

EGEE

WORK PLAN

FOR THE EGEE MIDDLEWARE IN PROJECT YEAR 2



| | |
|----------------------|---|
| Document identifier: | EGEE-JRA1-TEC-573493-Work_Plan-v0.16.doc |
| Date: | 01/07/2005 |
| Activity: | Activity: JRA1 - Middleware |
| Document status: | DRAFT |
| Document link: | https://edms.cern.ch/document/468699/version |

Abstract: This document describes the work plan for the EGEE Middleware software in project year 2 (April 2005 – March 2006) and is part of DJRA1.4 – EGEE Middleware Architecture and Planning (Release 2)

Document Log

| Issue | Date | Comment | Author/Partner |
|-------|------------|---|--|
| 0.1 | | Frist draft | Erwin Laure |
| 0.2 | 09/03/2005 | First input from all clusters | Erwin Laure |
| 0.3 | 16/03/2005 | Security additions | Ake Edlund |
| 0.4 | 21/03/2005 | Added specific issues for discussion | Erwin Laure |
| 0.5 | 22/03/2005 | Changes in release procedure | Alberto Di Meglio |
| 0.6 | 31/03/2005 | Added details for R-GMA, logging, Data mgmt; some restructuring | Erwin Laure (with input from Steve Fisher and Akos Frohner) |
| 0.7 | 31/03/2005 | Added details to Security section | Martijn Steenbakkens, Thomas Sandholm, Olle Mulmo, Mehran Ahsant |
| 0.8 | 01/04/2005 | Added details for IT/CZ parts | Massimo Sgaravatto |
| 0.9 | 06/04/2005 | Modifications before final check | Erwin Laure |
| 0.10 | 08/04/2005 | Modifications of Section 2 and Additions to Security Section | Alberto di Meglio Ake Edlund |
| 0.11 | 11/04/2005 | Final modifications before delivery | Erwin Laure, Frederic Hemmer |
| 0.12 | 31/05/2005 | Modifications for inclusion in DJRA1.4 | Erwin Laure |
| 0.13 | 13/06/2005 | Modifications from UK, CERN DM, and IT/CZ | Erwin Laure |
| 0.14 | 15/05/2005 | Final details for NC | Martijn Steenbakkens |
| 0.15 | 15/05/2005 | Final details for IT/CZ | Erwin Laure/Francesco Prelz |
| | | | |

Document Change Record

| Issue | Item | Reason for Change |
|-------|------|-------------------|
| | | |

CONTENT

| | |
|--|-----------|
| 1. INTRODUCTION | 4 |
| 1.1. PURPOSE..... | 4 |
| 1.2. APPLICATION AREA..... | 4 |
| 1.3. REFERENCES | 4 |
| 1.4. DOCUMENT EVOLUTION PROCEDURE..... | 4 |
| 1.5. TERMINOLOGY | 4 |
| 2. RELEASE PROCEDURE..... | 9 |
| 3. DEFECT FIXING | 10 |
| 4. COMMON CONFIGURATION, LOGGING, ERROR HANDLING, AND INSTRUMENTATION.... | 11 |
| 4.1. SERVICE LEVEL | 11 |
| 4.2. USER LEVEL | 11 |
| 5. SECURITY..... | 13 |
| 5.1. SECURITY INFRASTRUCTURE | 13 |
| 5.2. SPECIFIC SECURITY ISSUES..... | 15 |
| 5.3. APPLICATIONS AND OPERATIONS INPUT TO SECURITY WORK PLAN | 16 |
| 6. MAJOR IMPROVEMENTS TO RELEASE 1 SERVICES | 18 |
| 6.1. COMMON INFRASTRUCTURE | 18 |
| 6.2. SPECIFIC SERVICES..... | 18 |
| 7. ADDITIONAL SERVICES..... | 23 |
| 7.1. HEAD NODE MONITORING | 23 |
| 7.2. ACCOUNTING | 23 |
| 7.3. JOB PROVENANCE | 23 |
| 7.4. CONFIGURATION AND INSTRUMENTATION SERVICE..... | 23 |

