PRELIMINARY and DRAFT

Report of the

Middleware Selection

GDB Working Group 1

John Renner-Hansen (chair), David Foster (tech. assistant),
Eric van Herwijnen(LHCb), Laura Perini(ATLAS), Tony Wildish(CMS), Yves Schutz(ALICE),
Simon Lin(ASIA), Oxana Smirnova(LCG Applications), Flavia Donno(LCG Deployment)

Date:	**5th December 2002**
Version:	**0.17**
Status:	**Draft**
Editor:	**David Foster**

Document Log

Issue	Date	Author	Comment
0.1	1/11/2002	DGF	First Version
0.2	7/11/2002	ML&DGF	Review by Miron
0.3	8/11/2002	DGF	Security Additions
0.4	12/11/2002	DGF	Post-WG1 meeting
0.5	15/11/2002	FD	Packaging and Monitoring, Software Additions
0.6	21/11/2002	RP&TW&DGF	Post-WG1 meeting + Review R. Pordes, T. Wildish.
0.7	25/11/2002	ML&DGF	Review by Miron
0.8	25/11/2002	Alice&DGF	Review by Alice and Comments
0.9	25/11/2002	WP2&DGF	Review by Peter Kunszt
0.10	27/11/2002	LHCb&EvH	Review by LHCb and comments
0.11	2/12/2002	EvH&OS&FD&DGF	Post-WG1 meeting
0.12	4/12/2002	FD&DGF	Post-WG1 meeting
0.14	5/12/2002	DGF	GDB Review Version

Document Change Record

Issue	Item	Reason for Change

1 Mandate

Select and recommend the grid middleware components to be deployed in LCG-1, with implementation schedules coordinated with the requirements of the experiments and with LCG-1 milestones. The requirements of the experiments, in terms of functionality needed for specific data challenges, will be based on a prioritized list of use-cases from the HEPCAL document. An advisory group representing the experiments will advise on the schedules for delivering the required functionality.

Specifically:

- *Recommend the minimum functionality required for LCG-1 and identify the middleware components required, define associated software delivery requirements for the projects providing the components.*
- *Define a process by which requirements, priorities, and schedules for subsequent additional functionality are determined.*
- *Suggest a middleware development and maintenance strategy. Recommend at which level LCG should assume maintenance and/or development responsibilities for middleware and associated scripts and tools.*
- *Define metrics and performance targets according to those metrics to be achieved by LCG-1, and recommend a process by which the performance becomes acceptable to the experiments.*

2 Table of Contents

Document Log ii
Document Change Record ii
1 Mandate 1
2 Table of Contents 2
3 LCG-1 Planning 3
4 LCG-1 Strategy and Timeline 4
5 Introduction 5
6 Structure 6
7 Overall Assumptions 7
8 The Principal Use Case: Production 8
9 Generic Production Components 9
10 Use Cases 11
11 Starting Proposal 12
12 Functional Components 14
13 Execution Model 20
14 Detailed Starting Proposal 21

3 LCG-1 Planning

"In the first year of the LCG, Grid services will be provided by the middleware produced by the European Data Grid project in Europe, and the Virtual Data Toolkit (VDT) of the US high energy and nuclear physics grid projects[1] that has been adopted by the US ATLAS and CMS User Facilities projects."

[1] PPDG – *The Particle Physics Data Grid* (http://www.ppdg.net/); GriPhyN – *Grid Physics Network* (http://www.griphyn.org/); iVDGL – the *International Virtual Data Grid Laboratory* (http://www.ivdgl.org/).

4 LCG-1 Strategy and Timeline

January 2003	Start working with the recommended technologies – LCG-1 Pilot
June 2003	Stabilise the final configuration of LCG-1 Prototype
July 2003	LCG-1 Prototype in production
July 2003	Start work on middleware requirements for analysis DC in 2004
September 2003	Build next generation of LCG-1 to meet the DC04 requirements
December 2003	LCG-1 Prototype Meets production targets
January 2004	Start preparation of LCG-1 for CMS and Alice DC04

This document only considers the timeframe to July 2003, i.e. LCG-1 Prototype in production.

