Deployment of Authorization Service

Christoph Witzig, SWITCH
(christoph.witzig@switch.ch)

GDB Feb 11, 2009

www.eu-egee.org

Outline

- Introduction
- Short Description of the Service
- Deployment Proposal
- Policy for Global Banning
- Summary
- Appendix

Institutions Involved

- **CNAF**
- **HIP**
- **NIKHEF**
- **SWITCH**

- **Deployment plan**
  - Devised together with SA1 / SA3
  - Reviewed and endorsed by TMB

- **Note abbreviation: authZ = authorization**

The Plan

- **Starting point: authorization study in EGEE-II**
  - Identified need for consistent authorization in gLite
  - authZ service part of the DoW for EGEE-III

- **Based on input from SA1/SA3 decided:**
  - EGEE-III year 1: development of service
  - EGEE-III year 2: deployment of service
  - Reason: Service should be deployed within EGEE-III

- **Current status:**
  - Service is expected to enter certification in first half of April

Enabling Grids for E-sciencE

Introduction:
Which Problems Are We Trying to Solve?

- **Different Services use different authorization mechanisms**
- **Some services even use internally more than one authorization framework**
- **Site administrators do not have simple debugging tools to check and understand their authorization configuration**
- **Site administrators must configure the authorization for each service at their site separately**
  - Consequence 1: At a site, there is no single point to ban users/groups of users for the entire site
  - Consequence 2: many site administrators don’t know how to ban users
  - There should be a command line tool for banning and un-banning users at a site

Introduction:
Which Problems Are We Trying to Solve?

- **There is no central grid-wide banning list to be used during incidents**
  – Consequence: Urgent ban cannot be taken for granted during incidents

- **No monitoring on authorization decisions**

Introduction:
Benefits of the Authorization Service

- **Main benefit within EGEE-III:**
  - Addressing the above list of short-comings

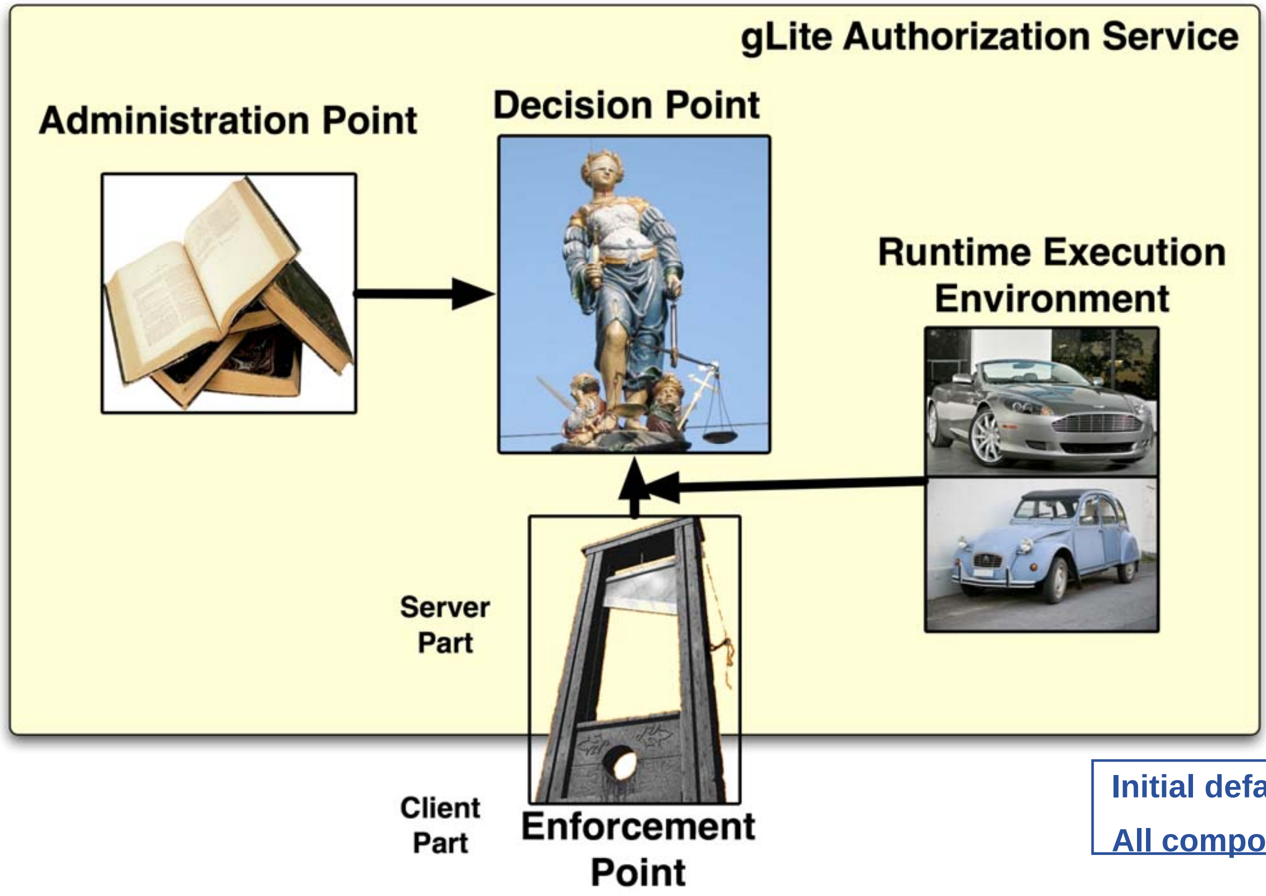
- **There are other benefits: see appendix**

