EGI-InSPIRE

Integrating Resources into the EGI Production Infrastructure

EU MILESTONE: MS414

Document identifier:	EGI-MS414-V0-3.doc
Date:	**19/07/2011**
Activity:	**SA1**
Lead Partner:	**KTH**
Document Status:	**DRAFT**
Dissemination Level:	**PUBLIC**
Document Link:	https://documents.egi.eu/document/650

Abstract

<< The abstract should provide a brief neutral overview of the document and its contents and main conclusions. Once complete the abstract should be copied into the abstract field on the document server.>>This document describes and defines the operational interfaces that must be supported for resources to be integrated into EGI. This includes operational tools provided by the EGI-InSPIRE JRA1 activity and procedures and policies defined to ensure interoperability within EGI and in the interaction with other DCIs, the adoption of best practices and compliance with service level agreements.

<<document handling and production procedure is provided in https://documents.egi.eu/document/33 >>

I. COPYRIGHT NOTICE

II. DELIVERY SLIP

	Name	Partner/Activity	Date
From	Michaela Barth	KTH/SA1	
Reviewed by	**Moderator:** **Reviewers:** <<To be completed by project office on submission to AMB/PMB>>		
Approved by	**AMB & PMB** <<To be completed by project office on submission to EC>>		

III. DOCUMENT LOG

Issue	Date	Comment	Author/Partner
0	05/07/2011	Incomplete placeholder	Michaela Barth /KTH
1	06/07/2011	ToC	Michaela Barth /KTH
2	12/07/2011	Input on Operations Portal	Cyril L'orphelin / IN2P3
3	19/07/2011 27/07/2011 29/07/2011 01/08/2011	Input on GOCDB, Input on Argus in general, Argus and gLite Input on GOCDB Input on Argus and ARC Comments	David Meridith / STFC Alvaro Simon / CESGA Torsten Antoni / KIT Ali Gholami / Nordugrid Michaela Barth / KTH

IV. APPLICATION AREA

This document is a formal deliverable for the European Commission, applicable to all members of the EGI-InSPIRE project, beneficiaries and Joint Research Unit members, as well as its collaborating projects.

V. DOCUMENT AMENDMENT PROCEDURE

Amendments, comments and suggestions should be sent to the authors. The procedures

documented in the EGI-InSPIRE “Document Management Procedure” will be followed: https://wiki.egi.eu/wiki/Procedures

VI. TERMINOLOGY

A complete project glossary is provided at the following page: http://www.egi.eu/about/glossary/.

<<The authors should check if the acronyms are covered by the glossary page and if the definition is still correct; all the amendments should be communicated to glossary@egi.eu>>

VII. PROJECT SUMMARY

To support science and innovation, a lasting operational model for e-Science is needed – both for coordinating the infrastructure and for delivering integrated services that cross national borders.

The EGI-InSPIRE project will support the transition from a project-based system to a sustainable pan-European e-Infrastructure, by supporting 'grids' of high-performance computing (HPC) and high-throughput computing (HTC) resources. EGI-InSPIRE will also be ideally placed to integrate new Distributed Computing Infrastructures (DCIs) such as clouds, supercomputing networks and desktop grids, to benefit user communities within the European Research Area.

EGI-InSPIRE will collect user requirements and provide support for the current and potential new user communities, for example within the ESFRI projects. Additional support will also be given to the current heavy users of the infrastructure, such as high energy physics, computational chemistry and life sciences, as they move their critical services and tools from a centralised support model to one driven by their own individual communities.

The objectives of the project are:

1. The continued operation and expansion of today's production infrastructure by transitioning to a governance model and operational infrastructure that can be increasingly sustained outside of specific project funding.
2. The continued support of researchers within Europe and their international collaborators that are using the current production infrastructure.
3. The support for current heavy users of the infrastructure in earth science, astronomy and astrophysics, fusion, computational chemistry and materials science technology, life sciences and high energy physics as they move to sustainable support models for their own communities.
4. Interfaces that expand access to new user communities including new potential heavy users of the infrastructure from the ESFRI projects.
5. Mechanisms to integrate existing infrastructure providers in Europe and around the world into the production infrastructure, so as to provide transparent access to all authorised users.
6. Establish processes and procedures to allow the integration of new DCI technologies (e.g. clouds, volunteer desktop grids) and heterogeneous resources (e.g. HTC and HPC) into a seamless production infrastructure as they mature and demonstrate value to the EGI community.