The explicit software recommendations contained here are for the work starting January 2003 unless otherwise specified.

The intention is to start work with a recommended set of components as soon as possible in order to gain experience and unearth unanticipated problems. The 6 months timeframe before production can be used to introduce later releases that can be shown to be fully functional. June 1st is the latest date by which the final software configuration will be frozen.

5 Introduction

This document proposes the middleware selection for the LCG-1 service to be made available up to the middle of next year. There are a number of important points that must be stated in the context of this document.

- The LCG middleware selection is an evolutionary process that will need constant review to allow new components to be included to enhance functionality and/or replace under-performing components.
- The initial middleware selection is a functionally conservative recommendation that does not exclude the inclusion of components offering additional functionality. To be considered these components have to be shown to be fully operational for the LCG-1 Prototype.
- The base components have been taken from the VDT and EDG activities but this does not exclude the inclusion or replacement of components during the timeframe of the LCG-1 service.

6 Structure

This document is structured in the following way (well not yet but this is the intention):

Assumptions	The overall assumptions on which this work is based are discussed.
Generic Production Model	A generic model of an experiment's production environment
Use cases	A prioritized list of minimal HEPCAL use cases to be implemented by LCG-1
Starting proposal	A selection of middleware and an indication of how this technology should be applied to satisfy the use cases; workarounds incase the technology fails
LCG-1 Architecture	This section shows how the LCG-1 architecture maps onto the top ten HEPCAL use cases.
Functional Model	This looks at the functional components in some detail taking into account the work on the joint US/EDG HEPCAL response.
Execution Model	The application and job issues are considered in this section.
Developments	The additional work required for the creation of the complete LCG-1 environment is considered.
Testing of LCG1	How and what should the experiments test, given that all are also heavily committed to testing EDG middleware.
Support Issues	Support issues and recommendations are detailed

7 Overall Assumptions

This “straw man” proposal starts with a specific recommendation that is then examined in detail. This does not follow the traditional top down approach of defining requirements and then selecting appropriate suppliers of the technology components. This is because:

- There are relatively few suppliers at this time.
- The functionalities supplied are not rigorously modular that is to say there are lots of detailed dependencies between components and implementations.
- There is no clear set of components that matches the HEPCAL requirements.
- We are required to produce a recommendation of functionally working components that can be used to build a service for LCG-1

We have assumed that the recommendations in this document are to be as explicit as possible. Therefore wherever possible we have specified software components and versions in detail.

8 The Principal Use Case: Production

The LCG-1 deployment comes well before the LHC experiments will start data taking and analysis, therefore the main application of interest for the experiments is their Data Challenge exercises, which involve production of big sets of data in order to test computing models and tune the software. It is unlikely that LCG-1 will be able to accommodate a full Data Challenge chain immediately, but it should be possible to perform a partial production in the Grid environment.

Given that there are no grid enabled analysis tools available at the moment, it is clear that the LCG-1 prototype will not directly support an analysis environment until the requirements and tools become available.

In what follows, the term “production” refers to such a specific LHC task as mass generation, simulation or reconstruction of events. An atomic unit of a production is the so-called “data transformation” referred to in the HEPCAL document, which is creation of new data set starting from a set of input data. This use case can be sub-divided into several other basic HEPCAL use cases:

- job submission,
- input data access,
- output data upload,
- update of a metadata catalogue.

A generic job flow can be described as follows:

1. User specifies job information including:
   - input dataset: logical name
   - transformation type and required runtime environment (specific experiment software)
   - output dataset: logical name and location
2. User submits job
3. Grid tools perform requirements match based on job specification
   - Match requested runtime environment and hardware parameters.
   - Chose a location closest to the input data.
4. Input dataset is accessed (either available locally or downloaded from a location discovered by replica location tools)
5. Transformation is performed
6. Output dataset is uploaded to a pre-defined location
7. Output dataset is (optionally) validated
8. Necessary (experiment-specific) metadata catalogues are updated

Functionality including job monitoring and job manipulations, such as cancellation, re-scheduling, re-submission are required. Such operations would also require the proper privileges and authorization for each production manager or assistant.