Outline

- Introduction
- **Short Description of the Service**
- Deployment Proposal
- Policy for Global Banning
- Summary
- Appendix

Service Components

Initial rules:
- Banning unbanning
- Pilot job



Initial default deployment:

All components on one host

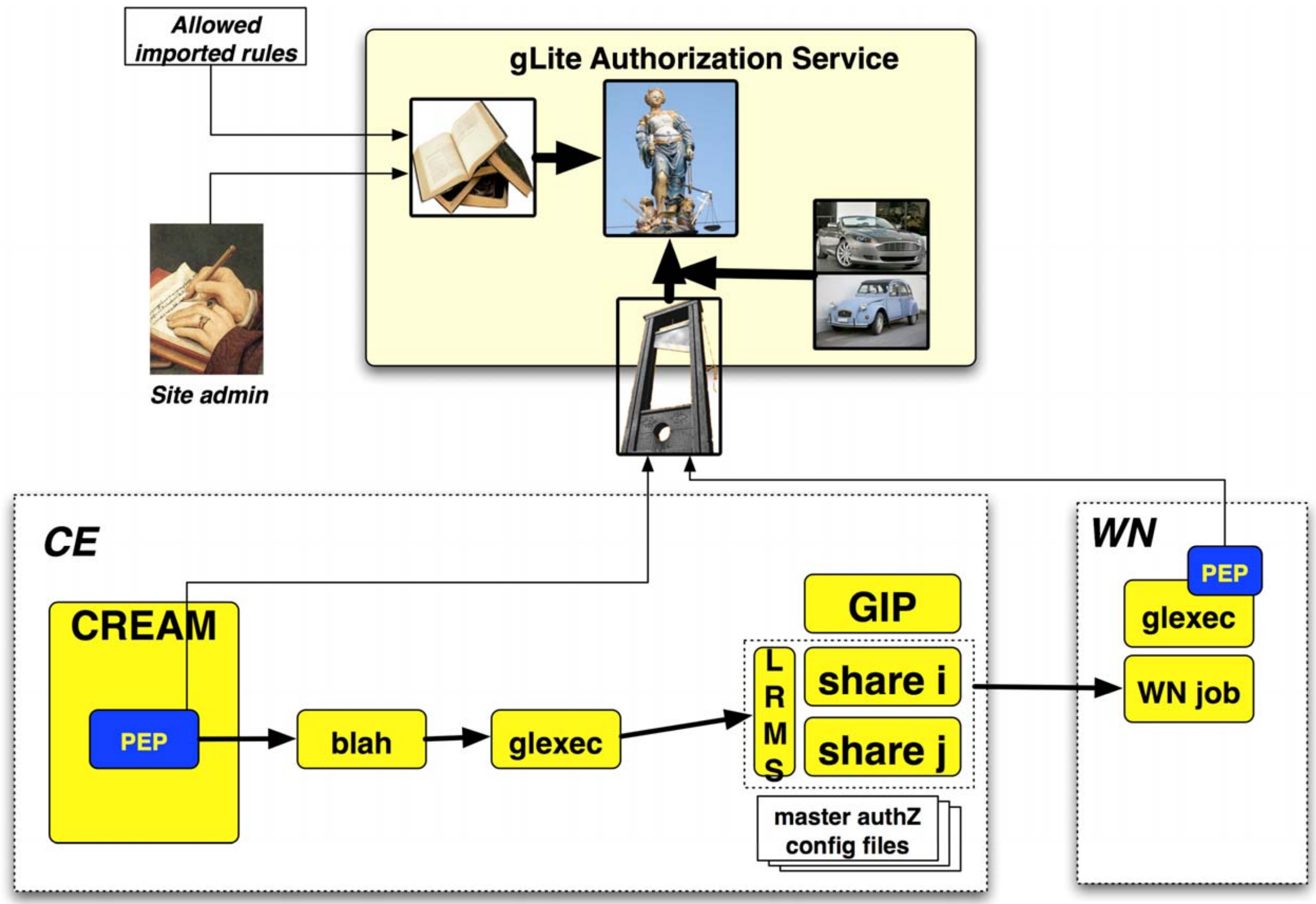
Administration Point: Formulating the rules through command line interface and/or file-based input

