gLite Authorization Service: Non-technical Overview

Chad La Joie, SWITCH

EGEE '08, Istanbul, Turkey

www.eu-egee.org

e-infrastructure

Agenda

- What's a Policy?
- Goals for the Service
- High-level Model of the Service
- Alternative Deployment Models
- Benefits of the Service
- Timeline
- Transitioning to the Service

What's a Policy

- **Simple policies**
  - Allow ATLAS jobs to run here.
  - Do not allow CMS to submit job to this WMS.
  - Do not allow Christoph to submit jobs as member of Higgs analysis group.
- **More complex policy**
  - Do not accept jobs into the CERN site CEs between 8am and 5pm unless the user is from ATLAS, the job is a SAM job, or the submitting user employed a credential issued by CERN.
  - Only allow jobs from people on this sites white list or where the user is from the Bio grid VO, created their initial proxy certificate less than 8 hours ago, and is not banned by OCST.
  - All jobs started by a pilot job must use a credential associated with the same VO that issued the pilot job.

Goals of the Service

- **Accurate, consistent authorization grid policies**
  - Individuals responsible for a policy create and maintain those policies
    - VO admins write policies dealing with VO matters, site admins write policies dealing with site matters
  - Policies are in effect across the grid in a timely manner
    - Measured in hours, not months
  - Policies are evaluated exactly the same by every service at any point in time
  - The same policy used when running a job is used when matching a job to a resource
- **Provide detailed audit logs**
  - What were the **exact** conditions that resulted in a decision

Goals for the Service

- **Provide good tools to admins**
  - Output should be meaningful, precise, and intelligible
  - Usage of tool should be simple and straight forward
- **Resistance to failure and simple means for scaling**
- **Make the client component very lightweight**
  - small amount of code
  - few dependencies – no conflicts with jobs or other components
  - portability – very low barrier to use on other OSes, in other software, and other languages
- **Flexible deployment model**
  - Allow deployers to balance latency, complexity, resource costs, privacy requirements, etc.
- **Minimize configuration synchronization**

Alternative Deployment Models

- **Groups components within a single process**
  - Minimizes latency
  - Increases dependencies that must be linked in to existing code
  - Increases resources (memory/cpu) needed by existing code
- **Run non-client components centrally**
  - Removes all deployment costs from a site
  - Increases latency
- **At CE sites, run all components on the CE and only install a thin-client on the work nodes**
  - Balances latency with deployment costs
  - Removes all software dependencies from worker node

Benefits of the Service

- **All the benefits derived from meeting the goals of the service**
- **Appropriate locus of control for policies**
- **Enables/eases various authorization tasks:**
  - Banning of users (VO, WMS, site, or grid wide)
  - Composition of policies – CERN policy + experiment policy + CE policy + OCST policy + NGI policy=> Effective policy
  - Support for authorization based on more detailed information about the job, action, and execution environment
  - Support for authorization based on attributes other than FQAN
  - Support for multiple credential formats (not just X.509)
- **Support for multiple types of execution environments**
  - static POSIX accounts, pool accounts, virtual machines, workspaces, ...

Timeline

- **Development is currently ongoing**
- **Estimate 1.0 Beta cycle in February/March Timeframe**
  - This is when services can begin coding against the APIs
- **Expectation is that some of the SCAS will be transformed in to one of the authorization service's components.**

Transitioning to the Service

1. Services that will use the AuthZ service should integrate the client. It is possible to integrate this alongside existing AuthZ code.
2. Sites deploy policy admin and evaluation components. They maintains the same policies as before but do so in the policy admin components. now.
3. As VOs bring policy admin components online, sites remove site-maintained VO policies and import them directly from the VO policy admin component
4. WMS would follow the same model

- Transition can be incremental and does not require a massive, synchronized update effort.
- Eventually developers can drop existing AuthZ code from their code base.