Production implies performing the steps above in a large scale (up to hundreds of millions processed events), and eventually in chains, performing different kinds of transformation. A sub-case is event generation, when no input data set exists. In any case, production involves movement of large amounts of data. Input and output data are expected to reside on Mass Storage Systems, although certain replicas may exist on disks.

9 Generic Production Components

In recommending software to be deployed in LCG-1, we were careful to take into account specific needs of the LHC collaborations with respect to the production environment, which should benefit from the Grid technologies. To describe the production process, several components must be taken into account. For an LHC experiment, they fall into following groups: hardware resources, services (middleware), experiment's software and users (manpower).

9.1 Hardware resources

While CERN will always host the Tier-0 and a Tier-1 center, the regional centers will play the major role in providing the analysis resources. These resources are:

- Production clusters, characterised mainly by CPU power. Size of such clusters may vary from dozens to hundreds of processors
- Storage resources, such as RAID arrays or conventional disk servers and mass storage systems, such as CASTOR or HPSS.
- Local and Wide Area networks. Given the amount of data, these are of crucial importance, and optimization of data transfer will be necessary.

9.2 Services

Independently of presence or absence of a Grid solution, a production relies on several services, most notably, various databases. Grid adds a set of extra services, not vital for a production as such, but greatly simplifying it in case of a distributed environment. The following services can be considered:

- Replica service – keeps track of file locations and replication tasks.
- Physics meta-data database, linking LDNs to a set of physical parameters
- Transformation meta-data database, containing production instructions
- Production database, keeping logs of the production process and snapshots of the current production status
- Database of production managers and/or authorised users (Virtual Organisation)
- Dynamic database of the production system status (Information or monitoring System)
- Resource discovery and scheduling service, taking care of optimal use of the production system (Resource Broker)

9.3 Experiments software

To this group belong all the software components which are specific for LHC experiments. Most notably, these are:

- Core experiment software distribution (libraries, APIs, tools, utilities etc)
- Experiment-specific production tools, providing necessary higher level interface to the above mentioned services

9.4 User base

Specifics of the LHC experiments is such that researchers and hence potential production managers and operators are distributed all over the world, being thus in need of a synchronised set of production tools. To use such remote operation tools, authentication and authorisation issues should be addressed seriously. A "conventional" approach of sharing production manager passwords can hardly be accepted. In general, people concerned with production can be listed as follows:

- Local system administrators, having all the privileges, but only on local resources
- Local production operators, regular users of the local resources (e.g., Ph.D. students)

- Production managers, authorised to supervise and influence all the production process
- The experiments policy dictates who has read-only and read-write access to the databases.

10 Use Cases

The HEPCAL document describes a variety of use cases which are expected to be in place by the time of LHC data taking. On the time scale of LCG-1, however, only a subset of them is of immediate importance. Experience with the existing VDT and EDG systems suggests the following list of top use cases (complete with implementation details):