Decision Point: Evaluating a request from a client based on the rules

Enforcement Point: Thin client part and server part: all complexity in server part

Runtime Execution Environment: Under which env. must I run? (UID, GID)

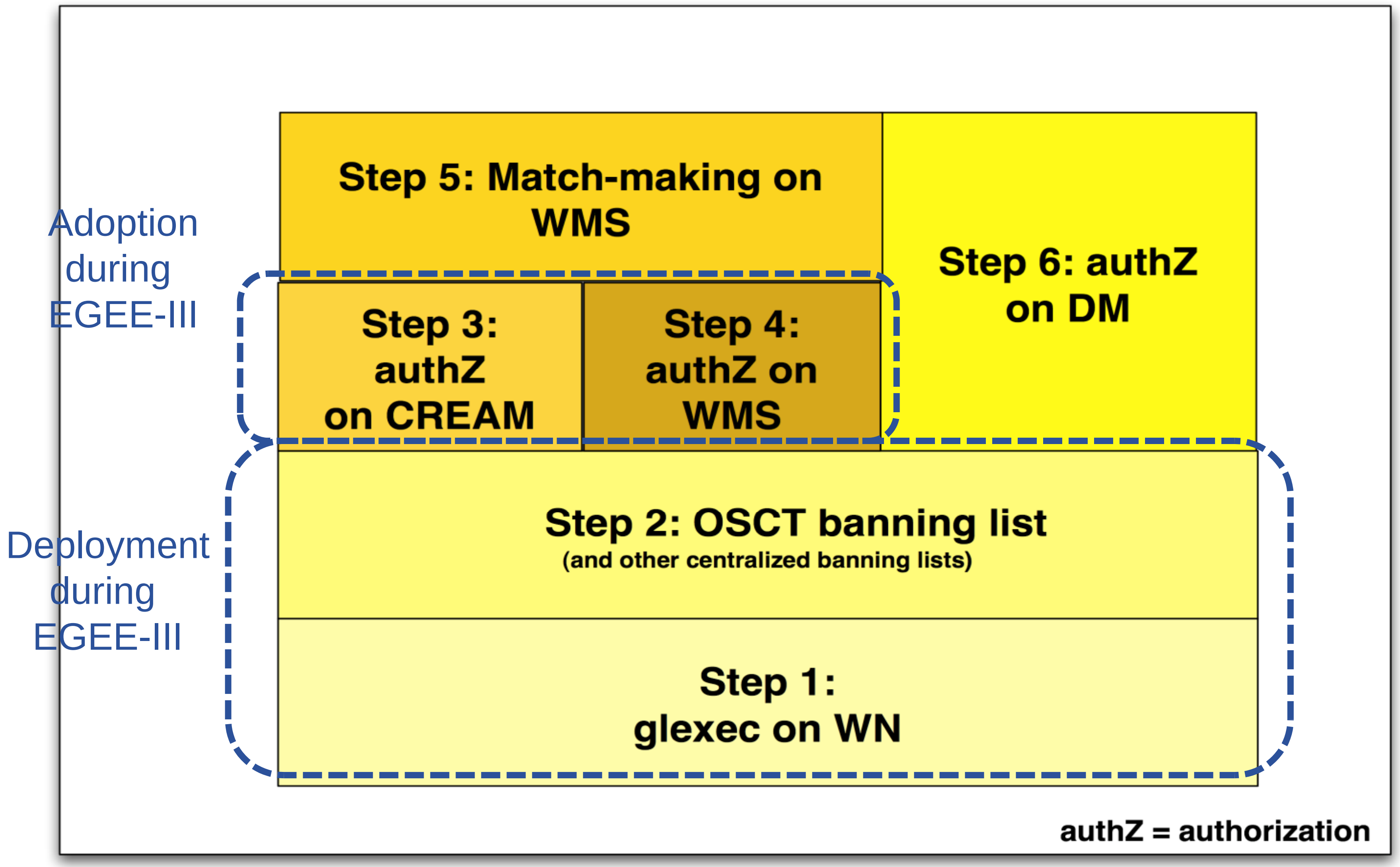
On the CE


Allowed imported rules
gLite Authorization Service
Site admin
CE
CREAM
PEP
blah
glexec
LRMS
GIP
share i
share j
master authZ config files
WN
PEP
glexec
WN job

Outline

- Introduction
- Short Description of the Service
- **Deployment Proposal**
- Policy for Global Banning
- Summary
- Appendix

Proposed Deployment Plan (1/3)

