Grant agreement no. 211714

neuGRID

A GRID-BASED e-INFRASTRUCTURE FOR DATA ARCHIVING/ COMMUNICATION AND COMPUTATIONALLY INTENSIVE APPLICATIONS IN THE MEDICAL SCIENCES

Combination of Collaborative Project and Coordination and Support Action

Objective INFRA-2007-1.2.2 - Deployment of e-Infrastructures for scientific communities

Deliverable reference number and title: **D7.3 Final Grid Middleware Migration Test Report**

Due date of deliverable: **Month 36**

Actual submission date: **February 8th, 2011**

Start date of project: **February 1st 2008** Duration: **36 months**

Organisation name of lead contractor for this deliverable: **maat Gknowledge**

Revision: **Version 1**

<table>
<tr><td colspan="3" align="center">Project co-funded by the European Commission within the Seventh Framework Programme (2007-2013)</td></tr>
<tr><td colspan="3" align="center">Dissemination Level</td></tr>
<tr><td>PU</td><td>Public</td><td>X</td></tr>
<tr><td>PP</td><td>Restricted to other programme participants (including the Commission Services)</td><td></td></tr>
<tr><td>RE</td><td>Restricted to a group specified by the consortium (including the Commission</td><td></td></tr>
<tr><td>CO</td><td>Confidential, only for members of the consortium (including the Commission Services)</td><td></td></tr>
</table>

Table of Contents

Preface

The neuGRID project (http://www.neugrid.eu) aims at developing a new user-friendly grid-based research e-Infrastructure enabling the European neuroscience community to carry out research required for the pressing study of degenerative brain diseases. In neuGRID, the collection/archiving of large amounts of imaging data will be paired with computationally intensive data analyses. Neuroscientists will be able to identify neurodegenerative disease markers through the analysis of 3D magnetic resonance brain images via the provision of sets of distributed medical and Grid services.

The work presented in this deliverable contributes to the project in the following areas:

- Infrastructure topology and available grid software/services,
- Grid middleware migrations within the Proof-of-Concept and Production Environments.

1 Overview

This document aims to report on the approach and procedures which were implemented in neuGRID to test and propagate available new releases and/or updates of the gLite grid middleware [1] being developed in the European Enabling the Grid for E-sciencEs (EGEE) project [2].

As can be appreciated from neuGRID's project description, one of the main goals of work package 7 (WP7) is to test new gLite middleware components in an isolated environment prior to deploying them in the production infrastructure, thus keeping end-users away from unstable software.

As a consequence, WP7 has been and still is successively migrating new versions of the software as they are felt mature and necessary in terms of functionality. Beyond this, WP7 has the following objectives (extract from project description):

- To evaluate and if necessary develop appropriate grid interfaces to gLite services, to programmatically make the grid functionality available to developers,
- To undergo education and training in the use of gLite services and its APIs,
- To ensure updates and releases of gLite are available, to test them in an isolated environment prior to deploying those in the production infrastructure,
- To liaise with EGEE and enrich the grid middleware requirements from new ones identified in neuGRID.

The remainder of this report recalls elements of deliverable D7.1 (in particular section 2) in order to give clarity to the following sections. The grid middleware and neuGRID infrastructure are therefore accurately introduced and followed by a concrete list of software packages migrations, which were operated over the last 3 years. A very exhaustive test of the new CREAM CE service was also done but this later is not yet installed in production due to some functionality that are not implemented yet and that are needed by some neuGRID components.

Note: To improve readers' understanding of the document and technical terms, "Bibliography" and "Glossary" sections are provided where pointers to more detailed information can be found as well as acronyms definitions as used throughout the text.

2 The EGEE gLite Grid Middleware

2.1 Overview

Grid Middleware refers to the security, resource management, data access, instrumentation, policy, accounting, and other services required for applications, users, and resource providers to operate effectively in a grid environment. Middleware acts as a sort of 'glue' which binds these services together. Grid middleware is built by layered interacting packages.

- A grid middleware is an internet based system that needs efficient and reliable communication and is a blend of high performance systems and high throughput computing,
- A grid middleware is data aware and all data access and replications decisions are based onbase bound at least for the following functions: grid topology management user access and certification dataset locations and replicas resource definition and dynamical management performance and user bookkeeping,
- A grid middleware is bound to efficient matching and scheduling algorithms to find best available resources for the task execution and resource brokering,
- A grid middleware depends on accurate clock performances to synchronize nodes and correctly handle task and job scheduling.

The gLite distribution [1] is an integrated set of components designed to enable resource sharing. In other words, this is a middleware for building a grid. It is developed by the EGEE European project. gLite distributions pull together contributions from many projects/partners, including (but not limited to) LCG and VDT. The distribution model is to construct different services ('node-types') from these components and then to ensure simple installation and configuration on the chosen platforms (i.e. currently Scientific Linux version 4).

The gLite Middleware is a quite complex framework which follows a Service Oriented Architecture (SOA) to facilitate interoperability among grid services and to allow easier compliance with upcoming standards. This architecture has the advantage to not be bound to specific implementations, and services are expected to work together but can also be used independently.

2.2 Middleware Architecture

This section gives a short overview of the components currently available in the EGEE gLite grid middleware. Figure 2-1 below depicts the gLite high level services, which can thematically be grouped into five service families: Access Services, Information and Monitoring Services, Job Management Services, Data Management Services and Security Related Services. For more detailed descriptions refer to [5].