	Use Case	***Technology Used Today***	***Grid implementation***
1	Obtain authorization	At every production center we need a generic account. Jobs are submitted under this account.	Globus Certification Authorities
2	login		Globus PKI implementation Proxy Renewals
3	Browse resources	Static Lists	Globus MDS and LDAP tools
4	Job submission	LSF, PBS or Condor job submit commands. The Grid would make this easier as we would be able to use the same command everywhere.	Globus/Condor submission, or a Resource Broker submission
5	Job control	LSF, PBS, Condor Tools.	Globus or EDG workload management tools
6	Job output access or retrieval	Stdout written to disk or mailed back to user. Data written to local disk or straight to mass store.	Globus or higher-level User Interface
7	Data transformation	LSF, PBS, Condor Tools.	Composition of cases 5-6, described either by the Globus RSL, or JDL (Condor classAdds)
8	Error recovery for failed production jobs	Failed jobs are moved to an error directory and resubmitted by the agent.	Some recovery for grid failed jobs. Manual recovery for the rest.
9	Dataset upload	Experiment specific tools	GridFTP, Replica Manager and GDMP
10	Dataset registration	Experiment specific database	Globus Replica Catalog, RLS, grid specific metadata catalog
11	Dataset transfer to/from non-grid storage		GDMP patch, Replica Manager
12	Dataset access	UNIX I/O	Replica Catalog/RLS, metadata catalogs GridFTP, UNIX I/O
13	Browse experiment database	Experiment specific	GSI-enabled database access tools
14	Dataset replication	Experiment specific	Globus Replica tools and GridFTP, GDMP, Replica Manager
15	Dataset deletion	Experiment Specific	Globus Replica tools and GridFTP, GDMP, Replica Manager
16	Job monitoring	LSF, PBS or Condor tools and experiment tools	Logging and bookkeeping service
17	Software publishing	Experiment Specific	Pacman or RPM

11 Starting Proposal

It is assumed that by mid-2003 the middleware and applications will be required to run in a RedHat Linux 7.3 environment with gcc 3.2 or whatever future agreement is made by then.

This proposal starts with a minimalistic middleware base that is shown to run in this configuration today.

Components from both the EDG and VDT software suites are proposed.

It is proposed to start with the package that is VDT 1.1.5 and add the higher level services provided by the EDG. It is assumed that VDT will evolve towards VDT 1.2.x (based on NMI.R2) during the first half of next year which itself incorporates Globus 2.2 and Condor 6.x.

Procedures for software delivery, bug reporting and delivery of bug fixes and functionality enhancements will be coordinated with EDG, the VDT team and the Globus project.

The detailed content of these packages is listed in Annex A.

Constraints

An attempt is made to place minimal restrictions on local fabrics used to participate as part of the LCG-1 Prototype grid. We are assuming that the limited number of machines used for the LCG-1 Pilot grid are dedicated to this purpose to avoid the complexity of considering the possible interactions with existing cluster environments. However it is clear that we need to evolve to a point where resources of shared clusters can be made available to the LCG as soon as possible.

There may be scalability issues with certain choices of infrastructure configuration. In addition there are many possible variants of batch queuing systems and cluster configurations, file systems, storage management and archival systems.

11.1 GLUE

The Grid Laboratory Universal Environment (GLUE) was an activity mandated by the HEP Intergrid Collaboration Board (HICB) to address the technical issues of interoperability between the different middleware initiatives in the US and Europe. Its main deliverables have been agreements on common schemas that allow the information services to interoperate and thereby make the resources available in either grid implementation available to the other.

It has therefore started to help define standards for interoperability between different implementations. This middleware selection document leverages this work by selecting

middleware that conforms to the GLUE standards wherever possible. It is to be hoped that future developments will build on this commonality and define common standards for interoperability and the required schemas from the beginning.

12 Functional Components

The functional model is broken down into broadly the same categories as used in the Joint EU/US HEPCAL response document but has been extended in scope.

- **User Tools and Portals**
- **Information Services**
- **AAA (Authentication, Authorisation, Accountability) and VO**
- **Meta-Data, Data management and Access**
- **Virtual Data Management**
- **Job Management and Scheduling**
- **Application Services and Higher Level Tools**
- **Packaging and Configuration**
- **Monitoring**

12.1 User Tools and Portals

This document attempts to specify user client tools that are part of the basic toolkits involved (VDT and EDG/WP1).

Implications

These tools and portals are yet to be studied but there is no reason to specify specific software packages at this stage.

12.2 Information Services

The information services will be provided by MDS 2.2 together with the GLUE schema 1.x. Static and dynamic information providers provided by EDG (WP3 & WP4) will also be installed.

Implications

We are not assuming the EDG-WP3 RGMA functionality at this time.