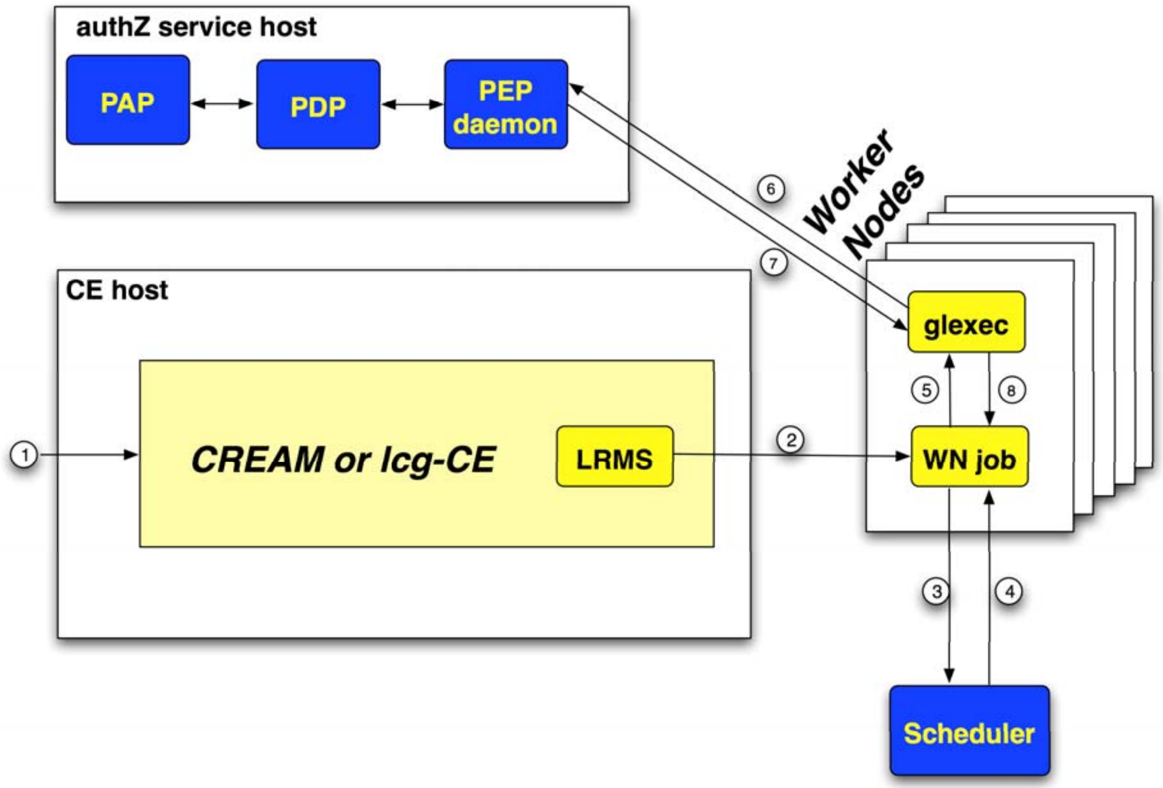


Proposed Deployment Plan (2/3)

Guiding Principle: No big bang but gradually increasing use of authZ service through six *self-contained* steps

1. **glExec on the WN:**

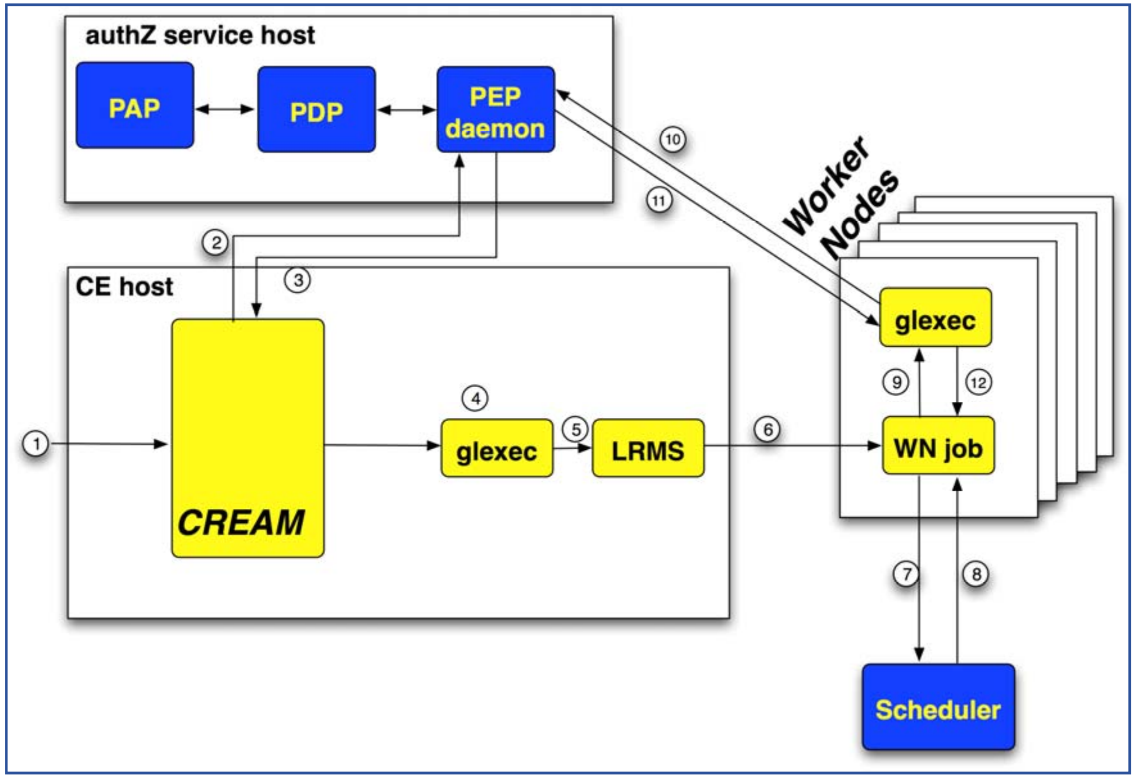
- Only change on WN is new version of glexec / LCMAPS
- Use of authZ service is a configuration option
- Installation of authZ service on one host through YAIM
- ALL policies are local (i.e. no remote policies)
  - Only banning rules and enforcement of pilot job policy
- Note: No change to CREAM or lcg-CE (authZ policy only affects pilot jobs)