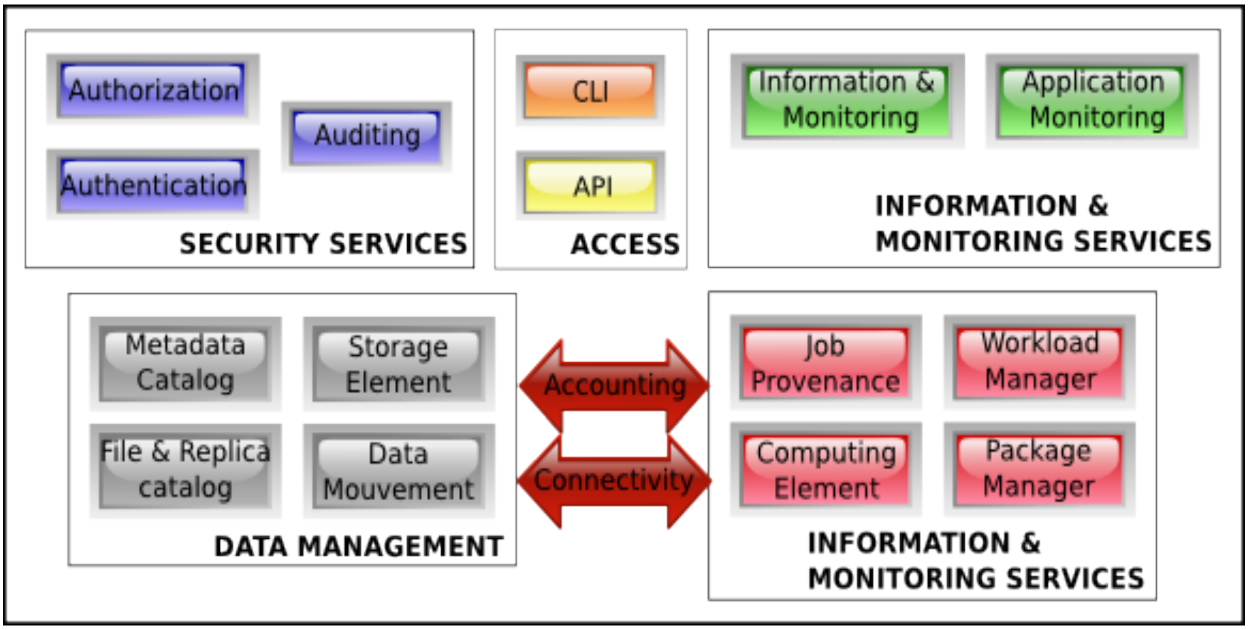

Figure 2-1: The gLite high level architecture

The following subsections briefly present the gLite grid middleware components grouped by service types. Only major gLite components considered to be relevant to the scope of the neuGRID project are listed and described.

2.2.1 Access Services

The aim of the Access Services is to give either command line or programmatic level access to the whole stack of publicly available gLite services. As an example, Figure 2-2 below presents a schematic view of gLite Data Management APIs and Command Line Interfaces (CLIs).

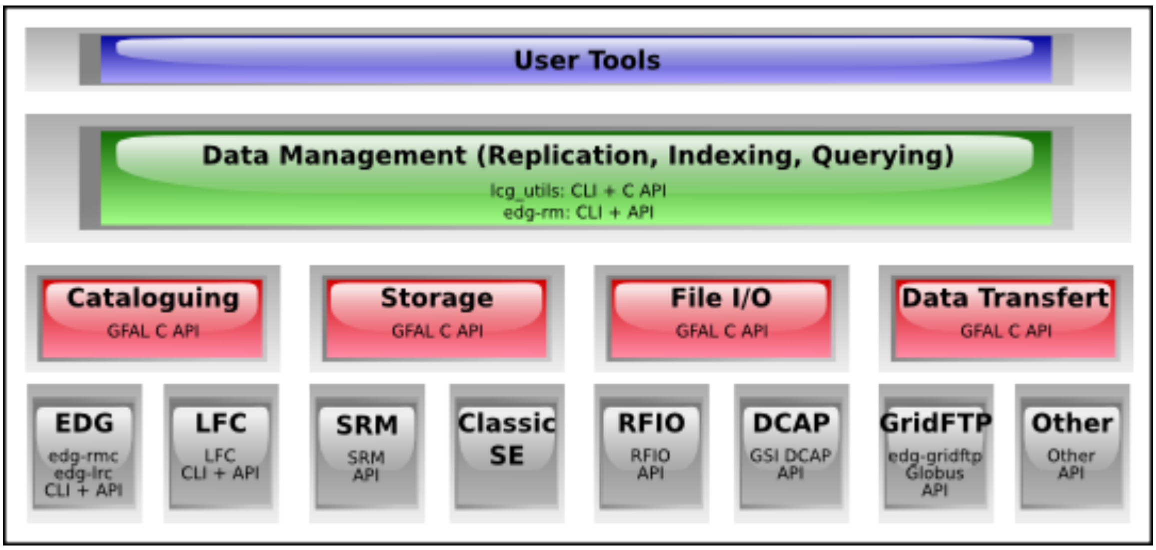

Figure 2-2: gLite Data Management APIs and CLIs

User Interface (UI) is gLite's separate deployment component which belongs to Access Services group. When installed on a server or desktop computer it allows users or user level applications to access the functionalities of the grid services like VOMS, WMS, File Catalog, SE, CE, IS etc. It provides a set of CLIs and different programming languages APIs.

2.2.2 Security Services

Security Services encompass the Authentication, Authorization, and Auditing services which enable the identification of entities of different nature (i.e. users, systems, and services), allow or deny access to services and resources, and provide information for post-mortem analysis of security related events. To carry out the tasks of Authentication and Authorization, gLite uses the Public Key Infrastructure (PKI) x509 technology using Certificate Authorities (CAs) as trusted third parties and MyProxy [6] extended by VOMS. It also provides functionality for data confidentiality and a dynamic connectivity service (i.e. means for a site to control network access patterns of applications and grid services utilizing its resources).

VOMS: The Virtual Organization Membership Service (VOMS) is a service to manage authorization information in a VO scope. The VOMS system should be used to include VO membership and any related authorization information in a user's proxy certificate. These proxies will be said to have VOMS extensions. The user gives the voms-proxy-init command instead of grid-proxy-init, and a VOMS server will be contacted to check the user's certificate and create a proxy certificate with VOMS information included. By using that certificate, the VO of a user will be present in every action that he will perform.

Hydra: is a gLite implementation of secure key storage. Symmetric encryption keys for encrypted files are stored in a specific set of servers called Hydra. Hydra provides controlled access to these keys (through certificate DN and VOMS attributes based ACLs) and secured communication to the requester. Hydra uses Shamir's secret-sharing scheme for splitting keys into 'n' fragments stored in different places. Only 'm<n' fragments are needed to reconstruct a complete key. However, owning less than 'm' key fragments, does not give any information on the complete key. Thus, the system is both resistant to attacks (at least 'm' key stores need to be compromised for an attacker to be able to reconstruct the key) and reliable (the disconnection of a limited number of servers does not prevent the key reconstruction).

Encrypted storage: Users access Hydra through the Encrypted storage C library which provides on-the-fly; block level data encryption and decryption. The component provides command line utilities for managing the keys in the Hydra key store. There are also command-line utilities, which integrate the library with the gLite I/O clients, thus one can retrieve/decrypt or store/encrypt files transparently.

