
http://cern.ch/hep-project-grid-scg/

Account management issues

- EDG currently uses Globus2 gatekeepers and file servers, and Globus's GSI model for authentication.
- Authorization is provided by simple text file with certificate names and corresponding local Unix account names.
  - /etc/grid-security/grid-mapfile consisting of lines like:
    "/O=Grid/O=UKHEP/OU=hep.man.ac.uk/CN=Andrew McNab" mcnab
- At the start of the project, site administrators expressed concerns about having to manually maintain accounts and grid-mapfile entries.
- Luca has already discussed how we maintain the grid-mapfile from authorization information published by Virtual Organisations.
- But this still leaves problem of creating a static, named account for each Testbed user at each Testbed site.

Pool Accounts

- Aim to remove local account-creation burden from admins:
  - pre-create pools of accounts and allocate these to users when they request access: atlas001, atlas002, ...
- Widely used by EDG Testbed sites, but not obligatory
  - in practice, almost all have chosen to use it
- Auditing possible since cert => UID mappings recorded in log files.
- Same pool mappings can be shared across a farm by sharing /etc/grid-security/gridmapdir/ lock files with NFS.
- Existing system works ok for CPU-only jobs.
  - but not really appropriate if users are creating long lived files at the site in question.
  - limitations are because files are still owned by Unix UID: can't recycle UID until all files created have been removed.