1. INTRODUCTION

1.1. PURPOSE

The purpose of this document is to describe the work plan of the EGEE Middleware software in year 2 of the project (April 2005 – March 2006). No timelines further than release 1.2 (July 1st) are given initially since the priorities of these topics and additional topics will be derived jointly with SA1 and NA4. We give however rough estimates as of when the code component could enter the integration and testing process at the earliest. This is noted as “functionality freeze”. The items mentioned below relate to architectural components that are described in the Middleware Architecture part of DJRA1.4 (<https://edms.cern.ch/document/594698/>) [R3] and the relevant sections are referenced where appropriate.

Note that in the following not all services mentioned in the architecture part of DJRA1.4 are covered. This is due to the fact that we tried to give a full picture of our architecture but detail here what realistically can be done in the second year of EGEE. Prototypes of these other services may become available but we don't expect to integrate them into a gLite release.

1.2. APPLICATION AREA

This document applies to the implementation of the EGEE Middleware within the scope of the EGEE Project [R1] and the JRA1 activity mandate.

1.3. REFERENCES

- [R1] [EGEE: Enabling Grids for E-science in Europe](#)
[R2] EDMS quick start guide for EGEE users, G. Zaquine, [EGEE-JRA2-TEC-415472-0-1-EdmsUserGuide.doc](#)
[R3] [EGEE Middleware Architecture](#), EU deliverable DJRA1.4

1.4. DOCUMENT EVOLUTION PROCEDURE

This document can be amended by the EGEE JRA1 Implementation Manager as resulting from feedback and discussion with the other team members as well as SA1 and NA4. The document shall be maintained using the tools provided by CERN EDMS system [R2].

1.5. TERMINOLOGY

Glossary

| | |
|-------|-----------------------------------|
| NC | JRA1 Northern Cluster |
| IT/CZ | JRA1 Italian-Czech Cluster |
| UK | JRA1 UK Cluster |
| DM | JRA1 CERN Data Management Cluster |

Definitions

| | |
|--|--|
| | |
|--|--|

2. EXECUTIVE SUMMARY

This document describes the implementation workplan for the gLite services described in the Middleware Architecture part of DJRA1.4 (<https://edms.cern.ch/document/594698/>) [R3] covering the second project year (April 2005 – March 2006). According to the technical annex a final functional freeze has to be applied at the end of October 2005 (MJRA1.7) hence the implementation plan ends in October 2005.

No timelines further than release 1.2 (July 1st) are given initially since the priorities of these topics and additional topics will be derived jointly with SA1 and NA4. We give however rough estimates as of when the code component could enter the integration and testing process at the earliest. This is noted as “functionality freeze”. After the functionality freeze time for integration and testing has to be allocated as described in Section 3.

The reminder of this document is structured as follows:

Section 3 describes the envisaged release procedure followed by a brief discussion on how defects are being followed up in Section 4. The effort needed for defect fixing may have consequences on the subsequent items of the workplan. The strategies for service and user level configuration, logging, error handling and instrumentation is discussed in Section 5. Section 6 describes the improvements planned for security which is followed by an enumeration of improvements planned to services already part of Release 1 in Section 7. Some additional services planned are described in Section 8.

The following table gives an overview on the services existing in the current Release 1.1 (<http://glite.web.cern.ch/glite/packages/R1.1/R20050430/default.asp>) as of May 12 2005, pointers to the improvements planned and lists the technologies used. We use the same structure as given in Figure 1 of the architecture part of the deliverable:

| Component | Release 1.1 | Plans | Est. Code freeze | Technologies |
|-------------------------------------|-------------|---|--|---|
| Security | | | | |
| Authentication | √ | | | X509 certificates (External), myProxy (External) |
| Authorization | √ | Section 6.1 points: 1. Delegation service 2. Encrypted storage 4. su-exec 5. AuthZ framework 6. VOMS version 1.5 7. Dual cert support | <i>RI.2</i> <i>October</i> <i>October</i> <i>September</i> <i>RI.2</i> <i>October</i> | VOMS, LCAS, LCMAPS, EGEE authorization framework, EGEE su-exec, GSI, GT2, GT4 WSS |
| Auditing | | Section 6.1 point: 3. Job repository | <i>July, October</i> | EDG/EGEE job repository |
| Dynamic connectivity | | Prototype development | | New EGEE JRA3 development |
| Information & monitoring | | | | |
| Basic services | √ | Section 6.2 point: 2. Authorization and | <i>October</i> | R-GMA |

