http://cern.ch/arda

2nd ARDA Workshop, 21 June 2004

"The first 30 days of Glite"

Derek Feichtinger

www.eu-egee.org

cern.ch/lcg

EGEE is a project funded by the European Union under contract IST-2003-508833

Contents

1. Introduction
2. Infrastructure
3. Glite prototype
4. Architecture document questions

Introduction

- Overview over the experiences of the ARDA group using the prototype
- A lot of work has been done by the JRA1 group in setting up and making available the testbed in this short time.
- We consider this early availability of the prototype a very positive achievement.
- Many points will be addressed in more detail in the sessions dedicated to the experiments and specific subjects.

Timeline

May 18	First presentation of prototype to ARDA members
…	Obtaining certificates from CERN CA and registering for EGEE VO
June 2	most ARDA members cannot log in due to a registration/login name issue of an organizational nature.
June 9	Username problems solved for ARDA members. Learning to use the system.

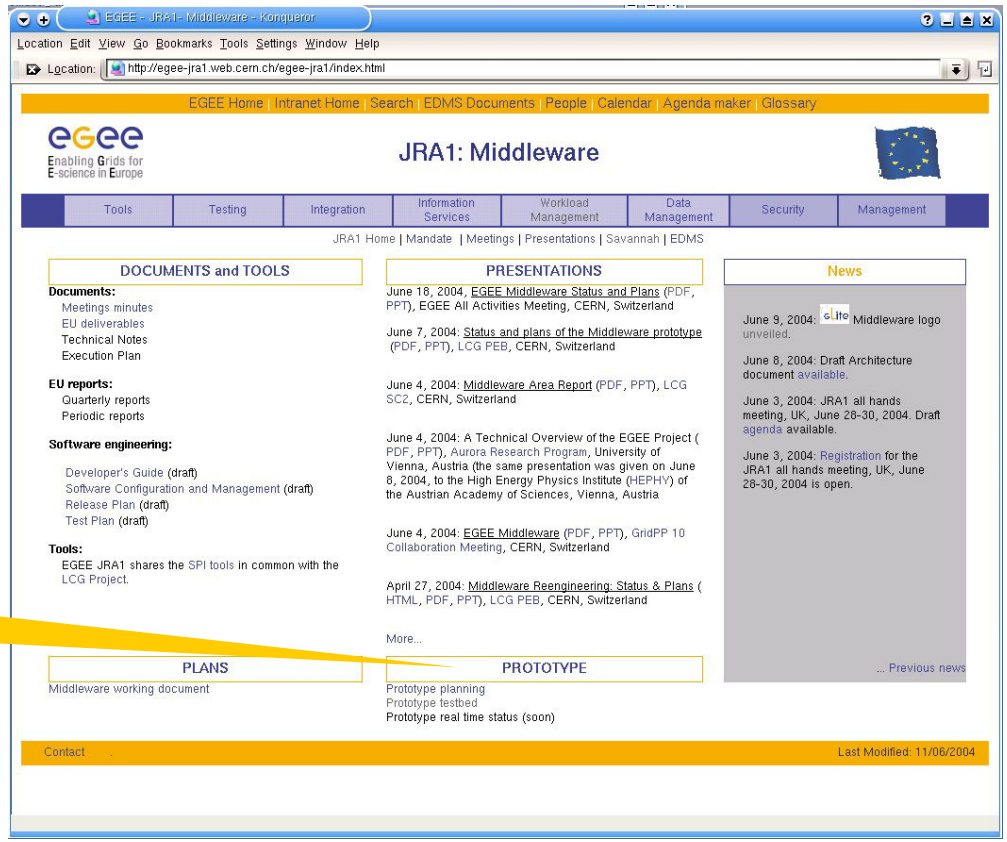
Actual testing took place during ~ 2 weeks

Infrastructure

1. Website
2. Signup procedure
3. Documentation
4. Savannah Trackers

Infrastructure: Website

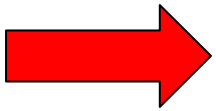
- http://egee-jra1.web.cern.ch/egee-jra1/
- Good structure
- Compact overview



Prototype testbed

Infrastructure: Signup Procedure

- We had agreed on providing a list with desired login names but users signing up to VOMS did not get the desired names.
- A naming convention was enforced by the VOMS manager, but the login names were not communicated to the users.
- The system had problems for some time handling users with different local user names and Glite user names.
- If user does not have the correct certificate loaded in his browser, the VOMS target site seems not to exist at all.