The EGI community is a federation of independent national and community resource providers, whose resources support specific research communities and international collaborators both within Europe and worldwide. EGI.eu, coordinator of EGI-InSPIRE, brings together partner institutions established within the community to provide a set of essential human and technical services that enable secure integrated access to distributed resources on behalf of the community.

The production infrastructure supports Virtual Research Communities (VRCs) – structured international user communities – that are grouped into specific research domains. VRCs are formally represented within EGI at both a technical and strategic level.

VIII. EXECUTIVE SUMMARY

<< The text should provide a summary of the full report so that the reader can 'in a page' understand the problem it has been written to cover. This includes an overview of the background material and motivation for the report, a summary of the analysis, and the report's main conclusions.>>

TABLE OF CONTENTS

1 INTRODUCTION

<< The ‘introduction’ of the document provides information on why it has been written, who the target audience is and what they will learn from reading it.>>

2 INTEGRATION OF MIDDLEWARE ON OPERATIONAL TOOL LEVEL

2.1 Interoperation at an Infrastructure Level

2.2 Overview Status of Middleware Integration for each Operational Tool

2.3 Definition and Description of a Management Interface

2.3.1 Functionality

A management interface is an operational interface which allows sites to store, maintain and view the topology of the production infrastructure and the basic information about the respective resources within it. Such an EGI management interface contains information and their placement in the topology order on:

- Participating National Grid Initiatives (NGIs) and possible other groups (Countries, regional operators) and related information
- Grid sites providing resources to the infrastructure including management, technical and security related contact points
- Resources and services, including maintenance plans and service status information access points for these resources
- Participating people, and their roles within EGI operations

Besides providing a central management tool to view and define production state, downtimes and maintenance status and whether a resource needs monitoring, it shall in essence depict what services are running where and who to contact for certain type of issues. The presented information can be a combined view of different regionalized or otherwise separated instances with their own local inputs.

2.3.2 Requirements

The EGI management interface has to support the functionality described above. System and security contacts and higher level organizational management contacts for a site need to be easily identified. The management interface may provide finer granularity for contact details by marking extended expertise on a specific middleware stack or an affinity to certain types of service(s).

Additionally, it must be possible to register new kinds of service types, groups or sites within the management interface. A site should be able to contain services from different middleware stacks. The description and/or the name of the service type should also contain information on any middleware dependencies.

Such a database needs a role based interaction model, so that people responsible for certain sites, services or resources can update and maintain the various entries representing the entities under their responsibility within typical daily operations scenarios. In particular, basic service status information shall be easily viewable and changeable. It shall be easily possible to register a service of a known service type, to edit system administration information and put whole sites or single

resources in and out of downtime according to predefined procedures. It shall be easy to identify whether a resource is monitored or not by the corresponding monitoring system. This monitoring bit can be set separately or implicitly within the different production states.

A management interface provides information about a resource through the certification process. The history and details of the certification process and other state transfers like site decertification and suspension are desirable additional information.

Since the management interface provides much needed basic information on the topology of the production infrastructure and its contact points, we expect a plug-in to an approved dashboard interface to be in existence or easily implementable by using canonical standards. Even though the information is mostly static, a regionalized version with a central collecting portal of the management interface would of course be preferred in order to emphasize the distributed nature of the grid community and to avoid single points of failure.

We follow up with GOCDB as a working example for an implementation of a management interface.

2.3.3 Integration of new Services into GOCDB

Services registered in GOCDB have; 1) a 'Service Type' identifier, 2) a required 'Service Endpoint' instance and 3) an optional 'Endpoint Location'.