| | | | | |
|--------------------------------------|---|---|---|--|
| | | multi-VO support Section 7.2 points: 17. WS enabled 18. Schema replication 19. Multi-VDB support 20. Boot-strapping 21. Additional DB backends | <i>July</i> <i>September</i> <i>September</i> <i>September</i> <i>October</i> | |
| Job monitoring | √ | | | R-GMA |
| Service discovery | | Section 7.2 point: 22. Integration of service discovery | <i>1.2 for bdII,</i> <i>September</i> | R-GMA, bdII, files |
| Network performance monitoring | | Prototype development | | New EGEE JRA4 development |
| Job management | | | | |
| Accounting | | Section 8.2: DGAS accounting | <i>July</i> | EGEE DGAS, APEL |
| Computing Element | √ | Section 6.2 point: 1. WSS integration Section 7.2 points: 6. Batch System interface 8. Forward requirements to batch system 9. MPI submission independent from file layout 11. Full DAGs executed at one CE 12. public blahp interface 13. new CEMon version 16. Interactive features Section 8.1 Head Node Monitoring | <i>October</i> <i>Condor R1.2,</i> <i>September</i> <i>October</i> <i>R1.2</i> <i>October</i> <i>October</i> <i>R1.2</i> <i>October</i> <i>October</i> | Condor-C, GT2/GT4 GRAM, WSS, EGEE blahpd External batch systems: LSF, PBS/TORQUE, Condor, SunGridEngine, BQS |
| Workload Management | √ | Section 7.2 points: 1. WMproxy 2. LB proxy 3. "pilot job" support 4. Closer integration with DM 5. Interface to Condor-C running as root 7. "Shallow" job re- submission 10. Interface to R-GMA | <i>July</i> <i>July</i> <i>October</i> <i>October</i> <i>October</i> <i>July</i> <i>July</i> | EDG/EGEE WMS, EDG/EGEE L&B, Condor-C |

| | | | | |
|--------------------------------------|---|--|---|---|
| | | 14. "reputability" ranking 15. Policy brokering | July October | |
| Job Provenance | | Section 8.3 L&B base job provenance | September | EDG/EGEE L&B |
| Package Manager | | No development planned | | |
| Data Services | | | | |
| Storage Element | √ | Section 6.2 points: 3. Delegation in DM chain 4. DM security model 5. support for SRMs with native ACL support 6. support for VOMS roles and groups Section 7.2 points: 23. Simpler APIs 27. gLite-I/O and xrootd | R1.2 September September July October October | External SRMs: Castor, dCache, DPM, ... EGEE gLite-I/O |
| Catalogs | √ | Section 6.2 points: 3. Delegation in DM chain 4. DM security model 5. support for SRMs with native ACL support 6. support for VOMS roles and groups Section 7.2 points: 23. Simpler APIs 25. DLI support | R1.2 September September July October R1.2 | EGEE Fireman catalog |
| Data Movement | √ | Section 6.2 points: 3. Delegation in DM chain 4. DM security model 5. support for SRMs with native ACL support 6. support for VOMS roles and groups Section 7.2 points: 23. Simpler APIs 24. Release FTS/FPS 26. Multi-VO multi-channel support in FTS | R1.2 September September July October R1.2 July | EGEE FTS, FPS GT2, GT4 gridFTP |
| Helper Services | | | | |
| Bandwidth allocation and Reservation | | Prototype development | | New EGEE JRA4 development |
| Agreement Service | | Prototype development | | New EGEE JRA1 development |
| Configuration and | | Section 8.4: | | New EGEE JRA1 |

| | | | | |
|-----------------|--|--|----------------|-------------|
| Instrumentation | | EGEE configuration and instrumentation service | <i>October</i> | development |
|-----------------|--|--|----------------|-------------|

3. RELEASE PROCEDURE

Integration builds will commence each week on Friday as now. The integration builds are used to perform quick integration and system tests, but they are not necessarily released to testing.