12.3 AAA and VO

The currently existing mechanisms for VO user management and resource access control are very limited. As a first selection we will deploy the EDG mkgridmap mechanism that creates the Globus gridmap files from a central VO catalog that is an LDAP directory of all VO members. Each VO needs to run their own LDAP VO catalog.

Other than that there are little or no tools for user account management for the grid.

So many security issues are not treated yet:

1. Account and Access procedures
2. Auditing and Logging Requirements
3. Firewalling and inter-site systems connectivity
4. Acceptable Use Policies
5. Incident Handling Procedures, including incident "ownership"
6. Vulnerability Assessment and follow up
7. IDS requirements, implementation & operational follow up
8. The relationship between individual user rights and site policies.

There are ongoing efforts to have real solution for at least the first item, i.e. authentication and authorization for the mid-term. The Globus as well as the EDG Security Groups have prototypes for managing VO user certificates and authorization (the Community Authorization Service CAS from Globus and the VO Membership Service VOMS from EDG). There are new dedicated working groups at the GGF where both these groups as well as other interested parties are represented.

The two above-mentioned services (CAS and VOMS) will need to be evaluated when they become available for usage in production systems. The advantage of the VOMS approach seems that VOMS proxy certificates would be backward compatible with existing non-VOMS enabled services (exhibiting some default behavior) while CAS certificates are not backwards compatible and all services would need to be upgraded when CAS is introduced. However, this may change in the future so both services will need to be tracked.

This document does not deal with the many policy issues associated with security.

For LCG-1, only the simplest mechanism for authentication and authorization is recommended. This includes:

- Interworking with DOE and EDG certificates. We are going to accept certificates signed by the DOE and EDG approved CAs and map them to local accounts.
    - Issue: Are these certificates mapped to “real” accounts or “temporary” accounts.
    - Issue: Are we recommending EDG dynamic accounts? What explicit facilities does this include?
    - Issue: For GDB/WG3. The certificate info needs to be reviewed to ensure that it is acceptable for DOE centers.
- Use of the EDG mkgridmap generator – including the associated perl modules.
    - Issue: for GDB/WG3: We can offer GSI and XK509. We may also consider MyProxy in case we want to support automatic proxy renewals.
- Use of the KCA certificate generator from Kerberos tickets.
    - Issue: How will the certificates be renewed for batch jobs.

12.4 Storage Services

The storage element (SE) provides services to manage storage space including allocation polices and provides a uniform interface for data access for all levels of the storage hierarchy. The storage element also implements the authorization policies for data access.

A complete solution for a storage element is missing and it should be considered as a priority for the realization of the LCG-1 prototype.

Technologies that are addressing part of the problem include such developments as the Storage Resource Manager (SRM) which provides a common interface to different storage systems (Disk, HPSS, Jasmine, Enstore – see SC2002).

The progress in this area needs to be tracked. The Castor team is working on their own implementation of the SRM interface.

Recommendation: CMS is currently investigating SRM/SRB and will report. This will not be completed by December.

Implications

This lack of a storage element has ramifications in the area of the job execution model.

An essential storage service will be the ability to write onto a generic storage backend such as Castor or HPSS (for instance via bbftp) in such a way that the data can be used outside the LCG-1, as the majority of the physicists that need to access this data will initially not be able to use LCG-1.

12.5 Data Management and Access

To avoid re-synchronisation problems, all job input data are assumed to be read-only. We assume that the problem of two jobs concurrently modifying two copies of the same grid data is not supported.

In order to track the data, we expect to use the existing replica catalogs; initially the Globus LDAP-based replica catalog which is currently being replaced and tested with the Replica Location Service that is a joint project by Globus and EDG and has a relational database as its backend. The functionality we expect in addition for the mid-term are the replica manager components from EDG-WP2. These include the Replica MetaData Catalog (RMC) and possibly the Replica Optimizer service although this is not required for the LCG-1 Pilot.

The Replica Manager interface to the RLS and the RMC has been provided for testing to the Pool team by mid November.

Issue: Pool, SRB, DCache, DRM, etc all come with their own metadata structures and services. How these can be used and interfaced to the storage element will need to be studied.