1. **Service Type;** a unique name that identifies a type of software component deployed on a Grid, including middleware (e.g. CE, WMS, SRM) and/or operational components (e.g. MessageBroker, RegionalNagios). The naming scheme for new service types follow a reverse DNS style syntax, usually naming the technology provider followed by technology type, i.e. '<provider>.<type>' (e.g. 'unicore6.StorageFactory'). This is consistent with the proposed EMI service registry naming scheme from GLUE2 that defines a service type enumeration. It would be preferable to rename all existing service types using this scheme, but this is potentially problematic for existing services that depend on established legacy names. The current list of service type definitions are given at: https://wiki.egi.eu/wiki/GOCDB/Input_System_User_Documentation#Service_types

2. **Service Endpoint;** represents a deployed instance of a service type.

3. **Endpoint Location**; a Service Endpoint may optionally define an Endpoint Location which locates the service (URL).

2.3.3.1 Procedure for registering new Service Types

New service types can be registered by GOCDB administrators. Once registered in GOCDB, users (site administrators, regional managers) can declare instances of the new service type as required. The complete procedure to integrate new service types is as follows;

1. If the service type is already registered in GOCDB, service endpoints can be added by users of GOCDB following the established procedure.
2. If the service type is not registered, a request should be made to the OTAG for its inclusion in GOCDB (e.g. by the new middleware provider or JRA1 community). If the

new service type belongs to a previously undeclared middleware stack, then a strategic decision is required to ensure only officially supported middleware is integrated into GOCDB. If the request is approved, it is communicated to the GOCDB developers to add the new service type.

3. The requesting party is notified (either the request is rejected or completed).

2.3.3.2 Regular review of the list of available service types

A regular review of the supported GOCDB service types will be made. This is the responsibility of GOCDB developers, who will consult the Technical Coordination Board (TCB) (software providers including EMI, EGI-JRA1) together with the OMB.

2.3.3.3 Integrated operational service types

- **Site-BDII**: [Site service] This service collects and publishes site's data for the Information System. All sites MUST install one Site-BDII.
- **Top-BDII**: [Central service] This is the "top-level BDII". These collect data from site-BDIIs and publish the data. Only a few instances per region are required.
- **OpsTool**: [Central service] generic service representing an operation tool (topology repository, dashboard, helpdesk system...)
- **MSG-Broker**: [Central service] A broker for the central/backbone messaging system.
- ~~**RGMA-IC**: [OBSOLETE Central service] This is the Registry for an R-GMA service. There will only ever be a few of these per grid.~~
- **Site-NAGIOS**: [Site service] site-level Nagios monitoring box
- **National-NAGIOS**: [Regional Service] NGI-level Nagios monitoring box
- **Regional-NAGIOS**: [Regional Service] ROC-level Nagios monitoring box
- **Project-NAGIOS**: [Central Service] project-level Nagios monitoring box
- **MyProxy**: [Central service] The My Proxy service is part of the authentication and authorization system. Often installed by sites installing the WMS service.

2.3.3.4 Integrated gLite service types

- **CE**: [Site service] The LCG Compute Element. Currently the standard CE within the gLite middleware stack. Soon to be replaced by the CREAM CE.
- ~~**gLite-CE**: [OBSOLETE Site service] The gLite Compute Element is now obsolete and is not supported. Please avoid using this middleware service.~~
- **CREAM-CE**: [Site service] The CREAM Compute Element is the new CE within the gLite middleware stack.
- **APEL**: [Site service] This is a "dummy" Service Type to enable the monitoring tests for APEL accounting. All sites must have one instance of this Service Type, associated with a CE.
- ~~**MON**: [OBSOLETE Site service] The gLite MonBox hosts the site R-GMA services.~~
- **UI**: [User service] The User Interface. Can be installed by users but more commonly installed by a site.
- **SRM**: [Site service] Storage Resource Manager. Mandatory for all sites running an SRM enabled storage element.