Tags for the weekly releases must be communicated by Friday morning. They can be either baseline or release tags. Baseline tags should be used for work which is unfinished and does not require a new version number. Release tags do require a new version number and should be used to indicate some milestone has been completed, such as a new piece of functionality or a set of bug fixes.

Once a month a new release build is started. This is typically done around the 15th of each month and replaces the integration build of that week. The build can only contain approved changes (functionality or bug fixes). The tags to be used must be communicated by the cluster release managers by the 15th of each month. The build is integrated for a week and passed to testing for further assessment. After one week of testing, the build can be released with the appropriate version number depending on the changes contained (bug fixes only or a combination of bug fixes and new functionality). All components in this release build must have a release tag, not a baseline tag. Components having a baseline tag must be retagged after testing using a release tag. After the release, testing continues on the same release until a new monthly release is integrated.

Any new features to be included in a release will be considered by the EMT in consultation with SA1 and NA4. Approved features will be eligible for inclusion in the next monthly release prepared by the Integration Team.

Any change that does not preserve backward compatibility needs also to be endorsed by the project before the EMT will consider it for inclusion in a monthly release.

Urgent fixes that cannot wait for the monthly cycle to complete will be released at any time as a QF (Quick Fix) containing only the modified RPMs and short release notes. The QFs will be subject to basic tests on the release currently installed in the test team testbed. The EMT will judge what may be released as a QF upon indication of SA1 or other users.

This scheme allows for three integration/testing cycles to be going on simultaneously: an integration build, a monthly release and a QF. If resources do not permit all three to be done properly, the QF has the highest priority, then the monthly release. The EMT will determine when one cycle has to be abandoned or extended due to lack of resources and the amount of changes in a build.

4. DEFECT FIXING

Defects found by SA1 and NA4 on the deployed Release 1 services need to be fixed with priority dependent on the severity and complexity of the defect. The EMT and eXtended Integration Team will analyze the defects and set the priority on a weekly basis.

A set of high priority defects are identified together with SA1 and NA4 and tracked on a weekly basis involving the project technical management. The current list is available at: <http://egee-jra1.web.cern.ch/egee-jra1/Bugs.doc>

5. COMMON CONFIGURATION, LOGGING, ERROR HANDLING, AND INSTRUMENTATION

5.1. SERVICE LEVEL

See also section 12 of the architecture document.

Common strategies for service configuration, logging, and error handling need to be worked out and applied by all services. Common service instrumentation is needed to extract consistent monitoring information.

Work to harmonize configuration, logging and error handling has already been started, but it has to continue in year 2.

A number of strategies have been devised to add instrumentation capabilities to the services and provide a common access mechanism to status and operations information. In general all services should be revised to add instrumentation capabilities based on standard techniques. The effort and time to adapt the existing services has to be evaluated.

There are two possible general strategies for the monitoring infrastructure:

- LB and R-GMA can be used (LB might need to be revised if necessary) to collect, store and display not only middleware operations information, but also service status and monitoring information (CPU utilization, hardware resources, network resources, etc)
- A new dedicated system instrumentation and configuration service can be added to the middleware stack to collect and forward system and status information to appropriate consumers (R-GMA itself or other standard instrumentation clients). This dedicated service could also be used as bridge between legacy services that cannot be enhanced and the instrumentation consoles. See also Section 8.4.

The logging mechanisms of all services should be revised to comply with the conventions agreed during year 1. In general logging should use, or be compatible with, one of the standard loggers in the Apache Logging Framework initiative (log4j, log4cxx, etc)

5.2. USER LEVEL

See also section 8.3.5 of the architecture document.

