



Enabling Grids for E-science

Grid Configuration Data

or “What should be on the Grid?”

James Casey

GDB, 14th January 2009

www.eu-egee.org



Information Society
and Media



- **How to reliably find a list of site -> service -> VO mappings in the EGEE production infrastructure**
- **Current method:**
 - Do a 'JOIN' between GOCDB (or SAM) and Top-level BDII contents
 - Given a site:
 - 1. Find list of hosts in the GOCDB for the site**
 - 2. Gather from GOCDB the list of services on each host**
 - 3. Lookup GlueService table in BDII for these hosts/services to find supported VOs**
 - a) And the endpoint specific information (e.g. SAPath, Space Token, Service URI)

- **Information is inconsistent**
 - BDII is 'live' view, and things drop out – sites, services,...
 - Incomplete semantics
 - When something drops out of the BDII, it is indistinguishable from the service having been decommissioned
- **What data source is definitive?**
 - What are the rules to reconcile differences?
 - E.g. what if a service is only in the BDII, should I monitor it ?
 - Should I use it for availability calculation?
- **Deletion of services**
 - Site admins need to do this in two places
 - And they forget until the monitoring goes red

- **Monitoring tools**

- SAM

- 'bdii2oracle' script is full of heuristics

- Nagios

- NCG has 'local checks' which try and probe the service directly to validate information

- Gstat

- Has entire infrastructure checking for inconsistencies and differences

- **Grid Middleware**

- FTS needs a canonical list of SRM endpoints

- Has to use caches on FTS nodes because BDII contents isn't reliable

Different caching policies and heuristics leads to operational problems

- **Site Admins**

- Two places to update information
 - Inconsistencies arise
 - Need to input information in a remote DB that they already have put into their own site configuration system
- Confusion on how to remove services from appearing in operational tools

- **ROC/NGIs**

- ‘Principle of subsidiarity’ says that things should be regional where possible.
- Central services are bad in terms of generating single points of failure, and need infrastructure for failover to be deployed
 - Complexity !!!!

- **Clearly, having a single definitive source of information would solve inconsistencies**

Two possibilities:

- 1. Providing the canonical list of VO mappings and endpoint information in GOCDB**
- 2. Providing the canonical list of services which should be at a site in the Site BDII**

- **Pros**

- Already the defacto place for static information
- It's a DB – no fixed schema so easy to add new data
- Makes the GOCDDB closer to it's original purpose
 - Bootstap of Information System
 - Lists of names/phone numbers we don't want in information system

- **Cons**

- It's not on the site
- Actually, not easy to add new things in current GOCDDB3 schema
- Fragmentation of the data will occur anyway in a regional model of multiple GOCDDBs and a cache will be needed anyway
 - All new code to be written
- For consistency, we would have to modify information system anyway to filter entries based on the canonical list

- **Pros**

- It's on the site, controlled by the site admin
- A single place where VO mappings and endpoints are published
 - Where they are already
 - GOCDB could cache this information if needed
 - *E.g Downtime publication.*
- Could also move downtimes here too
 - Single source of data – supported by GLUE 2.0

- **Cons**

- Could be difficult to shoe-horn into GLUE
 - Work still needed on technical implementation
- Some development work needed in middleware rather than operational tools (GIP + BDII)
 - And more YAIM configuration

- **All this is discussion at this point**
 - Within the Operations Automation Team
 - We felt we now needed wider input
 - ... on which way forward is most acceptable to site manager community
 - Or suggestions for different ways to do it
- **Different teams would be responsible for the work...**
 - So we need to then work out the details and effort required with the particular teams

Your input is needed – now !