Issues resulting from the name confusion were for most users the principal problem during this phase

These issues may seem trivial but they can significantly decrease our exposure to the prototype. For working with many users, fundamental procedures need to be smooth.

Infrastructure: Documentation

- Registration procedure should be described
  - Obtaining/handling of certificates
  - VOMS signup, possible problems
- Middleware documentation very useful for starting with the system (more in the MW section of the talk)
- A FAQ or HOWTO document outlining solutions for common problems will become useful.

Prototype Documentation:

http://egee-jra1.web.cern.ch/egee-jra1/Prototype/Documentation/glite_tutorial.htm

Infrastructure: Savannah

- Very good experience regarding response time to bug submissions.
- Non logged in users should be prompted for their email address (can be done in savannah configuration).
- Many category values don't make sense to external users. Maybe rather solve this using a field only visible to project members.
- Suggestion: Use of support tracker for user requests that are not actual bugs.

Mailing list and archives:

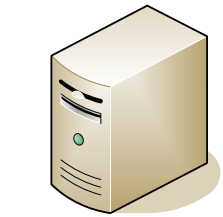
project-egee-middleware-integration-support@cern.ch

https://wwwlistbox.cern.ch/earchive/project-egee-middleware-integration-support/

Glite Prototype

1. First impressions
2. Glite shell: Basic feature requests
3. Tests done
4. First integration activities
5. General problems
6. Requirements for further tests

Testbed



VOMS/myProxy
(lxn5210)

Database, proxy, ldap
(lxn5220)

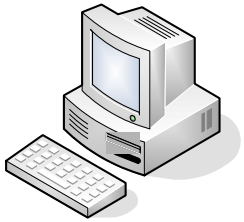
CE (lxn5210)

WN (lxn5211)

GAS
(lxn5216)

Core services
(lxn5219)

SE dCache
(lxn5208)



SE Castor
(lxn5209)

Source: http://egee-jra1.web.cern.ch/egee-jra1/Prototype/testbed.htm

Glite: First Impressions

- Good documentation for getting started.
- Glite shell commands provide online help and show mostly consistent behavior.
- Glite shell sports some nice features like command/filename expansion
- Interesting feature: ‘add’ can store references to external files in the catalogue

Glite Shell: Basic Feature Requests

- Glite shell scripting (i.e. using scripts of glite commands and not workaround via `glite –exec command`)
- some users would prefer having the commands as extensions inside their usual shell instead of a special limited glite shell. Goodies like command/filename expansion can still be implemented in shells like bash (see GNU readline, and bash-completion package)
- Control-C should interrupt the active command and not drop you out of your shell

Glite: Tests done

- GAS was only cursorily tested due to its current limited functionality (simple catalogue interactions)
- Trivial tests exercising the basic catalogue and job execution functionality
- Running a few more complex tests using applications installed via AFS as a workaround for the not yet available package management.
- Some first mass registration and metadata tests
  - following same procedure as has been done for the ATLAS AMI and LHCb metadata catalogues
  - Writing of large number of files (~ 100'000) shows that storage time per files increases.

Glite: First integration activities

- First simple GANGA plugin for job submission and monitoring of job status
- ARDA group members associated with experiments trying to get more complicated, experiment related applications to run (target is four end to end prototypes).
- Developing a new C++ API and plugin for ROOT, which interacts efficiently with the present system (commands are executed via a service situated close to the core system. All transfers use a single custom encoded SOAP string)

Glite: General Problems (1)

Since the newest update was installed only last Friday some issues may already have been resolved.

- Stability of the services
- Tutorial documentation good, but does not provide much help for more than basic usage.
- Documentation does not cover the commands extensively (options to commands missing and also Syntax for some arguments)
- High initial latency for job execution even if queue is empty (up to 10 min.)

Glite: General Problems (2)

Minor issues:

- Online help behavior of commands not useful for unary commands. There should be a standard help flag like –h.
- Some messages returned to the user are cryptic and have no obvious connection with the action (~ more like debug messages).
- Easier retrieval of a job's output sandbox
- Question: should users use the "debug #" command to have a better error description when they submit bugs?

