Authorization in L&B

Daniel Kouřil and Aleš Křenek

Major usecases

- job registration
  - determines who is allowed to use particular L&Bserver
- storing events asynchronously
  - store-and-forward protocol
  - user credentials available for 1st hop only
- sharing jobs among users
  - both read and write access (annotations)
  - job, user, and VO levels
  - privileged access (VO admin)
  - restricted views (site admin sees jobs executed at her site)
- user identity change
  - pilot jobs + glexec

AuthZ of store operations in L&B

- LCAS choosen as the AuthZ interface
  - stable C library
  - allows local AuthZ evaluation
  - extensible with plugins
- prototype implemented and tested
  - invocation of LCAS calls from L&Bserver
  - use of generic plugins (banned users, allowed VOs)
  - experimental L&B-specific plugin

Prototype implementation

- LCAS routine invoked for each arriving event
  - client's certificate chain, including VOMS extensions
  - name of the event (as RSL argument)
  - plugins called from within in standard way
- L&B-specific plugin
  - invoked after general plugins
  - finer-grained access control
  - maps event names to allowed client's ids, e.g.,

```
RegJob = {
  *
}

* = {
  /DC=cz/DC=cesnet-ca/O=University of West Bohemia/CN=scientific.civ.zcu.cz
}
```

Prototype implementation (2)

- we will propose merge with LCAS VOMS plugin
  - overlapping functionality
- L&B specific stuff
  - another policy language acceptable by VOMS plugin, or
  - "operation" field of GACL

Summary

- prototype available (CVS HEAD)
- gained LCAS flexibility
- first two usecases addressed
  - job registration
    - e.g.: any user of a given VO is authorized to register a job
  - loggers' web of trust
    - e.g.: only interloggers from VO's WMS can log "important" events
- fix current weakness of the L&B infrastructure
- still lot of work ahead