To improve the amount of information available to the end-user and to harmonize the access to this information, the existing Logging and Bookkeeping service will be extended in the following ways:

1. More information on error conditions will be added to the trace
2. Statistical information on jobs (CPU consumption etc.) could be included – it is unclear whether this should be the L&B to store that or the accounting system (see Section 7.2).
3. The state transitions in data transfer requests will be logged to the L&B and hence become available in a way similar to job submission requests.
4. The possibility of including the L&B into VDT and thus making it available to a broader user base (in particular in the US) will be explored.

6. SECURITY

6.1. SECURITY INFRASTRUCTURE

1. Delegation service

See also section 5.2.1 of the architecture document.

This service is essential for all other services.

a. Delegation interface and library

The web service interface definition and the library to do the logic. They can be easily incorporated into each web service that needs delegation.

Estimated release: 1.2

b. Standalone delegation service and client

The delegation service and client that can be used by non web service services to do out of band delegation and also as an example of the usage.

Estimated functionality freeze: August

c. General delegation protocol

Standardization work and prototyping towards (possibly WS-Trust based) general widely agreed delegation protocol

2. Encrypted storage

Sensitive applications, for example Biomed, have strict security requirements for data. The data has to be encrypted not only during transfer, but also on storage to prevent data exposure in event of theft or compromise of the storage. To facilitate this the data key management system will offer functionality (based where possible on Data Management modules) to encrypt the data where it is generated, to store the data and encryption secret, to retrieve these and to decrypt the data where the data is needed. This system in addition to protecting the data on storage also makes it harder for malicious system admins to acquire the encryption secret by spreading the secret over several servers.

Estimated functionality freeze: October

3. Job repository for auditing

See also section 5.3 of the architecture document.

Any action taken on a resource via the grid interfaces must be audited and logged. The Job Repository will record all information relevant to the mapping of credentials, the credentials created or linked, and the associated attributes (VOMS groups, roles and capabilities). It is a local component, and does not communicate with and is not intended to be used from outside the site.

Estimated functionality freeze: July (EDG version with revised schema), **October** (standardized interfaces)

4. The following extensions to the security mechanisms, in particular available on the CE, are planned in collaboration with ANL:

See also section 8.2 of the architecture document.

- a. su-exec wrapper for setuid services. Needs to work together with Condor-C, Workspace Service, Cream CE, gridsite apache servers

Services/code running with privileges should be avoided as much as possible. For services that need to switch user identity (setuid, setgid) this can be done implementing a light-weight wrapper program that is easily auditable at the source level, that will always execute with elevated privileges, and will subsequently change user identity. This wrapper will execute a service-determined executable with local credentials that are derived from the user's identity and any accompanying authorization assertions (VOMS attributes). This component will interact with Condor-C, Workspace Service, Cream CE, gridsite apache servers.

Two scenarios will be covered:

The hybrid scenario comprises these two steps:

1. VO scheduler retrieves (multiple) accounts from the WSS using the LCMAPS backend.
2. The VO scheduler switches identity by calling the suexec+LCMAPS-callout and starts the job

The full scenario comprises these two steps:

1. VO scheduler retrieves (multiple) accounts from the WSS using the LCMAPS backend.
2. The VO scheduler switches identity by calling the suexec+WSS-callout and starts the job.

Estimated functionality freeze: October

- b. Interoperability: add globus (>gt3) authZ callout interface to LCAS and LCMAPS (to be defined)

The globus code (gt3 and higher) contains hooks to make callouts to authorization libraries, such as LCAS. LCAS and LCMAPS have to implement this callout interface. In collaboration with globus the hooks in the new GridFTP server have to be defined and added.

Estimated functionality freeze: August

5. Authorization framework PIP and PDP

See also section 52 of the architecture document.

A number of Policy Information and Decision Points (PIPs and PDPs) should be added to the java authorization framework and LCAS (as plugins):

- VOMS PDP: yes or no decision based on VOMS attributes (Java)
- Proxy lifetime checking PDP: ban long living proxies.
- Limited proxy checking PDP: It should be policy configurable for a service not to accept limited proxies. This can be checked by checking the DN or, for RFC3280 proxies, by checking a proxy policy language OID.