Grid Policy BOX (G-PBox): is a policy framework designed to operate on Grid environments. G-PBox is a tool for VO and site administrators. For VO administrators it allows writing policies for internal VO groups/roles defined in the VO VOMS server and to manage policies received from site G-PBoxes. In turn, by site administrators G-PBox is used write policies for internal sites users and to manage policies received from VO G-PBoxes. G-PBox is queried by resources like CEs and SEs and services (as WMS) also not owned by VOs or Sites. Simply speaking, it helps in creation and application of authentication policies between grid services - for example, between VO WMSes and site CEs and SEs.

MyProxy (PX)[1]: On grids, users authenticate themselves using temporary credentials called proxy certificates, which contain also the corresponding private key. Proxy certificates do not represent a significant security risk only if they are reasonably short-lived (by default, a dozen hours). For longer jobs, PX plays a role of online credential repository [6]. For such long running jobs a proxy renewal system is used, consisting of a **Proxy Renewal Service (PRS)** on the RB and a PX server on a dedicated host. A PX stores long-lived user proxies (with a lifetime of several days, usually) which it uses to generate, on request of the PRS, short-lived proxies for jobs whose

[1] in gLite this service comes with VDT [7] distribution

proxies are about to expire.

2.2.3 Information and Monitoring Services

- **Information and Monitoring Services (IMS)** provide a mechanism to publish and consume information and to use it for monitoring purposes. The information and monitoring system can be used directly to publish, for example, information concerning the resources on the Grid. More specialized services, such as the Job Monitoring Service and Network Performance Monitoring services can be built on top. Major components of gLite IMS are: BDII, R-GMA and Service Discovery.

- **BDII:** The Information System (IS) provides information about the status of Grid services and available resources. Job and data management services publish their status through Grid Resource Information Server (GRIS). GRIS runs on every service node and is implemented using OpenLDAP, an open source implementation of the Lightweight Directory Access Protocol (LDAP). Ever grid site also runs one Grid Index Information Server (GIIS). The GIIS queries the service GRISes on the site and acts as a cache storing information about all available site services. Finally, the information from GIISes is collected by Berkeley Database Information Index (BDII). The BDII (also called top BDII: tBDII) queries the GIISes (sometimes also called site BDII: sBDII) and acts as a cache storing information about all available Grid services in its database. Figure 2-3 shows this flow of information. Users and programs interested in status of the Grid usually query the top level BDII as it contains information about all the services that are available.

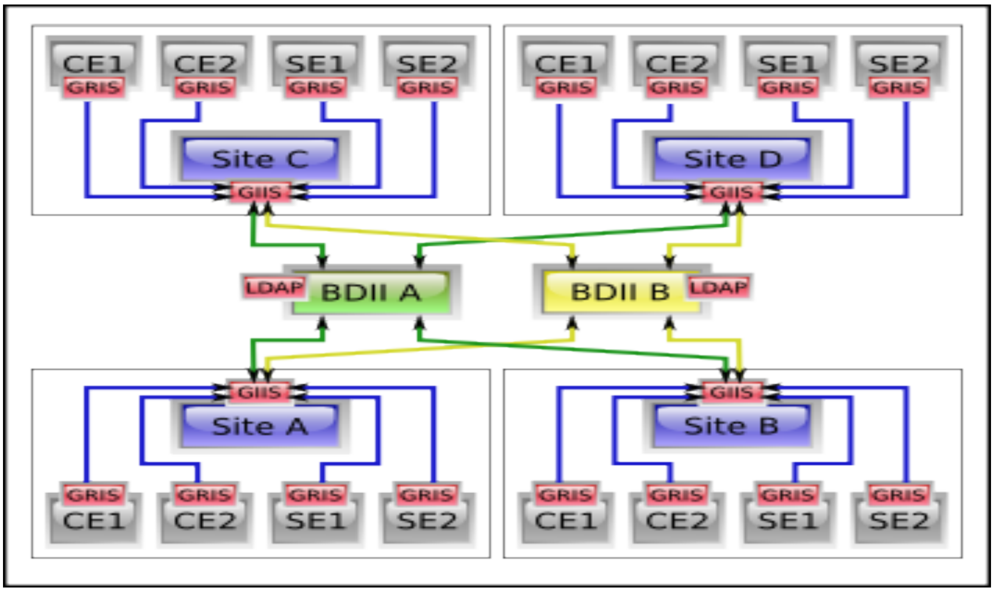

Figure 2-3: gLite Information System hierarchy

- **R-GMA** is a relational implementation of the Grid Monitoring Architecture defined by GGF. It is information and monitoring system for use both by the Grid middleware and by applications. Producer and consumer services are available at every site and currently make use of a single registry and schema service. R-GMA is currently being used by job management services, operational tools and by users for monitoring their applications.

- **Service Discovery** is a facility for locating suitable services offered to both end users and other services. It is implemented as a client library front-end to one or more information systems. The information systems are made available by a plug-in mechanism. It is intended to be lightweight and simple to use. Currently it supports BDII, R-GMA and XML files as backends. Service Discovery is used by Workload and Data Management components. This component is also available on UI for users do discover services currently available on an infrastructure.

- Grid services provide information about their status in a form defined by the Grid Laboratory for a Uniform Environment (GLUE) schema. GLUE schema is the result of an ad-hoc international collaboration. It is maintained in two forms: one for BDII and one for R-GMA. The information is logically the same but one follows a hierarchical schema (for BDII) and the other is relational (for R-GMA).

2.2.4 Data Management Services

Architecturally speaking, the Data Management Services of the gLite middleware stack consist of three major components: Data Storage, Metadata and Catalog Services and Data Scheduling.

2.2.4.1 Data Storage

- **Grid File Transfer Protocol (GridFTP)** is a high-performance, secure, reliable data transfer protocol optimized for high-bandwidth wide-area networks. It is based on the Internet FTP protocol, and it implements extensions for high-performance operation. GridFTP uses basic Grid security on both control (command) and data channels. Other features include multiple data channels for parallel transfers, partial file transfers, third-party (direct server-to-server) transfers, reusable data channels, and command pipelining. GridFTP is used as a primary data transfer interface to Storage Elements.

- **Storage Resource Managers (SRM)** is a service somewhat similar to the cluster batch system but instead of managing processors and jobs it manages requests for storage space and files. The storage space managed can be disk space, tape space or a combination of the two. There is a number of SRM implementations for disk storage management that are widely deployed (DPM, dCache, CASTOR).