12.6 Virtual Data Management

There is no virtual data management system assumed for LCG-1. However, we will do our best to accommodate evolving virtual data management tools as they mature.

12.7 Job Management and Scheduling

The job management and scheduling software is an important part of LCG-1, and absolute stability and proven reliability will be required.

We are assuming that the extra functionality of the EDG WP1 resource broker will work with the VDT 1.1.5 package for the LCG-1 Prototype. Work to validate this should start as soon as possible.

The requirement for DAGMan functionality for LCG-1 will be investigated.

Implications

There must be a committed development activity for VDT and the EDG to support this.

If any of these components are not working, it will be possible to submit jobs directly to the batch queuing system without use of the middleware.

12.8 Application Services and Higher Level Tools

For LCG-1 we are not assuming any higher-level services such as reservation. However, we can use local scheduling policies as agreed by the GDB to manage throughput distribution among groups/experiments/ etc.

We do not assume system management tool, problem determination tools, user support tools or end user tools above those that come from the VDT release 1.1.5 and the EDG testbed 1.3. However, we anticipate that future systems architecture activities and implementations (such as OGSA) will identify and facilitate implementations of tools in each of these areas.

For these future systems we will also need to develop transition plans and compatibility requirements.

Implications

Use of grid resources in an optimal manner with higher level services that treat quality of service or service level agreements scenarios will not be possible with LCG-1 Prototype.

12.9 Distribution and Configuration

The process of software creation and deployment goes through various phases. During the development phase, developers setup a directory structure for the code, put it in a CVS repository, use ***software building tools*** to build the software binaries on a specific OS or on multiple platforms with multiple flavors (threaded, debug, 32-bits, etc.).

Once the software is built, ***packaging tools*** are used to create a software *binary* distribution that can be installed at other locations. At each site, different ***software installation tools*** can be deployed to start from some software repository, which can be local, and use it to install software on the machines managed by a system administrator. The repository can also serve local users to install the software on private desktops. (A simple example of such repository can be a CD and a software distribution tool can be the software used by the specific OS to read the CD and install the software starting from it). Once the software has been installed, ***software configuration tools*** are used to properly configure the software reflecting the local site policies and configurations.

LCG has to provide a solution to the following problem: give support in terms of packaging/ distribution, installation and configuration of GRID middleware and application software for large and middle size farms, manually or automatically managed and for desktop users. In particular solutions for first time installations, updates, patches, upgrades and un-installation of software bundles should be provided. The resulting installation must be the same in terms of functionality and setup no matter what is the size of the installation and the procedure followed among the supported ones.

Developers are left free to choose the most appropriate software building tool (such as the GNU autotools): LCG will not deal with software building issues.

In the current situation EDG and VDT provide distribution tools that are quite different in terms of approach. While the EDG distribution is based on the Red Hat RPM package and installed and configured at a site using (ng)LCFG, a farm management tool, VDT adopted a more portable and user oriented model based on PACMAN.

Two working groups are investigating for giving recommendations and/or solutions on packaging, distribution, installation and configuration for large farms and desktops. These are the GDB/WG4 and the working group setup by the GLUE effort under the coordination of HICB. The results of their work will be taken by LCG.

Issue: The versioning method used for the LCG distribution should reflect in some way the main releases of the packages part of the distribution.

- How can this be achieved?
- What will happen with VDT and EDG versions?
- How can we ensure that the right releases of the software are actually deployed and distributed coherently ?
- How do we propagate updates ?

Issue: The methodology for software updates needs to be defined. For example, there are 3 main approaches:

1. “Load on demand” where a missing executable version is retrieved from a repository as a job is run.
2. “Sandbox” where the required file versions are sent with the job.
3. “Static Management” where the administrators are informed when a new file version is required to be made available.

We expect the issues above to find an answer in the result of the work of GDB/WG4 and GLUE.