Estimated functionality freeze:

- ***VOMS PDP (java authZ framework): September***
- ***limited proxy check/OID check (java authZ framework): September***
- ***limited proxy check/OID check (C, LCAS): September***
- ***Proxy lifetime check (java authZ framework): September***
- ***Proxy lifetime check (C, LCAS): September***

6. VOMS version 1.5

See also section 5.2 of the architecture document.

This adds support for Oracle DB backend.

Estimated release: 1.2

7. Dual certificate support in LCAS, LCMAPS

Estimated release: October

6.2. SPECIFIC SECURITY ISSUES

1. Full integration of WSS as part of the CE

See also section 8.2 of the architecture document.

Estimated functionality freeze: See 6.1. point 4.a

2. Authorization and multi-VO support in R-GMA

See also section 7 of the architecture document.

This will be developed in conjunction with the multi-VO support (see point 17 in Section 6.2). It also depends on the re-factorization of R-GMA described in Section 6.2 which will provide at the same time, authenticated Web Service access and the old Servlet access. Once this is deployed everywhere it will be possible to reconfigure all clients to use only the authenticated Web Services. We will then be in a position to release fine-grained authorization controls.

Estimated functionality freeze: October

3. Delegation in DM chain

See also section 9.5 of the architecture document.

File Transfer jobs are to be submitted to a File Placement/Transfer service. The jobs would be queued and at the appropriate moment executed, starting typically a gridftp transfer among two storage elements. These transfers have to run with the file owner's credential, therefore we need to delegate these credentials through the whole chain from the job submission. We plan to utilize the new delegation infrastructure of JRA3 (Section 5.1, point 1) for the individual steps in this chain, and the WMS proxy renewal service (already existing) in the File Transfer service for the queuing.

Estimated functionality freeze: 1.2 (was already available in 1.1 – configuration option)

4. Work out Grid and local security models

See also section 9.5 of the architecture document.

The data management security model as described in the architecture document needs to be fully implemented.

Estimated functionality freeze: September

5. Integrated with SRMs that have native ACL support (like LCG-DPM)

See also section 9 of the architecture document.

Advanced storage elements could implement a full, fine grained access control themselves, therefore we would not have to enforce it in a file access service (gLite I/O server), depending on the security model (see point 4 from above). This would simplify the file access service, but would bring up the problem of synchronizing file ACLs in a distributed system. The implementation of such an infrastructure requires deeper understanding of the requirements. One possibility is to base the solution on a distributed cataloging system, or use update jobs, similar to the transfer service.

Estimated functionality freeze: September (depends on 4 above)

6. VOMS groups and roles based policies enforced in data and job mgmt; R-GMA

- a) Data mgmt: One has to be able to use certificate DNs and VOMS attributes in file ACLs and service access control settings interchangeably.

Estimated functionality freeze: July

6.3. APPLICATIONS AND OPERATIONS INPUT TO SECURITY WORK PLAN

The security work plan was prepared after input from the architecture documents, Global Security Architecture document (DJRA3.1) and from the Site Access Control Architecture document (DJRA3.2). These documents were prepared in collaboration with SA1, NA4 and OSG.

The suggested list of security modules to reengineer was iterated with the horizontal groups – JSPG and MWSG, and prioritized (first version) at the February 2005 MWSG meeting.

7. MAJOR IMPROVEMENTS TO RELEASE 1 SERVICES

7.1. COMMON INFRASTRUCTURE

1. Migrate to gSoap 2.7
 - o Will fix issues in WS-I compatibility

Estimated migration point: August
2. Migrate to axis 1.2
 - o Will fix issues in WS-I compatibility

Estimated migration point: August
3. Migrate to mysql 4.1, MySQL v5.0 or Postgres SQL
 - o Use stored procedures, nested queries

Estimated migration point: August

7.2. SPECIFIC SERVICES

1. Deployment of the WM proxy

See also section 8.3 of the architecture document.