- **Disk Pool Manager (DPM)** is a recommended solution for lightweight deployment of smaller sites because it is easy to install and requires very low maintenance effort. It features full implementation of SRM. Bigger sites usually choose dCache because of robustness, scalability and advanced features. CERN Advanced STORage (CASTOR) is an implementation used by sites that have both disk and tape storage.

2.2.4.2 Metadata and Catalog Services

- **LCG File Catalog (LFC)** offers a hierarchical view of files to users, with a UNIX-like client interface. The LFC provides Logical File Name (LFN) to Storage URL (SURL) mappings and authorization for file access. The LFNs are aliases created by a user to refer to actual data. Simple metadata can be associated to them. The authorization is performed using UNIX-style permissions, POSIX Access Control Lists (ACL) and VOMS support. The LFC uses a client-server model with a proprietary protocol. LFC server communicates with a database (either Oracle or MySQL), where all the data is stored. LFC catalogue also exposes a Data Location Interface (DLI) - a web service used by applications and Resource Brokers. Provided with a LFN, the DLI returns the actual location of the file replicas.

2.2.4.3 Data Scheduling

- **File Transfer service (FTS)** is a reliable, low-level data movement service for transferring files between Storage Elements. It also provides features for administration and monitoring these transfers. The FTS exposes an interface to submit asynchronous bulk requests and performs the transfers using either third-party GridFTP or SRM Copy.

2.2.5 Job Management Services

Three major components constituting the Job Management Services group are Computing Element, Workload Management and Accounting. Although primarily related to the job management services; accounting is a special case as it will eventually take into account not only computing, but also storage and network resources.

- **Computing Element (CE)** is the service representing a computing resource. It provides a virtualization of the computing resource localized at a site (typically a batch queue of a cluster but also supercomputers or even single workstations). It provides information about the underlying resource and offers a common interface to submit and manage jobs on the resource. CE includes: a Grid Gate (GG) - Gatekeeper for CE based on Globus - which acts as a generic interface to the cluster; LRMS (sometimes called batch system); the cluster itself - a collection of Worker Nodes (WN) or just one multiprocessor WN, the node(s) where the jobs are run. There are two GG implementations in gLite 3.0: the LCG-CE and the gLite-CE; sites can choose what to install. The GG is responsible for accepting jobs and dispatching them for execution on the WN(s) via the LRMS.
  Another type of CE developed in EGEE project is CREAM (Computing Resource Execution And Management). It is a simple, lightweight service that implements all the operations required at the CE level. Its interface is defined using WSDL. The service is compliant with the existing BES standard. CREAM can be used by the WMS, via the ICE component (see next description of WMS), or by a generic client, e.g. an end-user willing to directly submit jobs to a CREAM CE. A C++ command line interface and Java clients are available for this purpose.
  The interface of gLite-CE and CREAM with the underlying LRMS is implemented via BLAH. All the resource management systems supported by BLAH are automatically supported by the CEs. In gLite 3.0 the supported LRMS types are OpenPBS, LSF, Maui/Torque, BQS and Condor.

- **Workload Management System (WMS)** is a Grid level meta-scheduler that schedules jobs on the available CEs according to user preferences and several policies. It also keeps track of the jobs it manages in a consistent way. The core component of the WMS is the Workload Manager (WM), whose purpose is to accept and satisfy requests for job management coming from its clients (i.e., computational job submission). In particular the meaning of the submission request is to pass the responsibility of the job to the WM. The WM will then pass the job to an appropriate Computing Element for execution, taking into account the requirements and the preferences expressed in the job description. The decision of which resource should be used is the outcome of a matchmaking process between submission requests and available resources. The Resource Broker (RB) or Matchmaker as WMS component offers support to the WM in taking the above mentioned decision. It provides a matchmaking service based on a given user's job description – find a resource that best match the request. A WMS instance interacts with several other services. Tracking job lifetime relies on the Logging and Bookkeeping Service. Information on service availability, resource status and data localization is gathered from appropriate sources, such as Service Discovery, LFC, BDII, RGMA. Security-related aspects are addressed interacting with VOMS, Proxy Renewal and G-PBox.

Another important component of the gLite WMS is the Interface to CREAM Environment (ICE). It provides the connection between the gLite WMS and the CREAM CE. ICE, running in the gLite WMS node along with the other processes of the gLite WMS, receives job submissions and other job management requests from the WM component of the WMS and then invokes the appropriate CREAM methods to perform the requested operation.

- **Logging and Bookkeeping Service (LB):** The primary purpose of the Logging and Bookkeeping service (LB) is tracking Grid jobs as they are processed by various Grid middleware components. It collects and stores in a database the job status information supplied by the different components of the WMS system. The collection is done by LB local-loggers, which run on the RB and on the CE, while the LB server, which normally runs on the RB, saves the collected information in the database. The database can be queried by the user from the UI, and by RB services. The information that is gathered in LB is used to inform Grid users on the job state. This information is also useful for example for debugging of user jobs.

3 The neuGRID Infrastructure

To support neuGRID software development cycles, experimentation with specific gLite middleware deployments, developers' and infrastructure administrators' trainings, 2 sub-infrastructures were identified and setup:

- The PoC (Proof-of-Concept), being the place for
  - evaluation of new gLite releases,
  - experimentation of gLite deployment strategies,
  - development and early testing of neuGRID services,
  - training of neuGRID developers and infrastructure administrators,
  - pipelines and client applications development and integration.

- The PROD (Production) – stable and secure environment for end-users, being the place for
  - operational platform services destined to neuGRID end-users.

PoC (WP7 responsibility), PROD (WP8 responsibility) are two distinct non-overlapping sub-infrastructures forming the neuGRID network.

The Figure 3-1 below schematically represents them and also shows grid middleware evaluation/validation/deployment and software development cycles.