- ~~**Classic-SE**: [OBSOLETE Site service] The Classic Storage Element is now obsolete and is not supported. Please avoid using this middleware service.~~
- **Central-LFC**: [Central service] An instance of the gLite file catalogue which holds entries for all files owned by a particular VO. NOTE: An LFC can be both Central and Local.
- **Local-LFC**: [Site service] An instance of the gLite file catalogue which holds entries for files owned by a particular VO, at your site. NOTE: An LFC can be both Central and Local.
- **WMS**: [Central service] gLite Workload Management Service. Acts as the broker for matching user jobs to available computing resources.
- ~~**RB**: [OBSOLETE Central service] The LCG Resource Broker is now obsolete and is not supported. Please avoid using this middleware service.~~
- **VOMS**: [Central service] VO Management System. Part of the authentication and authorization system. This service only needs to be installed on the request of a VO.
- **LB**: [Central service] gLite Logging and Bookkeeping. Usually installed by sites running a WMS. One LB service can support several WMS instances.
- **AMGA**: [Central service] gLite metadata catalogue. This service only needs to be installed on the request of a VO.
- **FTM**: [Site service] gLite File Transfer Monitor. Monitors the FTS service at a site.
- **FTS**: [Central service] The gLite File Transfer Service manages the transfer of files between sites. This service only needs to be installed on the request of a VO.
- **VO-box**: [Site service] The gLite VO box allows a VO to run their own services at a site. This service only needs to be installed on the request of a VO.
- **gLite-APEL**: [Site service] The gLite-APEL hosts the site Accounting client (3.2 replacement of the MonBox)
- **gLExec**: [Site service] A light-weight gatekeeper to authenticate and authorize credentials according to local site policy and execute commands.

2.3.3.5 Integrated ARC service types

As of release 0.8 of ARC, the ARC-CE runs a resource BDII with GLUE schema 1.3, in the same way as gLite resources. Hence setting up a special site BDII is no longer needed. More details are found in [R 22]. → Verify

- **ARC-CE**: [Site service] The Compute Element within the ARC middleware stack.
- **SGAS**: [Site service] An accounting service used by ARC.

2.3.3.6 Integrated UNICORE service types

- **unicore6.Registry**: [Central service] All UNICORE services register here; clients ask the registry for available services in the Grid. Normally one Registry per Grid infrastructure which collects URLs of services.
- **unicore6.Gateway**: [Site service] Sits in front of one or more UNICORE services as a gateway to the internet. Normally one Gateway per site.
- **unicore6.TargetSystemFactory** [Site service] used as an entry-point for submitting single jobs. It can create Target System Services (TSSs) and submit jobs to those TSSs.
- **unicore6.StorageFactory** [Site service] Creates StorageManagement instances. A user can create dynamic storage management services for own purposes with it. Often used to provide file space during workflow execution.

- **unicore6.StorageManagement** [Site service] Provides an abstract file system-like view on a storage resource. A Storage Management Service (SMS) can be created by a Storage Factory or can be configured statically way by a configuration file.
- **unicore6.ServiceOrchestrator** [Site service] Handles dispatching of a workflow's atomic jobs, and brokering. Normally there is one per grid infrastructure.
- **unicore6.WorkflowFactory** [Site service] Used as an entry point for submitting workflow jobs. The Workflow factory is creating workflow instances and can submit workflows to them. It is the workflow submission equivalent to the Target System Factory used for single job submission.
- **unicore6.UVOSAssertionQueryService** [Site service] Provides data and user information via the SAML standard as needed for authorization and environment customization.

Add comment on why unicore6.UNICOREX was removed as discussed in the UNICORE integration taskforce, since it is a just an undefined collection of an undefined number of other services like TargetSystemFactory, StorageFactory, StorageManagement.

Add a list of other possible future services as listed in the intergration mailinglist

2.3.3.7 Integration of Globus resources

- **GRAM5**: [Site service] job submission service for Globus version 5.x (GRAM5)
- **globus-GRIDFTP**: [Site service] storage endpoint and data transfer service for the Globus middleware stack
- **globus-GSISSHD**: [Site service] certificate based interactive login service for the Globus middleware stack

Update with latest discussion in Globus integration task force (MDS, GridSAFE, ISS,...)