Web service interface to the WMS. New functionality will be introduced, in particular:

 - o bulk submission
 - * support for job collections
 - * support for parametric jobs
 - o support for jobs with shared sandboxes

Estimated release: July
2. LB proxy feature

See also section 8.3 of the architecture document.

LB proxy is a component to streamline access to the LB server, primarily by the WMS daemons (not the end users). As locallogger, it accepts LB events and forwards them in a reliable way to the LB server. However, it also processes the events to provide a local approximation of job states. On the other hand, WMS daemons can query LB proxy for jobs state and related information (e.g. number of resubmissions). All communication between WMS daemons and LB proxy is done through a UNIX socket, reducing the SSL communication overhead. It is completely synchronous, i.e. the WMS code can rely on logged information to be always present in the proxy, so that handling LB calls becomes much simpler.

Estimated functionality freeze: July
3. Support for “pilot job” submission

See also section 8 of the architecture document.

Generalization of the pilot job approach used by LHC-B, which can be useful when submitting jobs sharing the same requirements, and accountable to the same entity. It allows submitting pilot jobs which control the execution environment and then retrieve (from a "job queue") and execute the real jobs

Estimated functionality freeze: October

4. Closer integration of brokering with data services

See also section 8 of the architecture document.

In particular description and execution of jobs with data placement/allocation stages.

Details of this still need to be worked out but it should include pro-active replication done by the WMS, DAGs combining data movement and job execution jobs, as well as usage of data matchmaking as ranking function.

Estimated functionality freeze: October

5. Interface to Condor-C running as root and/or VO scheduler

See also section 8.2 of the architecture document.

Estimated functionality freeze: October

6. Interface to other batch systems like Sun GridEngine, Condor pools, BQS

See also section 8.2 of the architecture document.

Estimated release: Condor pools: 1.2; September

7. Support for "shallow" job re-submission

See also section 8 of the architecture document.

Instead of re-submitting jobs in case of failures in any stage of the job lifecycle, re-submission will only occur in case a failure happens in the submission process. Once the job started running, it will not be re-submitted.

Estimated functionality freeze: July

8. Forward requirements to local batch system

See also section 8.2 of the architecture document.

Estimated functionality freeze: October

9. Make MPI submission independent from local file system layout

See also section 8.2 of the architecture document.

Estimated release: 1.2

10. Interface Information Supermarket to R-GMA

See also section 8.3 of the architecture document.

Estimated functionality freeze: July

11. Submission of full DAGs to a CE or force execution of all DAG nodes at the same CE

See also section 8 of the architecture document.

Estimated functionality freeze: October

12. Make blahp available to VO schedulers other than Condor-C

See also section 8.2 of the architecture document.

Estimated functionality freeze: October

13. New version of CEMon that allows more easy configuration of VO policies

See also section 8.2 of the architecture document.

Estimated release: 1.2

14. LB-based resource 'reputability' ranking.

This will allow to collect resource failure information and take it into account for match making.

Estimated functionality freeze: July

15. Policy aware brokering

Integration of G-Pbox with the matchmaker. G-PBOX is a system used to define, using standard mechanisms (XACML), and distribute policies. For example VO policies can be centrally defined and then distributed to the various relevant sites. Each site is then free to apply or ignore them. These policies will have to be taken into account during the matchmaking in the resource selection phase.

Estimated functionality freeze: October

16. File perusal and other "interactive" features (e.g. for remote job debugging)

To be able to access files on the WN during the execution of the job and in case also run a "restricted" shell (e.g. to debug the execution of the job). This requires further discussions on the real requirements and the actual effort and estimated functionality freeze will depend on it.

Estimated functionality freeze: October

17. R-GMA WS enabled and re-factored

See also section 7 of the architecture document.

Some significant re-factoring of R-GMA will be carried out. This work should be completed by July 15th. As an example, a PrimaryProducer servlet will be written which will act as 3 different old servlets. This will call a PrimaryProducerImpl, which will finally make use of the existing code. At this stage the PrimaryProducerImpl will correspond precisely to our existing WSDL and could be used as a Web Service, however we will not put any effort into making

the APIs contact the PrimaryProducerImpl directly. This will give a firm foundation upon which to build.