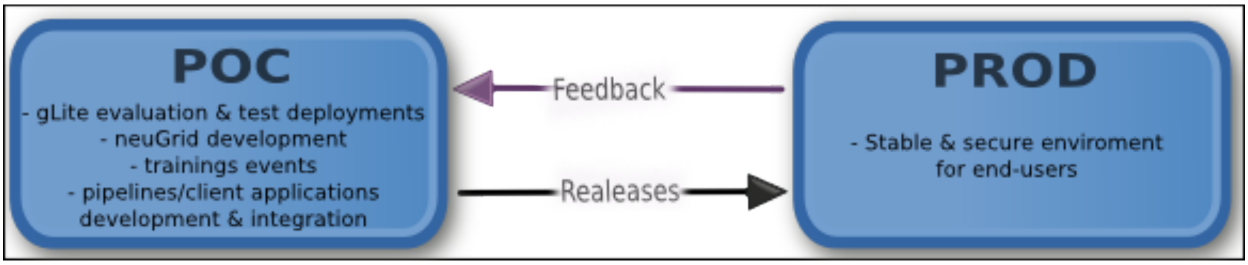

Figure 3-1: neuGRID infrastructure (schematic view)

3.1 Certificate Authority (CA)

On grids, public key infrastructure (PKI) arrangements are used for binding electronic cryptographic keys (key-pair) with respective user identity. In this scheme Certification Authority (CA) has a role of trusted third party and of high importance. It plays a vital role on the infrastructure while not directly being a part of the grid infrastructure.

WP7 has therefore considered 3 possibilities for obtaining cryptographic certificates:

- buying them from widely accepted CA like for example VeriSign [8],
- setting up an instance of CA belonging to neuGRID project only, and/or
- obtaining them from the eu-gridpma initiative in Europe.

In order to be more flexible, it was decided that in the PoC infrastructure a specific neuGRID CA would be used for development purposes (thus not incurring any additional costs). This CA is provided by MAAT (based on the Open Source free software OpenCA). However, in the PROD infrastructure, all eu-gridpma CAs are accepted.

3.2 Infrastructure Services

Based on gathered requirements and from the technical partners' experience, a list of gLite services which were needed in neuGRID was initially determined and progressively enriched from received feedback. As a consequence, the following grid services have been selected, tested and deployed:

- **VOMS:** to manage the neuGRID community.
- **MyProxy** to handle long running jobs.
- **LFC** to reference all files stored in the grid.
- **DPM** to store the imaging data and other physical files.
- **tBDII** and **sBDII** services for the grid information system.
- **CE/WN** and **WMS/LB** services to schedule and run jobs in the grid.

VOMS, MyProxy, LFC, tBDII and WMS/LB are part of the Grid Coordination Center (i.e. GCC) as core services of the neuGRID infrastructure. Indeed, they can be seen as gLite core services under which all other neuGRID services can be added/mixed. The VOMS server is shared by the 2 sub-infrastructures (i.e. PoC and PROD) as it is able to deal with sub-groups. This is also the case for MyProxy, as it is only a proxy certificate repository into which users can dump their delegated credentials.

Concerning LFC, tBDII and WMS/LB, there is one instance of each service per sub-infrastructure in order to avoid all potential resource sharing problems and to ease users' access rights management.

sBDII, DPM and CE/WN were installed on each of the Data Archiving and Computing Sites (DACS) to provide them with data storage and computing power capacity. This approach is considered to be a standard practice for gLite middleware deployment.

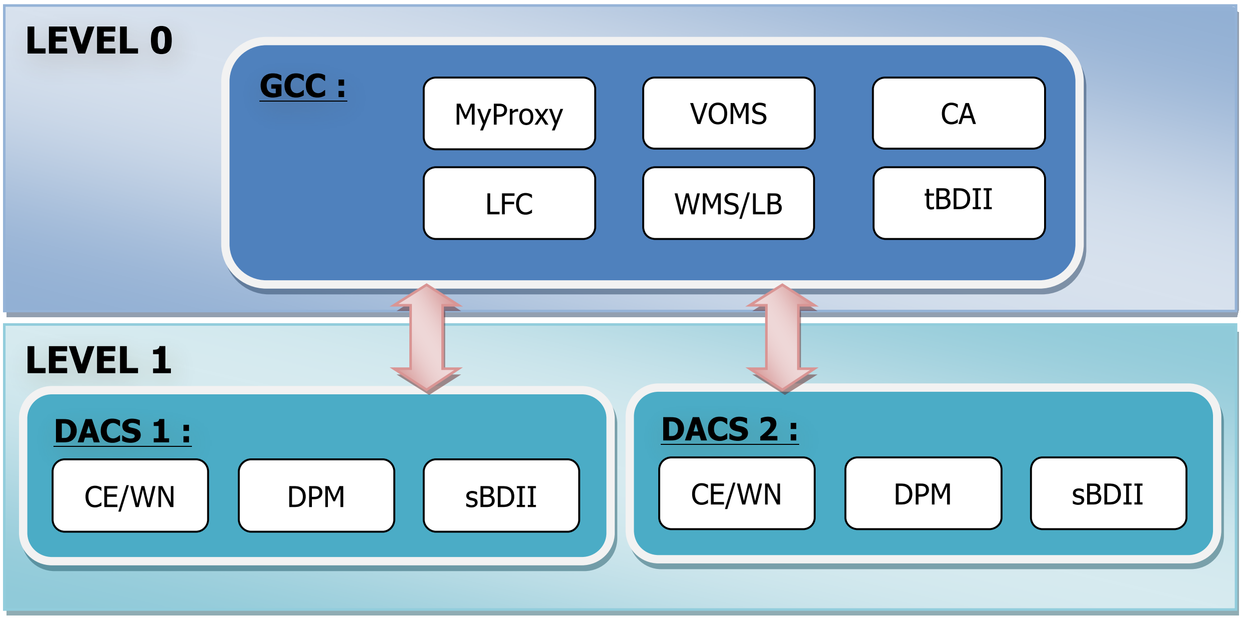

Figure 3-2: gLite Services Repartition Inside of POC

As it is presented in the project description of work document, "LEVEL 0" represents the "infrastructure ground truth" level, made of infrastructure core services, whereas "LEVEL 1" represents the Data Archiving and Computing Sites (DACS) hosting the physical resources used to store and compute data.

Two partners are providing the core services of the neuGRID infrastructure, respectively:

- **MAAT** which owns the GCC and one development DACS (for testing purposes),
- **PRODEMA** which owns the Data Coordination Center (DCC).