Proposed Deployment Plan (3/3)

2. **Grid-wide banning by OSCT**
   - OSCT offers centralized banning list to the sites
   - Policy for this list currently under discussion (see section policy for global banning)

3. **Integration into CREAM**



Alternate Deployment Options

- **Flexibility of the service allows different deployment models**

- **Proposal:**
  - YAIM supports deployment on one single host
  - Alternate deployment options are initially supported by authZ development team on a case-by-case basis

Outline

- Introduction
- Short Description of the Service
- Deployment Proposal
- **Policy for Global Banning**
- Summary
- Appendix

Operational Policy

- **Each site manages its own access policies**
  - Local site autonomy
- **OSCT operates a central banning service (CBS)**
  - Sites SHOULD deploy CBS
  - Sites SHOULD give CBS priority over local policies
  - Sites SHOULD configure CBS so any ban/restore action is active in under 6 hours
    - Time period still under discussion
  - Grid Security Operations MUST inform VO manager whenever user/group access is changed (ban & restore)
- **SHOULD= Obligation with escape clause**
  - Inform Grid Security Office.
- **Currently proposed by JSPG**
  - Discussions continuing.

Enabling Grids for E-sciencE

Policy for Global Banning
(Full text - currently under Discussion)

- Each site manages its own local access policies to its resources. In addition, Grid security operations SHOULD operate a central banning service. Whenever Grid security operations bans a user or group of users, or restores their access, they MUST inform the appropriate VO Manager.

- Sites SHOULD deploy this central banning service and give it priority over local policies.

- The site implementation of the central banning service SHOULD be configured such that any ban or restore action made by Grid security operations is active at the site without a delay of more than 6 hours

Outline

- Introduction
- Short Description of the Service
- Deployment Proposal
- Policy for Global Banning
- **Summary**
- Appendix

Summary

- **Expect service to enter certification in first half of April**
- **Gradual deployment in six self-contained steps**
  - Initial focus on glexec on WN and OSCT ban list
    - Configuration option for glexec
  - Integration into CREAM for authorization
- **Feedback and volunteer sites for trying service out are highly welcome**

Further Information

- **About the service:**
  - authZ service design document: https://edms.cern.ch/document/944192/1
  - Deployment plan: https://edms.cern.ch/document/984088/1

- **General grid security:**
  - Authorization study: https://edms.cern.ch/document/887174/1
  - gLite security: architecture: https://edms.cern.ch/document/935451/2

- **Other:**
  - Wiki: (just started, so pretty empty right now!) https://twiki.cern.ch/twiki/bin/view/EGEE/AuthorizationFramework
  - EGEE08 presentations:
    - http://indico.cern.ch/sessionDisplay.py?sessionId=94&confId=32220
    - http://indico.cern.ch/sessionDisplay.py?sessionId=95&slotId=0&confId=32220 - 2008-09-25

Outline

- Introduction
- Short Description of the Service
- Deployment Proposal
- Policy for Global Banning
- Summary

- **Appendix:**
  - Benefits of the service
  - Feature list of the service

Benefits of the Authorization Service (1/2)

- **Main benefit within EGEE-III:**
  - Addressing the list of short-comings mentioned in the introduction
- **In addition:**
  - Resistance to failure and simple means for scaling the service
    - Flexible deployment model
    - No dependency on a shared file system
    - High availability option
  - Client component is very lightweight
    - Small amount of code
    - Few dependencies (especially on WN)
    - Portability: support on other OS and languages easy

Benefits of the Authorization Service (2/2)

- **In addition (cont.):**
  - 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
    - Virtual machines, workspaces, …
  - Nagios plug-ins provided for monitoring of service

Feature List

1. Policy examples
2. Architectual features
3. Implementation features
4. Deployment features
5. Operational features (for a site admin)

Note:

- Prio1 = within EGEE-III
- Prio2 = beyond EGEE-III
- Label: +, -, o for advantage, skeptic, neutral
- Some of it are features by design, others are features that we are aiming at

Policy Examples: Banning (1/4)