Estimated functionality freeze: July

18. R-GMA schema replication

See also section 7 of the architecture document.

Estimated functionality freeze: September

19. Multi VDB support in R-GMA to support multi-VO deployment

See also section 7 of the architecture document.

This is required for the fine grained authorization described in Section 5.2.

Estimated functionality freeze: September

20. R-GMA boot-strapping

See also section 7 of the architecture document

R-GMA configuration is currently handled at a site level. This makes it possible to have inconsistent settings across sites. We will make R-GMA use Service Discovery with cached results to facilitate restart to identify peer services. This is best introduced after the multi-VDB to support multi-VO has been completed.

Estimated functionality freeze: September

21. Additional DB backends to R-GMA (in particular Oracle)

See also section 7 of the architecture document.

The choice of which backend RDBMS to integrate first depends upon user requirements. Oracle is considered a likely candidate. This is currently low priority and may not be completed in time

Estimated functionality freeze: October

22. Integration of r-gma service discovery

See also section 7.3 of the architecture document.

- a. Data mgmt: The catalog simplified API and command line clients already make use of the service discovery infrastructure. This work will be extended for the transfer clients and will be updated according to any changes to the service discovery API.
- b. Workload mgmt: the necessary SE information as well as catalog endpoints should be available via this.

File based and bdII based service discovery might also be needed.

Estimated release: 1.2 for bdII based SD. September for R-GMA

23. Simplification of DM clients and APIs

See also section 6 of the architecture document.

The auto-generated gSOAP C and C++ API is not easy to use, therefore simplified API will be provided for the catalog and transfer interfaces. The work already started with the library for the catalog command line utilities, which is to be extended with a similar library and command line clients for transfer. These libraries will be updated with service interface changes, but they could themselves hide some of these changes from the programmers.

Estimated functionality freeze: October

24. Include FTS/FPS in the release

See also section 9.4 of the architecture document.

The version of the FTS/FPS released in gLite 1.0 is just a technical preview. The re-factored version of these services, done according to the service challenges needs, need to be included in the proper gLite release.

Estimated release: 1.2

25. Add DLI support to the catalogs

See also section 9.3 of the architecture document.

In addition to the StorageIndex interface, the catalogs will also support the DLI interface.

Estimated release: 1.2

26. Multi-VO multi-channel support of the FTS / Transfer Agents

See also section 9.4 of the architecture document.

The FTS needs to be refactored to be able to provide multi VO and channel configuration properly.

Estimated functionality freeze: End July

27. glite-io and xrootd

Investigation how xrootd works with glite-io needs to be made, and if possible, glite-io should be interfaced to xrootd.

Estimated functionality freeze: October

8. ADDITIONAL SERVICES

8.1. HEAD NODE MONITORING

See also section 8.2 of the architecture document.

The current CE lacks the capability of the resource owner to restrict access to the head node and also requires a separate Condor schedD per user. The plan is to move the CE to a VO based scheduler, which can for instance be a Condor schedD but other software might be possible as well, which is guarded by a head node monitoring system. This work is planned in collaboration with Condor and Globus and in particular includes all the services described in Section 6.1 point 4).

Estimated functionality freeze: October

8.2. ACCOUNTING

See also section 8.1 of the architecture document.

Deployment of DGAS accounting.

Optional instrumentation of CEs with DGAS sensors to "measure" the resources used by the various jobs (or collection of GGF-style usage records measured otherwise). The DGAS system needs to be integrated with the currently deployed APEL system.

Estimated functionality freeze: July

8.3. JOB PROVENANCE

See also section 8.4 of the architecture document.

To keep track of definition of submitted jobs, execution conditions and job life cycle for a long time. It can be used to re-execute jobs, for debugging, post-mortem analysis, and comparison of job executions in an evolving environment

Estimated functionality freeze: September

8.4. CONFIGURATION AND INSTRUMENTATION SERVICE

See also section 12 of the architecture document.

Estimated functionality freeze: October