For the GCC, the deployed services are as follows:

HOSTNAME/IP	SERVICES
voms.maatg.eu	VOMS
lfc.maatg.eu	LFC
bdii.maatg.eu	tBDII
myproxy.maatg.eu	MyProxy
wms.maatg.eu	WMS + LB
openca.ng-maat-server1.maatg.eu	CA

For the development DACS made available by MAAT, the deployed services are as follows:

HOSTNAME/IP	SERVICES
ng-maat-server3.maatg.eu	DPM
bdii-site.maatg.eu	sBDII
ng-maat-server9. maatg.eu	CREAM CE
ng-maat-server8. maatg.eu	WN
ng-maat-server11. maatg.eu	WN
ng-maat-server12. maatg.eu	WN

4 Grid Middleware Migrations and Test Report

As formerly presented, there are two environments in neuGRID, the so-called PoC (Proof-of-Concept) and PROD (Production). This section describes the simple yet efficient procedures and tests which have been performed in the infrastructure before and after every updates.

gLite updates can be of different types:

- Regular bug fixes: Some functional bugs are corrected in the different components of the system,
- Security bug fixes: Some security threats were corrected in one or more components.
- Functionality improvements: New features were integrated in the components

An update can be a mix of these 3.

Before applying software patches, it is important to decide if the available updates are suitable to the neuGRID infrastructure, as well as the potential consequences. One responsibility of WP7 is to determine which updates should be installed in the infrastructure or not. The adopted policy is as follows:

- All the security bug fixes MUST be installed,
- For other bug fixes, the infrastructure is updated if affected by the concerned bug or if there is a benefit in correcting it,
- The same applies for functionality improvements. If the service improvement can somehow improve the neuGRID infrastructure performing, it is applied.

Before updating a service and to precisely understand, the corresponding information is extracted from the official gLite web site (i.e. http://glite.web.cern.ch/glite/packages/R3.1). Updates release notes can be found at this website together with the installation instructions. The most important informations needed for each update are:

- Description: A brief description of the changes included in the update, indicating files affected and corresponding comments,
- Updated RPMs: rpm packages updated with this new version (interesting in case of software package rollback),
- Service reconfiguration after update: which indicates whether it is necessary to reconfigure the service after the RPM update. In this section can also be found all the necessary manipulations that have to be performed to restore the service once updated,
- Service restart after update: indicates whether it is necessary to restart the service after the update,
- How to apply the fix: provides indications on how to apply the updates.

Updating a service within the infrastructure always incurs a risk. Indeed, the update can break the current installations by adding new problems/complexity such as version inconsistencies, dependencies etc. This is the reason why updates must be carefully selected, applied and tested within the PoC environment before entering production.

4.1 Update Selection and Test Procedure

- **Installation of the gLite Middleware**

In this stage the most important thing is to ensure that all the gLite updates released by EGEE work fine with the neuGRID system, and that there are no incompatible software packages, in particular with those of the servers' Operating Systems (OS). At this stage, updates are installed simply using the yum tool as provided by the OS, which connects to the software repository at CERN and detects the available updates necessary to maintain the infrastructure up-to-date.

- **Configuration of the gLite Middleware**

Once the new version of the software is installed, most of the time, a reconfiguration is needed. As gLite middleware is constantly evolving, new features may be included, in some cases extra variables are needed to add or remove, to properly reconfigure a working service. This configuration is performed using the Yaim tool. This utility is bundled with gLite to easily configure the middleware services. More information about how to reconfigure gLite services can be found at gLite middleware reference cards at CERN twiki web page at:

https://twiki.cern.ch/twiki/bin/view/EGEE/ServiceReferenceCards

- **Configuration Tests**

Once updated it is mandatory to run a set of tests to ensure that the grid is running properly. This set of test is the one delivered by WP11 (in deliverable D11.1) that ensures that the gLite middleware is performing correctly.

As a consequence these tests, applicable updates are then forwarded to WP8, which propagates them to the production environment.

4.2 Middleware Updates

4.2.1 Regular Updates

In the following paragraphs can be found the different updates that were applied in the infrastructure over the last 3 years. Updates are classified by middleware services.

4.2.1.1 lcg-CE

Version	Release Date	Description
3.1.40-0	16.03.2010	Corrected BUGS
3.1.39-0	18.02.2010	Corrected BUGS
3.1.38-0	11.01.2010	Security
3.1.37-0	28.10.2009	Updated Functionality
3.1.35-0	31.08.2009	Corrected BUGS
3.1.34-0	24.08.2009	Fixes know issues with LB
3.1.33-0	29.07.2009	Security
3.1.32-0	06.07.2009	Corrected BUGS
3.1.31-0	24.08.2009	Updated and Improved Functionality
3.1.30-0	16.06.2009	Corrected BUGS
3.1.29-0	11.05.2009	Corrected BUGS

4.2.1.2 glite-CREAM

Version	Release Date	Description
3.2.8-2.sl5	10.11.2010	Corrected BUGS
3.2.7-2.sl5	18.08.2010	Corrected BUGS
3.2.5-0.sl5	03.05.2010	Corrected BUGS
3.2.2-0	08.02.2010	Updated Functionality (GLITE 3.2 MIGRATION)
3.1.20-0	06.10.2009	Security

3.1.19-0	31.08.2009	Updated and Improved Functionality
3.1.18-0	24.08.2009	Updated and Improved Functionality
3.1.17-0	29.07.2009	Security
3.1.16-0	06.07.2009	Updated and Improved Functionality
3.1.14-0	24.06.2009	Updated and Improved Functionality
3.1.13-0	11.05.2009	Corrected BUGS

4.2.1.3 gLite Worker Node (x86 platform) (Obsolete in neuGRID beginning of 2010)

Version	Release Date	Description
3.1.40-0	05.11.2009	Corrected BUGS
3.1.39-0	28.10.2009	Updated and Improved Functionality
3.1.37-0	22.09.2009	Corrected BUGS
3.1.36-0	31.08.2009	Updated and Improved Functionality
3.1.35-0	24.08.2009	Updated and Improved Functionality
3.1.34-0	29.07.2009	Security
3.1.33-0	15.07.2009	Corrected BUGS
3.1.32-0	14.07.2009	Updated and Improved Functionality
3.1.31-0	24.06.2009	Updated and Improved Functionality
3.1.30-0	16.06.2009	Corrected Minor BUG fixes
3.1.29-0	11.05.2009	Corrected BUGS

4.2.1.4 gLite Worker Node (x86_64 platform)

