ACL FOR DCACHE

DAVID MELKUMYAN, DESY ZEUTHEN

CACHE WORKSHOP, DESY, JANUARY 2007

Tigran Mkrtchyan, Patrick Fuhrmann
David Melkumyan, Dirk Pleiter, Peter Wegner

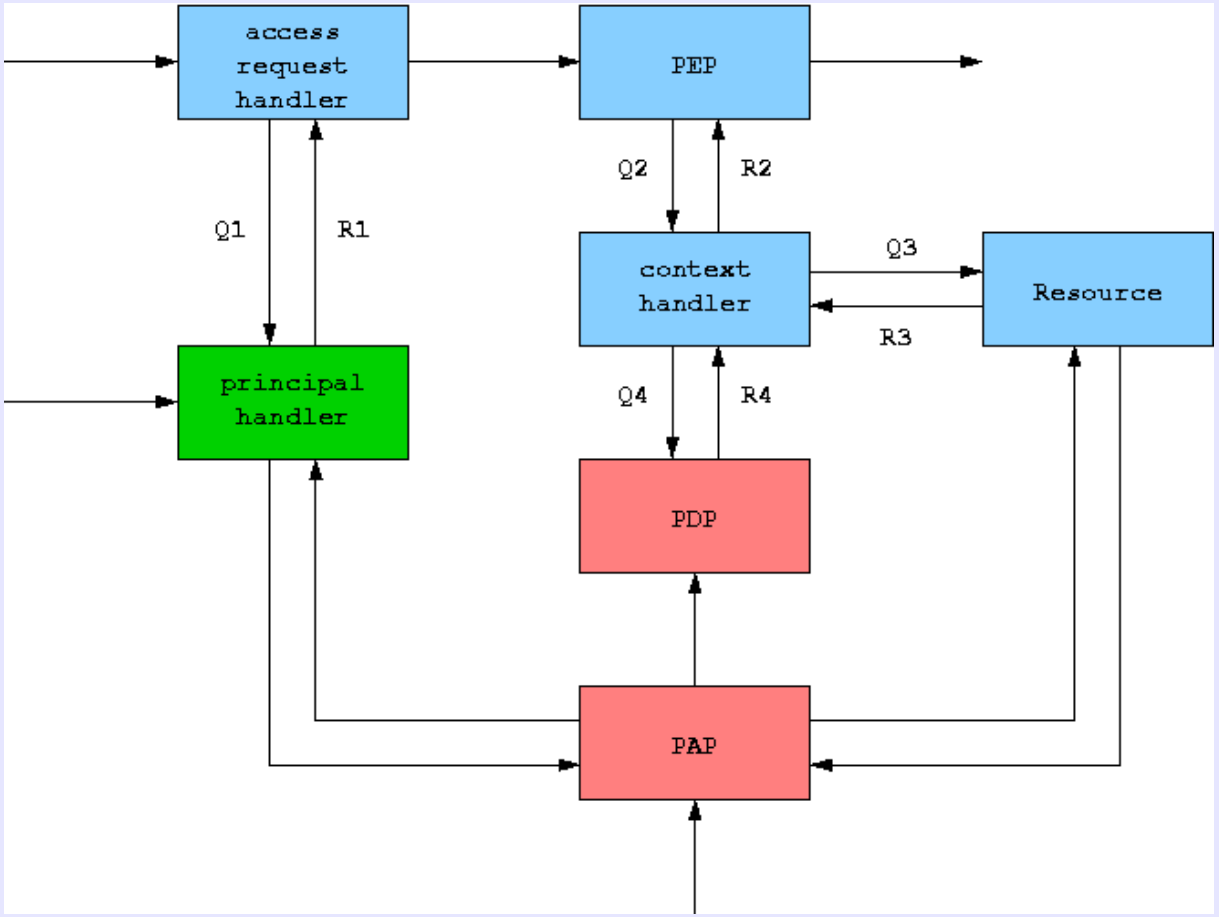
REQUIREMENTS

- Functionality which is compatible to other relevant LCG-middleware components
  - e.g. support for ”POSIX ACL”
- Consistent support of different interfaces
  - POSIX vs. NFS4 ACL model
- Ability to control any kind of dCache/Chimera objects
  - Directories, files, VO spaces, dCache operations
- Modular design that (in principle) allows implementation of different policies

CONCEPTS (1): VIRTUAL IDS

- Subject which performs action is identified by
  - Virtual user ID
  - Primary/secondary virtual group IDs
- **Persistent** and **managable** mapping
  - E.g. Unix ID and DN of the same subject can be mapped on same virtual ID
- Probably implemented via gPlazma plugin

INFORMATION FLOW MODEL



- Components
  - Enforcement point
  - Context handler
  - Decision point
  - Administration point

CONCEPTS (2): NFS4 ACL MODEL

- Access Control Entries
  - Type: access allow, access deny
  - Actions: Read data, list directory, write data, add file, execute, delete, ...
- Subject
  - User, group
  - Special: owner, owner group, everyone, anonymous, authenticated
- Extension: request origin address

NFS4 ↔ POSIX ACL MODEL

- Mapping proposed by IETF draft
- It is possible to
  - Map any POSIX ACL to NFS4 ACL with nearly identical semantics
  - Map any NFS4 ACL to a POSIX ACL preserving certain guarantees
    → Server should not pretend to be more secure than it really is

QUESTIONS?