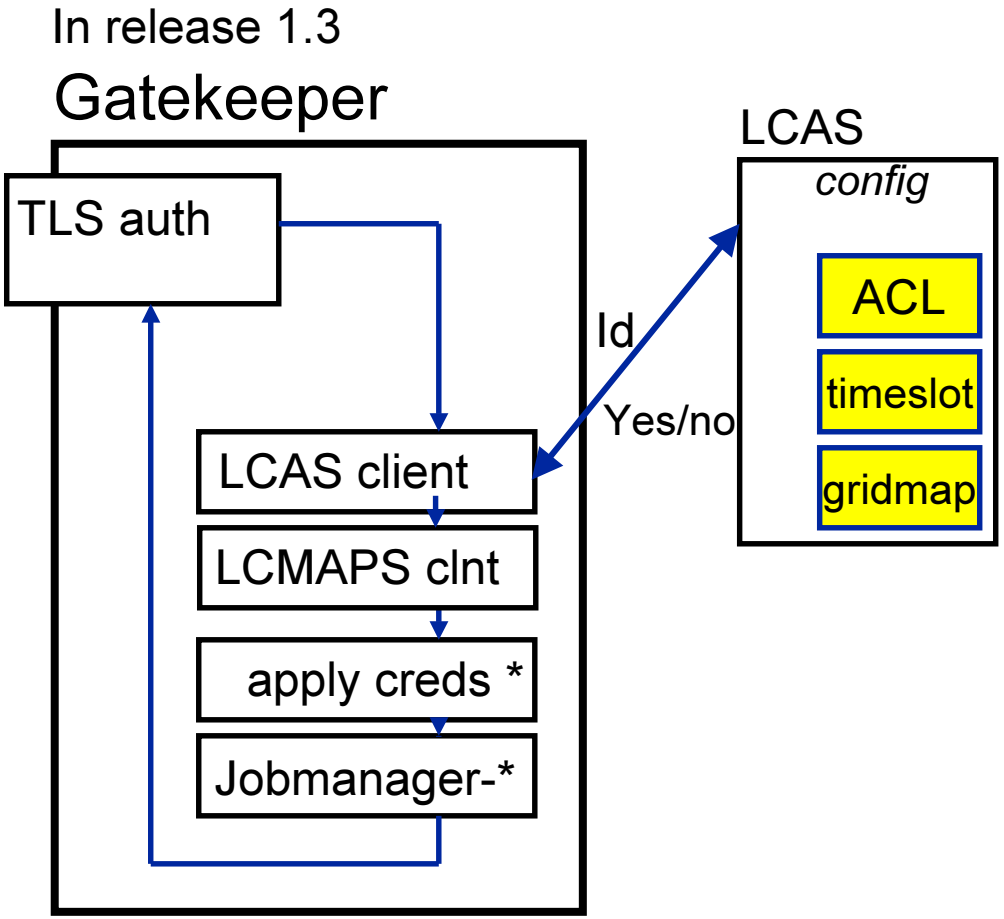
Plugable Authorisation
LCAS and beyond?

WP4 – David Groep

hep-proj-grid-fabric@cern.ch

WP4 LCAS Authorization Service

- LCAS plugable authorization
  - User Credentials
    - Name
    - Full proxy
  - Job request details
- Framework
  - combination of individual modules
  - Simple policy scheme (ordered)
  - Extensible (modules are `.so`'s)
- Near future (03Q1): 'daemon'



LCAS – modifications to service-level driver

- Design goal for LCAS: job-dependent "fine-grained" authorization
- Modules should avail over
  - User credential info
  - Job information

- This context info exceeds what's available on the GSI level:

```
int lcas_get_fabric_authorization(
        gss_cred_id_t delegated_cred_handle,
        char          *lcas_request);
```

- therefore, modification to service (gatekeeper) are required
- same hold for similar extension to GridFTP server
  (still needs API standardisation)

Authorization Call-outs – GSI-only direction

- Von's proposal of September 13th:
  - Modify globus_gss_assist_gridmapfile
  - Support site-defined authorization + *uid mapping* call-outs
  - No fine-grained (no job-dependent authorization)
  - Requires mod's to gridmap.c only (like the PoolAccounts)

- Solves part of the authorization problem
  - Keeps authorization and credential mapping linked together
  - Jobs have to continue till site RMS to get rejected on budget, etc.
  - Is easy to do and has high potential for rapid acceptance (in PPDG+)

- If we want fine-grained authZ,
  we should continue in the new GGF AuthZ working group!

Per user policies and prios in the CE

- Current schedulers (e.g. maui) do job management based on
  - Credentials (Unix uid+gid) and Accounts
  - Queue waiting time
  - Past usage and fair-share
  - Job attributed (requested time, memory, etc) e.g. for backfilling
  - Resources already used / in use
  - *"queues" are no longer used!*
- You can influence scheduling locally by setting weights already now
- Can lead to unexpected quirks!
  - Global scheduling based on free CPUs and Est. Traversal Time (ETT)
  - Currently: single estimate per "queue", no policy info