4.2.1.4.1 glite 3.1

Version	Release Date	Description
3.1.30-0	16.03.2010	Corrected BUGS
3.1.29-0	18.02.2010	Corrected BUGS
3.1.28-0	11.01.2010	Security
3.1.27-0	05.11.2009	Corrected BUGS
3.1.26-0	28.10.2009	Updated and Improved Functionality
3.1.24-0	22.09.2009	Corrected BUGS
3.1.23-0	31.08.2009	Updated and Improved Functionality
3.1.22-0	24.08.2009	Updated and Improved Functionality
3.1.21-0	29.07.2009	Security
3.1.20-0	15.07.2009	Corrected BUGS
3.1.19-0	14.07.2009	Updated and Improved Functionality
3.1.18-0	25.06.2009	Updated and Improved Functionality
3.1.17-0	16.06.2009	Corrected Minor BUG fixes
3.1.16-0	11.05.2009	Corrected BUGS

4.2.1.4.2 glite 3.2

Version	Release Date	Description
3.2.9-0.sl5	04.08.2010	Corrected BUGS
3.2.7-0	24.03.2010	Corrected BUGS
3.2.6-0	08.02.2010	Corrected BUGS

4.2.1.5 glite-BDII

Version	Release Date	Description
3.2.9-0.sl5	26.08.2010	Corrected BUGS

3.2.8-0.sl5	27.04.2010	Corrected BUGS
3.2.6-0	24.03.2010	Corrected BUGS
3.2.5-0	08.02.2010	Corrected BUGS (GLITE 3.2 MIGRATION)
3.1.19-0	22.10.2009	Rolled-back
3.1.18-0	31.08.2009	Corrected BUGS
3.1.17-0	06.07.2009	Corrected Minor BUGS

4.2.1.6 gLite Workload Management System (WMS)

Version	Release Date	Description
3.1.30-0.slc4	16.11.2010	Updated and Improved Functionality
3.1.29-0.slc4	05.07.2010	Updated and Improved Functionality
3.1.27-0	16.03.2010	Updated and Improved Functionality
3.1.26-0	18.02.2010	Updated and Improved Functionality
3.1.25-0	11.01.2010	Security
3.1.24-0	28.10.2009	Updated and Improved Functionality
3.1.22-0	22.09.2009	Updated and Improved Functionality
3.1.21-0	31.08.2009	Updated and Improved Functionality
3.1.20-0	24.08.2009	Updated and Improved Functionality
3.1.19-0	29.07.2009	Security
3.1.18-0	06.07.2009	Updated and Improved Functionality
3.1.17-0	24.06.2009	Updated and Improved Functionality
3.1.15-0	11.05.2009	Security

4.2.1.7 gLite Logging and Bookkeeping Service (LB)

Version	Release Date	Description
3.1.19-0	16.03.2010	Updated and Improved Functionality
3.1.18-0	16.03.2010	Updated and Improved Functionality
3.1.16-0	31.08.2009	Updated and Improved Functionality
3.1.15-0	24.08.2009	Updated and Improved Functionality
3.1.14-0	06.07.2009	Corrected BUGS
3.1.12-0	24.06.2009	Updated and Improved Functionality
3.1.11-0	11.05.2009	Corrected BUGS

4.2.1.8 gLite LCG File Catalog (LFC) for MySQL

Version	Release Date	Description
3.2.7-2.sl5	21.07.2010	Updated and Improved Functionality
3.2.6-1.sl5	30.06.2010	Updated and Improved Functionality
3.2.5-0	24.03.2010	Updated and Improved Functionality
3.2.4-0	08.02.2010	Corrected BUGS (GLITE 3.2 MIGRATION)
3.1.32-0	11.01.2010	Security
3.1.31-0	28.10.2009	Updated and Improved Functionality
3.1.29-0	31.08.2009	Updated and Improved Functionality
3.1.28-0	29.07.2009	Security
3.1.27-0	15.07.2009	Updated and Improved Functionality
3.1.26-0	06.07.2009	Corrected BUGS
3.1.25-0	25.06.2009	Updated and Improved Functionality
3.1.24-0	11.05.2009	Updated and Improved Functionality

4.2.1.9 gLite Disk Pool Manager (DPM) Storage Element (SE) for MySQL

Version	Release Date	Description
3.2.7-2.sl5	21.07.2010	Updated and Improved Functionality
3.2.6-2.sl5	30.06.2010	Updated and Improved Functionality
3.2.5-0	24.03.2010	Updated and Improved Functionality
3.2.4-0	08.02.2010	Improved Functionality (GLITE 3.2 MIGRATION)
3.1.32-0	28.10.2009	Updated and Improved Functionality
3.1.30-0	31.08.2009	Updated and Improved Functionality
3.1.29-0	29.07.2009	Security
3.1.28-0	15.07.2009	Updated and Improved Functionality
3.1.27-0	06.07.2009	Corrected BUGS
3.1.26-0	25.06.2009	Updated and Improved Functionality
3.1.25-0	16.06.2009	Updated and Improved Functionality
3.1.24-0	11.05.2009	Updated and Improved Functionality

4.2.1.10 gLite TORQUE Clients

Version	Release Date	Description
3.2.6-0	08.02.2010	Improved Functionality (GLITE 3.2 MIGRATION)
3.1.7-0	31.08.2009	Updated and Improved Functionality

4.2.1.11 gLite TORQUE Server

Version	Release Date	Description
3.2.2-0	08.02.2010	Improved Functionality (GLITE 3.2 MIGRATION)
3.1.7-0	31.08.2009	Updated and Improved Functionality

4.2.1.12 gLite TORQUE Utils

Version	Release Date	Description
3.2.2-0	08.02.2010	Improved Functionality (GLITE 3.2 MIGRATION)
3.1.7-0	31.08.2009	Updated and Improved Functionality

4.2.1.13 gLite VOMS Server (MySQL Backend Version)

Version	Release Date	Description
3.1.27-0	16.03.2010	Updated and Improved Functionality
3.1.26-0	11.01.2010	Updated and Improved Functionality
3.1.25-0	28.10.2009	Updated and Improved Functionality
3.1.23-0	29.07.2009	Security
3.1.22-0	06.07.2009	Corrected BUGS
3.1.21-0	25.06.2009	Updated and Improved Functionality
3.1.20-0	11.05.2009	Corrected BUGS

4.2.2 Specific Test: The CREAM CE Service

The neuGRID project uses intensively the computational dimension of the grid. In production, the regular Computing Element (CE) service was installed, as this is a production ready service from the gLite middleware which was already thoroughly tested and used across several communities.