2.4 Definition and Description of a Monitoring Interface

2.4.1 Functionality

2.4.2 Requirements

2.4.3 Interoperability of different MW Stacks with SAM/Nagios

2.4.4 Procedures to integrate new Nagios Probes

https://wiki.egi.eu/wiki/PROC07
https://wiki.egi.eu/wiki/PROC06

2.4.4.1 Tests and Nagios probes for gLite resources

2.4.4.2 Tests and Nagios probes for ARC resources

The ARC monitoring tests became operational on 7.04.2011.

http://wiki.nordugrid.org/index.php/Nagios_Tests

2.4.4.3 Tests and Nagios probes for UNICORE resources

2.4.4.4 Tests and Nagios probes for Globus resources

2.5 Definition and Description of an Accounting Interface

2.5.1 Functionality

2.5.2 Requirements

2.5.3 Current Status

Needs surely reference to EMI Compute Accounting working group

2.6 Definition and Description of a Support Interface

2.6.1 Functionality

The user support infrastructure in use within EGI is distributed consisting of various topical and regional helpdesk systems that are linked together through a central integration platform, the GGUS helpdesk. This central helpdesk enables formalized communication between all partners involved in user support by providing an interface to which all other tools can connect and enabling central tracking of a problem, independent of the origin of the problem and the tool in which the work on the problem is done.

The interlinking of all ticket systems in place throughout the project enables to pass trouble tickets from one system to the other in a way that is transparent to the user. It also enables the communication and ticket assignment between experts from different areas (e.g. middleware experts and application experts) while at the same time allowing them to work with the tools they are used to. A standard has been defined for the interface between ticket systems and also a template for a ticket layout exists to ensure the quality of service. These are documented in the GGUS documentation [R 36].

For EGEE, and now EGI, a functional body has been defined to keep track of the ticket processing management (TPM). The TPM keeps a global overview of the state of all tickets and is responsible for those tickets that have to be assigned manually, i.e. so that they get forwarded to the correct

support units. The TPM teams act as a 1st line support chain and have also to keep track of long-term trouble tickets and help to solve them with their very good general grid knowledge. In this way, a problem submitted to GGUS can be quickly identified as either a grid problem or a VO specific problem and addressed to the appropriate second line specialized support units or the dedicated VO support teams whose members have specific VO knowledge.

The second line support is formed by many support units. Each support unit is formed from members who are specialists in various areas of grid middleware, or regional supporters for operations problems, or VO specific supporters. The membership of the support units is maintained on mailing lists. A single e-mail address is available through which users can request GGUS for help. E-mails sent to this address are automatically converted into tickets and treated by the system.

2.6.2 Requirements

Regardless of the number of parties involved, the submitter of a trouble ticket should be able to transparently follow the chain of actions needed to solve the reported problem. This transparency together with the independence from the actual ticket system is used by the experts from the different areas who get assigned to the ticket. It can be seen that the main requirement of the ticketing system is that information flows between different parts of the EGI support network.

This is especially important since the support interface is not only used for 3rd level support dedicated to the end user, but also for the relevant parts of internal trouble ticket communication fulfilling standard operational, grid oversight and partially also development functionalities.

Other relevant requirements on the support interface is the existence of a functional body like the TPM as described above and the connection to a useful, searchable and well maintained knowledge base.

Other basic requirements that can be expected from a more advanced support ticket system:

- Differentiating between real problem tickets and service requests
- Ability to mark a ticket as spam
- Mail notification when a ticket is assigned to a support unit or person possible
- Possibility to involve several experts at the same time
- Searching tickets via ticket ID as well as via parameters
- Automatic reminders
- Several tickets describing the same problem can be put into a master-slave relation.
- Other dependencies can be represented with child and parent relations.

