VO Box Meeting: Summary & Observations

C. Loomis (LAL-Orsay)

Grid Deployment Board Meeting (CERN)
8 February 2006

Outline

- Goals
- Presentations
- Motivations for VO Box
- VO Box Services
- Observations
- Conclusions

Goals

- **Understand how the experiments are using the resources at a site and how they interact with various services located at the site and elsewhere.**

- **In particular, for VO Box services:**
  - Understand the services which run on the VO boxes, their interactions with other grid & non-grid services, and their operational implications.
  - Determine which aspects of these services could be provided by common grid services (either extended or new services).

Meeting

- **Information available from agenda page:**
  - http://agenda.nikhef.nl/fullAgenda.php?ida=a0613
  - Draft minutes.
  - Draft sequence diagrams for use cases.
- **Participants:**
  - ALICE, ATLAS, LHCb
  - Grid deployment group
  - 6 Tier 1 Centers

S. Bagnasco (ALICE)	M. Branco (ATLAS)	S. Campana (CERN IT)	F. Carminati (ALICE)	S. Gabriel (FZK)
P. Girard (CC-IN2P3)	C. Loomis (LAL, Chair)	G. Merino (PIC)	D. Salomoni (CNAF)	M. Schulz (CERN IT-GD)
J. Templon (NIKHEF)	S. Traylen (RAL)	A. Tsaregorodtsev (LHCb)		

Presentations

- **ATLAS & LHCb**
  - Similar architectures
    - § Persistent state in database
    - § Agents do work based on that state
  - Asynchronous data transfers (both)
  - Messaging & Job mgt. (LHCb)

- **ALICE**
  - Experiment interfaces to computing, storage & software
  - Allows “pull” model but using standard services

- **CMS**
  - Data mgt. (PhEDEx)

Motivations

- **Experiments will need application-level services to provide high-level functionality on top of middleware services.**

- **VO Box motivations:**
  - Distributed services: load balancing, reliability, availability
  - Better performance: optimized requests, lower latency
  - Easier integration: other grids, application services

- **Needed in short-term to overcome deficiencies in the middleware.**
- **Longer-term needs further discussion as deficiencies are fixed.**

Services

- **Necessary**
  - Interactive login (gsissh)
  - Proxy renewal utilities
  - Standard grid client tools
  - Access to shared experiment software area
- **Unnecessary**
  - Gatekeeper
  - GridFTP
- **Limited, well-defined network access from**
  - External VO-services or users
  - Jobs running on worker nodes

Credential Handling

- **The handling of user and service credentials by application-level services raises a couple of policy issues.**

- **Application-level service credentials**
  - Use of host certificates by services.
  - Obtaining service certificates for the services.

- **Proxy handling implies VO "superuser"**
  - Alters significantly the grid trust model
  - Particularly problematic if user is member of multiple VOs
  - ACLs (?) could separate "control" from "impersonation"

Security Infrastructure

- **Need clear description of grid security model, along with standard implementations and best practices.**

- **Delegated credentials**
  - Copy of proxy is not a delegated one.
  - Standard code, interface needed
- **Attribute certificates**
  - VOMS-like tickets for application services
  - Split into API to make available to applications
- **Integration of MyProxy servers**
  - Finding location of servers (embed in proxy?)
  - Controlling configuration of servers
- **Certification of new implementations**
  - Large costs, should be last resort option

Group Credentials

- **Multiple people using a single credential**
  - Practical for large productions where there are few users
  - Less adapted for analysis phase with large number of users
  - Raises accounting issues, especially with fabric-level services
  - No strong need of this from the applications
- **Eliminate use of shared credentials**
  - Possible for each experiment to do so
  - Does typically complicate the architecture and implementation
- **User switching**
  - Can avoid reduce overall scheduling costs for set of jobs
  - Significantly complicates accounting at fabric level
  - On balance, a weak motivation for this functionality

App. Service Framework

- **Generic, secure service container**
  - Would move security management back into middleware
  - Provide standard mechanism for controlling app. services
  - Requires significant development
  - Not clear single framework would satisfy all needs
  - Not clear if all security concerns are solved

- **Application services as special jobs**
  - Would need infrastructure for specifying special requirements (inbound network access, unlimited CPU, etc.)
  - How would a persistent state be handled? (Note: same problem exists for generic middleware.)
  - Would high-priority, low-latency scheduling help? (E.g. perhaps for software installation.)

File Transfer Issues

- **Need reliable system which permits transfers to/from any storage element in grid.**

- **FTS**
  - Not ideal for end-user (complicated cfg., limited reach)
  - Serious mismatch in security models
    - § Uses new proxies from MyProxy server, not renewed proxies
    - § Having passwords floating around grid compromises security

- **"VO-plugin" for services?**
  - Do need pre-, post-processing of transfers.
  - Many questions with plug-in model:
    - § where and how are they run?
    - § with what credentials?

Messaging

- **Messaging**
  - Applications must be able to contact exterior services
  - Needs to be reliable and secure
  - Can be used for logging, monitoring, service requests, ...
- **Notification**
  - Middleware cannot be “closed system”; need to interact with non-middleware services
  - Need to perform application-specific tasks based on state of grid
  - E.g.: Registration of file after transfer, validation of file, ...
- **Common solution needed:**
  - R-GMA (?)
  - Dedicated system for messaging (?)

Outbound Network Access

- **Must define a grid-wide policy on outbound network access.**

- **Outbound access not guaranteed:**
  - Complicates significantly the implementation of services
  - Must provide service to bridge firewall for messages
  - End up reinventing NAT functionality
  - Simpler for resource center (maybe...)

- **Outbound access guaranteed:**
  - Large bandwidth not necessarily provided (data transfers should go through appropriate services)
  - No need to modify applications for contacting services
  - NAT could become bottleneck

Common Software

- **Common services**
  - Commitment from experiments, requirements and usage
  - Realization that switching is a cost for experiments
  - Need to have faster development/deployment cycle
- **Grid service APIs**
  - Standard APIs for hiding differences between grids
  - Reduced dependencies between services
  - Evaluation of new protocols (e.g. xrootd)
    - § provides better usability?
    - § worry about having multiple protocols for same service
    - § possible integration with SRM
- **Overall need pragmatic discussions to push toward convergence.**

Responsibilities

- **System administrators**
  - Only responsible for maintaining OS and standard grid services installed on VO Boxes (e.g. gsissh).
  - Ensure that releases are upgraded in a timely fashion.
- **Virtual organizations (experiments)**
  - Responsible for installation, maintainance, operation of VO-specific services.
  - Maintain well-defined releases of VO-specific services and ensure uniform installation of those releases.
- **Monitoring**
  - Generic SFT test for grid services on VO Box.
  - SFT test for VO-specific services.

Conclusions

- **The VO Box discussion raised important policy and technical issues directly and indirectly related to the VO box services.**

- **Need further discussions:**
  - Finalize report from group with suggested list of developments and actions to be taken.
  - F2F or phone conference in early March for this.

- **Need for longer-term discussions of issued raised:**
  - Inclusion of other applications in the discussion
  - Periodic re-evaluation of application-level services at site
  - Integration with TCG?