However and since a couple of months, a new service was released in production named CREAM CE (which stands for Computing Resource Execution And Management). The main difference between the regular gLite CE and the new CREAM CE is that the latter can be accessed directly through a well defined web service interface and is much more performant. Of course, CREAM CE can also be accessed through the regular WMS scheduler via the ICE plug-in.

In order to measure potential reactivity and ease of use improvements, WP7 therefore decided to test it. Indeed, CREAM CE uses a push notification mechanism to report on events to WMS, whereas the regular CE uses a push mechanism, which has a direct impact on user-perceived performances.

As a consequence, ng-maat-server9.maat-g.com server from the PoC environment was migrated to CREAM CE and different tests were applied, as listed below:

- *Direct job submission of 1 simple job*: one job was directly submitted to the CREAM service (direct submission is a new functionality introduced by CREAM CE). In this configuration the functionality improvement was quite impressive as almost no overhead was noticed. Indeed, for a simple job that took 10 seconds of actual execution, using the regular gLite CE it took approximately 4 to 5 minutes to see the real job completion (as it takes a long time to report through the gLite CE/WMS architecture). In the case of CREAM CE, the job status was nearly reported in real time to the Information System.

- *Direct job submission of 1000 simple jobs*: This test was useful to appreciate that the CREAM service is very efficient even when overloaded. The memory usage is well managed in the latest version (Note: this is also the case with the gLite CE).

- *Submission of 1 job through WMS/ICE*: Same here, the difference between a gLite CE and CREAM is significant. It is however less reactive than when submitting directly to the CREAM service but instead of having 4 to 5 minutes, the response time is turned down to 15 seconds. Of course, this performance heavily depends on the load of WMS.

- *Submission of 1000 jobs through WMS/ICE*: This was operated to test the ICE plug-in. The first test was not a complete success because ICE uses a lot of memory, compared to a simple WMS usage. But this problem was corrected in the last version and now the memory usage is reasonable.

In conclusion to these preliminary tests, the new CREAM CE service reveals very efficient and performant, thus justifying it to become soon a candidate for migration to production. However, end-users of neuGRID need to be able to submit more complex jobs such as Direct Acyclic Graph (DAG). Unfortunately, it appears that CREAM CE is not able yet to interpret such DAGs descriptions even when submitting them through the WMS scheduler. Thus, until this requirement is met, CREAM CE cannot be used in production. WP7 will therefore keep track of up-coming CREAM CE developments until it becomes DAG aware and thus of interest for production.

5 Conclusion

As a conclusion, the adopted selection and test approach in neuGRID WP7 allowed a rapid identification and validation of necessary grid middleware components as well as their deployment from the PoC to the Production environments, in collaboration with WP8.

As a consequence the test-bed was ready to host development tests very early in the project work plan and has been the place for a number of successful experiments, while the Production environment has been operational since the deployment of its first DACS site. Both the PoC and Prod nowadays expose the latest stable versions of the middleware services.

The separation of concerns introduced by the PoC environment has allowed developers testing their latest products while securing a stable production grid for end-users. The latter has demonstrated usefulness during the AC/DC2 data challenge execution, in WP11.

On a more general note, WP7 has been and still is following strictly the releases and recommendations as provided by EGEE before and while planning for migrations. In that sense, WP7 heavily relies on the relevance and robustness of processes at set out over the last 6 years within EGEE.

6 List of Abbreviations

API	Application Program Interface
BDII	Berkeley Database Information Index
BES	Basic Execution Services
BLAH	Batch Local ASCII Helper Protocol
CA	Certification Authority
CASTOR	CERN Advanced STORage Manager
CE	Computing Element
CEMon	Computing Element MONitor
CLI	Command Line Interface
CREAM	Computing Resource Execution And Management
DGAS	Distributed Grid Accounting System
DLI	Data Location Interface
DN	Distinguished Name
DPM	Disk Pool Manager
EGEE	Enabling Grids for E-sciencE
FTS	File Transfer Service
GFAL	Grid File Access Library
GG	Grid Gate
GIIS	Grid Index Information Server
gLite	EGEE Grid middleware stack
GLUE	Grid Laboratory for a Uniform Environment
GMA	Grid Monitoring Architecture
GRIS	Grid Resource Information Server
ICE	Interface to CREAM Environment
IS	Information System (grid-level)
LSF	Local Sharing Facility
LB	Logging and Bookkeeping
LFN	Logical File Name
MDS	Monitoring and Discovery Service
neuGRID platform	neuGRID services + gLite middleware
PBS	Portable Batch System
PKI	Public Key Infrastructure
POC	neuGRID Proof Of Concept sub-infrastructure – *neuGRID test-bed*
PROD	neuGRID Production sub-infrastructure
R-GMA	Relational Grid Monitoring Architecture
RFIO	Remote File Input/Output
RB	Resource Broker
SAM	Service Availability Monitoring framework
SD	Service Discovery
SE	Storage Element

SL	Scientific Linux
SL4	Scientific Linux 4
SRM	Storage Resource Manager
SURL	Storage URL
SWIG	Simplified Wrapper and Interface Generator
TURL	Transport URL
UI	User Interface
VDT	Virtual Data Toolkit
VO	Virtual Organization
VOMS	Virtual Organization Membership Service
WN	Worker Node
WMS	Workload Management System

7 Bibliography

[1] gLite Grid middleware website - http://www.glite.org/

[2] EGEE website - http://www.eu-egee.org/

[3] neuGRID WP9 deliverable D9.1 - "Users Requirements Specifications (URS) document first release" – due month 14 of the project

[4] neuGRID WP9 deliverable D9.2 - "Users Requirements Specifications (URS) document final release" – due month 26 of the project

[5] EGEE-II-MJRA1.2 - "Functional description of grid components" - https://edms.cern.ch/document/736259/2

[6] MyProxy - http://grid.ncsa.uiuc.edu/myproxy/

[7] VDT, Virtual Data Toolkit - http://vdt.cs.wisc.edu/

[8] VeriSign - http://www.verisign.com

[9] The gLite-UI - http://glite.web.cern.ch/glite/packages/R3.1/deployment/glite-UI/glite-UI.asp

[10] java.lang.Runtime online javadoc: http://java.sun.com/j2se/1.5.0/docs/api/java/lang/Runtime.html

[11] JavaGAT - http://www.cs.vu.nl/ibis/javagat.html

[12] I. Foster, C. Kesselman and S. Tuecke, The Anatomy of the Grid, International Journal of High performance Computing Applications, 15, 3, 2001