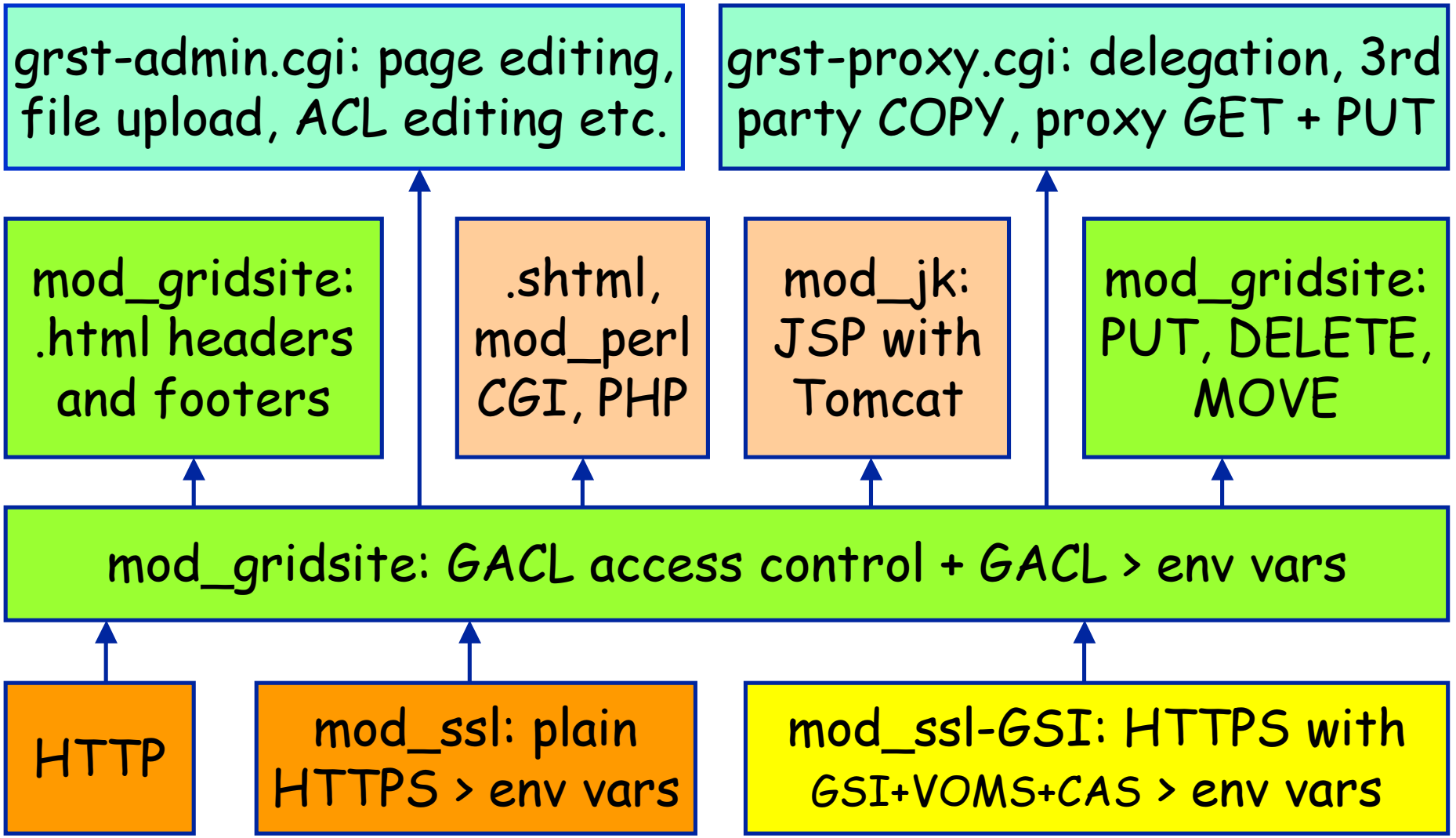
SlashGrid filesystems

- Framework for creating "Grid-aware" filesystems
  - different types of filesystem provided by dynamically loaded (and potentially third-party) plugins.
- certfs.so plugin provides local storage governed by Access Control Lists based on Grid certificate name's and VO groups
  - certfs is very solid: you can build a bootable Linux kernel on a certfs filesystem (~100,000 file operations in a few minutes)
- Since new ACL's just have creator's name, this is equivalent to file ownership by certificate name rather than UID.
  - solves admin worries about long lived files owned by pool accounts.
  - if pool accounts are prevented from writing to normal disks, then no chance they will write something unpleasant somewhere unexpected.
- HTTP/HTTPS plugin (curlfs) ultimately aims to provide some NFS/AFS-like functionality, again governed by Grid creds + ACL's

GridSite

- GridSite manages access to websites and HTTP(S) fileservers
  - Users and admins load GSI cert + key into unmodified web browsers
  - Originally produced for www.gridpp.ac.uk
- ACL's control level of read and write access to file/directory
  - Write access by HTML forms (interactive) or HTTP PUT (programmatic)
- Website admins can define groups of users with specific rights
  - Can delegate administration of that group to one or more members.
  - Group membership can also be published in EDG VO LDAP format.
- GridSite used by several external projects, including UK e-Science Level 2 Grid Support website.
- New 0.9 architecture also provides support for efficient HTTP GET and PUT operations via Apache module.
  - ACL enforcement now available for PHP, CGI, JSP etc as well as HTML

GridSite architecture



Grid ACL's

- When building SlashGrid and other systems like GridSite and the EDG Storage Element, we needed a simple ACL format to use for prototyping.
- Current SlashGrid and GridSite use per-directory XML ACL in .gacl
  - As a file, this can be stored in directories, copied via unmodified https or gsiftp channels and easily manipulated by scripts and applications.
  - Sysadmins wanted disk filesystem ACL's on same physical disk as files if possible (or managed off-site!)
- GACL not a standard, and have provided libgacl with C/C++ API to insulate applications from future changes (eg to subset of XACML?)
- Aim to support "authorization publishing" systems, like LDAP VO, and also "authorization certifying" systems like VOMS and CAS.

Current Grid ACL format

```
<gacl version="0.0.1">
<entry>
    <dn-list>
        <url>ldap://ldap.abc.ac.uk/ou=xyz,dc=abc,dc=ac,dc=uk</url>
    </dn-list>
    <voms-cred>
        <voms>/O=Grid/OU=abc.ac.uk/DN=AbcVOMS</voms>
        <vo>Abc</vo>
        <group>readers</group>
    </voms-cred>
    <allow><read/></allow>
</entry>

<entry>
    <person>
        <dn>/O=Grid/DN=Andrew</dn>
    </person>
    <allow><read/><list/><write/></allow>
    <deny><admin/></deny>
</entry>
</gacl>
```

Grid ACL vs fine-grained VOMS, CAS

- CAS or VOMS can provide ACL-like feature, specifying what capability ("write") is permissible on objects ("higgs-wg-montecarlo")
- However, we think this is too coarse-grained and too heavyweight for all contexts
  - eg if my job creates a temporary, working directory in /grid/tmp, I don't want to have to set up a new entry on the central CAS or VOMS machine
- The two types of system should be seen as complementary
  - when you create some Higgs Monte Carlo data, you set its ACL to give write access for people with "higgs-wg-montecarlo-admin" credential.
  - when you create a temporary working directory, you set its ACL to give only you read and write access.
  - applications should "find their own level" when splitting policy between local ACL or VO-wide authorisation service

Summary & Future Work

- Major management overhead concerns of Testbed site admins have been addressed
- LDAP VO system is currently sufficient, but VOMS will be more flexible and scalable.
- Pool accounts are useful but limited by UID file ownership issues.
- SlashGrid / certfs provides a solution to this.
- GridSite provides a way of controlling HTTP(S) via Grid credentials.
- GACL and gacl library provides Grid ACL's and API.
- Extending this work into Usage Control, not just Access Control.
- See http://www.gridpp.ac.uk/authz/ for links to source code and details of all tools mentioned in this talk