12.10 Monitoring

At the moment complete monitoring solutions available to control the behaviour of the grid infrastructure are missing. In order to spot problems as soon as they arise and guarantee a good behaviour of the testbed, especially in a production environment, the problem of the monitoring tools should be considered seriously.

DataTAG and iVDGL in the context of the GLUE effort have developed good prototypes of Monitoring tools based on the Nagios and Ganglia technologies. Frameworks such as MonaLisa are being used for experiment specific solutions.

We consider that those tools are good candidates at least for a first pilot of the LCG testbed. The usability, reliability and usefulness of such tools can be experimented in large scale and a good interaction with the developers can be established.

Issue:

Development and support of such tools are an open question.

13 Execution Model

The execution model of a job on the LCG-1 grid is the form of job execution that is most widely used today. The job should not necessarily be “grid aware” to run.

This simple model has three logical steps:

1. Pre-Job execution. In this step the environment for the job can be prepared including copying any necessary files to the worker node. This could involve interrogating the RLS and locating a copy, checking access rights to resources, etc.
2. Job Execution. During the job execution, output files may be produced. Although the job can determine itself where output files are written (for example directly to a remote storage system during job execution) it is anticipated that the most common model will be to write files directly on the worker node.
3. Post-Job execution. This “tidy up” phase can be used to tidy up the execution environment and transfer the results of the job execution to some storage system.

Issue: In addition it will be required to gather and archive job fault information and logs for later analysis, and post-mortem analysis as well as job re-scheduling. Currently the resource broker will catch jobs that have failed due to grid problems. These jobs will then be re-submitted. At this time job failures due to the job itself are not processed and the end user will have to deal with that case. All logs are passed to the logging and bookkeeping service. An appropriate development may be the re-direction of SOUT and SERR to the user directly.

Alternatively, some well defined activities (such as data movement) could be prepared and executed as jobs in their own right unders some circumstances.

Issue: If the job fails at step 2 and the wrapper dies with the job, who takes care of tidying up the environment?

Variants on this model are not excluded, for example the preparation of the job environment automatically by the Resource Broker.

Implications

Jobs requiring or producing large amounts of data will require worker nodes with appropriately sized disks. The technology of abstracting storage through a generic interface (storage element) that allows a job to execute without knowledge of the underlying storage architecture is not assumed to be ready or deployable for the LCG-1 Prototype.

Jobs can explicitly issue storage requests through GRIDFTP or RFIO services. This has implications on the authorization model. Pooled accounts running on the worker nodes need to be properly mapped to the RFIO accounts by some means.

Issue: Will POSIX I/O be ready? From where, by when? What does Condor provide?

14 Detailed Starting Proposal

14.1 Middleware to be deployed on the first LCG-1 Pilot for January 2003

VDT 1.1.5:

Globus 2.0 + openssl security patch + new job manager.

ClassAds: 0.9.3

Condor: 6.4.3

Condor-G: 6.4.3

EDG Certificates 0.12-1

EDG CRL Update: 1.2.5-1

MDS 2.2 + static GLUE schema

EDG mkgridmap

EDG 1.3.0:

GDMP 3.2.5 (to be migrated to 4.0)

Replica Catalogue Server (to be migrated to RLS) edg-rc-server 3.1-1

Replica Manager edg-replica-manager 2.0.6

Replica Catalogue API /CLI ReplicaCatalogue 3.2-3

Workload Management (informationindex-1.2.9-1, jobsubmission-[glue-aware], lbserver-1.2.14-1, locallogger-1.2.12-1, proxy-1.2.8-1, userinterface-1.2.15-1)

LCG:

A program to tell you what versions of the VDT and EDG software has been installed: lcg-version.

Configuration solutions.

DataTAG 1.0:

GLUE Information Providers

Edt-monitor

Dependencies:

Perl--5.003

Python 1.5.2

Convert-ASN1-0.16

Filesys-DiskFree-0.06

IO-Socket-SSL-0.91

Net_SSLeay.pm-1.20

perl-ldap-0.26

FTSH 0.0.9

Pacman 2.098

Implications

Experiment specific packages are not considered here.