Glite: Requirements for Further Tests

- Software package management
- APIs to GAS (and maybe services). Currently, we will make use of the perl API (q.v. samples of the AliEn test suite).
- Hardware: remote sites are urgently needed to compare wide area characteristics and scaling with other existing middleware (Wisconsin site seems to go online soon).

General question: SOAP

- Not well suited for large data transfers (much larger data representation and slowdown due to XML parsing).
- Not suited for transfers of binary data

- Based on own experiences and experiences with the studied metadata catalogues.
- Phone conference with IBM.
- Publications:

[1] R. A. Engelen et al., “Pushing the SOAP envelope with web services for scientific computing”
http://www.cs.fsu.edu/~engelen/icws03.pdf

[2] K. Chiu et al., “Investigating the limits of SOAP performance for scientific computing”
http://www.extreme.indiana.edu/xgws/papers/soap-hpdc2002/soap-hpdc2002.pdf

MW Architecture Document

Draft: https://edms.cern.ch/document/476451/

- The following slides contain questions that came up during group internal discussions and meetings with members from the experiments.
- Danger of misconceptions because the same terms are used for different concepts by different groups.
- All parties should be aware of the implications of the current architectural design as soon as possible, so they can map their requirements to this.

Architecture: GAS

- GAS (Grid Access Service) seems single point of entry. Document mentions that user may bypass GAS to contact services on his own. Which services, API?

➡ Section 5.1, p 52 (Use cases)

"Note that it is not compulsory to use the GAS; the user may contact and manage the individual Grid services also on his own."

Architecture: GUIDs ,LFNs, …

➡ Section 4.12.2, p 40 (Filenames)

- LFN namespace is a global hierarchical namespace
  e.g. /grid/myVO/production/run/123456/cal-table100
- 1:1 relationship between LFN and GUID
- LFN can be renamed (implies also: pointed at another GUID)
- The GUID can **not** be used as a pointer to a file that can be updated while keeping the same GUID.

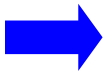
➡ Section 5.1, p 52 (Use cases)

"Currently we do not expect files to be updated."

Document mentions that this functionality will be implemented later

- Question: Are updatable files needed soon? Will there be some kind of versioning system?

Architecture: Metadata



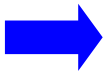
Section 4.14, p 49 (Metadata Catalogue)

"… all metadata catalogues should optimally be provided by the application and not the core middleware layer"

- This is in line with the plans of the experiments, who have their 'proprietary' catalogue implementations.
- Will the prototype offer an own implementation of metadata?
  - needed for system internal MD?
  - May be interesting for workgroups without an own MD catalogue or because queries run more efficiently.
- What will happen with the current Metadata implementation?

"In this context we consider metadata only in the context of file based metadata … with either LFN or GUID as the key binding"

Architecture Document: Datasets



Section 4.12.8/9, p 43 (Directories)

- "Dataset" is used by different groups with different meanings
  - A) A predefined set of Files.
  - B) A dynamically generated set of files, usually created during a job run as result of some metadata query.
- Architecture offers a working model for both cases, if metadata is defined on a per file level.
  - A) A predefined DS consists of files placed inside a single directory of the hierarchical file catalogue.
  - B) A dynamically generated DS creates symbolic links to the member files inside a special directory
- Question: Is the constraint of metadata defined on a per file level too restrictive?

Conclusion

- We were able to conduct a number of simple tests on the testbed.
- Currently stability issues are still the main obstacle.
- First integration attempts for pieces of the experimental software are underway.
- From the interactions with both the JRA1 team and the experiments we believe that the present approach is promising.

We would like to thank the JRA1 team for making the testbed available so fast and for their responsiveness to problems.

ARDA team

- Massimo Lamanna
- Birger Koblitz

- Andrey Demichev
- Viktor Pose

Russia

- Derek Feichtinger
- Andreas Peters

ALICE

- Wei-Long Ueng
- Tao-Sheng Chen

Taiwan

- Dietrich Liko
- Frederik Orellana

ATLAS

- Julia Andreeva
- Juha Herrala

CMS

- Andrew Maier
- Kuba Moscicki

LHCb

Experiment interfaces

Piergiorgio Cerello (ALICE)
David Adams (ATLAS)
Lucia Silvestris (CMS)
Ulrik Egede (LHCb)