- **Banning users for a site (prio 1)**
  - + easy banning of users for a CE site administrator
  - + banning groups of users, entire VOs, CAs, ….
  - o single banning point for a site (site-wide banning)
    - + possible
    - - needs integration into DM

- **Grid-wide banning (prio 1)**
  - + OSCT maintains a grid-wide ban list
  - o sites must trust external policy

- **VO-banning of users (prio 2)**
  - + VOs can ban the user without deregistering him

- **Regional banning (prio 2)**
  - + regions/federations/nations can enforce banning rules

Policy Examples: VO policies (2/4)

- VO policies (prio2)
  - – - sites may oppose remote policies that they don’t understand
  - – + VO have a consistent means to communicate their policies to sites
- authZ users to run certain applications (prio2)
  - – + VOMS groups/roles are very limiting and don’t consider different types of applications (only admin role)
  - – + who is allowed to submit pilot/payload jobs
- + easy integration into VO specific services (prio2)
  - – o VO schedulers?
- o Decoupling FQAN-shares (prio2)
  - – Less important now (pilot jobs)
  - – Deferred topic - how relevant is it really today?

Policy Examples: DM (3/4)

- **+ use case of banning**
  - - implementation TBD
  - - performance TBD
  - - different SE implementations (DPM, dCache, CASTOR,…)

- **o quota**
  - Open issue

Policy Examples: Other (4/4)

- **+ Better sharing of resources (prio2)**
  - E.g.access based on time
- **+ Better separation of responsibilities across Grid stakeholders (prio2)**
  - + combining different policies from the different stakeholders
  - + adding new policies in a scalable way
- **+ Support for complex sites**
  - Ex: CERN site policy vs site specific VO policy vs running 20+ CEs

Architectural Features

- **+ Exposes policy of a site to the outside**
  - + Pre-requisite for a consistent authorization infrastructure across services
  - + other services/users don't have to second guess whether the job will be accepted
  - + site has possibility for private policies
  - + option of publishing policy or remote PDP invocation

- **+ High availability**
  - + extremely robust
  - + every service component has HA
  - + no single point of failure
  - + no shared file system needed

Implementation Features

- **+ Thin PEP client**
  - + no dependencies on WN !
  - + adding other language bindings is easy
  - + easy to integrate into other services

- **+ Standard compliant**
  - + use of a powerful authZ language (XACML) (+extendable)
  - + SAML2-XACML2 profile
  - + support for SAML assertions built-in from the beginning
    - + credentials beyond PKI, VOMS SAML asssertions

- **o Complexities of XACML hidden**
    - + CLI tools

- **+ Good performance**
    - o hard to get real requirements
    - + aim for several hundred invocations per second

- **+ Several institutions are involved**
  - + long-term support

Deployment Features

- **+ Flexible deployment models**
  - Service can be deployed in various modes
  - Default deployment model assumes installation of all components on one single host (supported by YAIM)

- **+ Gradual introduction into production infrastructure**
  - + no big bang
  - + more services can use authZ service depending on their development cycle
  - + no requirement that all sites make switch to use authZ simultaneously

Operational Features (for site admin)

- **+ easy to use (command line interface)**
- **+ consistent logging, support for incident handling**
  - As defined in security command line tools
- **+ easy and simple monitoring interface**
  - Easy to find out whether all service components work and what it does (Nagios plug-ins will be delivered as part of the service)
  - Command line interface
- **+ easy to troubleshoot**
- **+ nagios plug-ins provided for service monitoring**

EES

- + Consistent handling authZ - scheduling within a CE
- + Consistent way to add new execution environments
- + Support for new execution environments
  - Virtual machines
  - Workspaces

- Is a BIG job
  - Hasn’t really been started yet

3rd Party Code Used

- **OpenSAML / OpenWS:**
  - Source: Shibboleth development team
  - User base: Shibboleth project (~20-30mio users), Danish e-gov, OpenLiberty, ClaritySecurity (National Ass. Of Realtors)
- **Jetty:**
  - Source: Mortbay
  - User base: one of the three major open source servlet containers
- **JBossCache: (in-memory replication)**
  - Source: JBoss / Red Hat
  - User base: JBoss, Shibboleth 1.3