2.6.3 Integration of new Resources into GGUS

There are three distinct cases to be considered when integrating new resources into the EGI user support infrastructure:

2.6.3.1 Integration of a new Resource Centre into the infrastructure

In case a new resource centre is added to the EGI infrastructure this is resources centre is always part of an NGI. This means that NGI management has to make sure that all steps are taken that are needed. For the user support area this is a simple case as the information about resource centres is extracted from GOCDB. This means that no manual steps are needed to integrate a new resource

centre in GGUS.

2.6.3.2 Integration of a new NGI into the infrastructure

If a new NGI joins the EGI infrastructure it is required to provide a ticket system which is integrated with GGUS. This can be done in different ways, depending of the size and the maturity of the NGI.

- The simplest way, which might be suitable for a small new NGI is to use GGUS directly. This has the limitation of just one support unit for the whole NGI. Tickets cannot be assigned to specialized groups or specific resource centres within the NGI. This further processing of the tickets is done independently from the EGI support infrastructure.
- The NGI can make use of xGUS a customisable slimmed-down regional instance of GGUS. xGUS is hosted and maintained by the GGUS team. Customization can be done via an administrative web interface, which enables creating and managing support units and defining special workflows. xGUS comes with the interface to GGUS built in.
- The NGI can set up its own ticket system. In this case the NGI has to make sure that their ticket system fulfils the requirements of the interface definition to GGUS. The NGI ticket system needs to be interfaced to GGUS and the NGI is responsible for maintaining this interface. This for example includes testing the interface after releases of the GGUS portal.
- Details on the NGI creation process can be found on a dedicated page in the wiki [R 37].

2.6.3.3 Integration of a new Technology Provider into the infrastructure

Should EGI decide to utilize software from a technology provider that has not so far involved with the project, an agreement has to be found with that technology provider on how to integrate its support infrastructure with the EGI's. This process has taken place for the EMI and IGE projects.

EGI has set up a Technology Helpdesk which is interfaced to GGUS for that purpose. No general description of the details of the integration of a new technology provider into the Technology Helpdesk can be given here, as this is highly dependent on the internal support structure of the respective technology provider. Nevertheless it is important that this is done in a way that enables EGI to have an overview of issues with the products provided by the technology provider and to gather statistics on the quality of the support given by the provider.

EMI has set up a structure within the Technology Helpdesk for its various products, including e.g. ARC or UNICORE.

3rd level support for Globus will be provided by IGE. IGE provides a support infrastructure for the European Globus users in all European, national, and regional e-Infrastructures with EGI and DEISA/PRACE being the most important ones. The Technology Helpdesk contains a queue to forward 3rd level support tickets directly to the IGE user support team.

For details on the Technology Helpdesk refer to MS410

2.7 Definition and Description of a Dashboard Interface

2.7.1 Functionality

In order to operate a distributed infrastructure, management and monitoring information has to be collected and presented in a labour saving way to assist the operators of the infrastructure in their

daily work. The dashboard interface combines and harmonizes different static and dynamic information and therewith enables the operators to react on alarms, to interact with the sites, to provide 1st line support and/or to really operate the sites by creating and supervising problem tickets on regional as well as central level.

The dashboard allows predefined communication templates and is adaptable to different operational roles (1st line support, regional, central). Sites in the dashboard scope can be regional, central or predefined out of a list and can be sorted and displayed according to numerous criteria to indicate actions needed for a single service, but also for a whole region or even the whole production infrastructure.

2.7.2 Requirements

A dashboard interface has to fulfil the functionality described above.

With the increasing relevance of the SAGA Service Discovery specification [here] (OGF) for a standards-based approach for interoperability one more requirement on the dashboard is to provide such a well defined interface in order to be prepared for the harmonized integration of many different third party information providers.

We assume that EGI as a whole should try to unify the input:

- All sites should publish their information via a harmonized information service independently of the middleware stack used (e.g. GLUE2 based BDII)
- Access should be regulated by a harmonized user authentication service like VOMS or something better (see also detailed discussion in section 2.8).

Thus the dashboard and other tools don't have to be adapted to too many different information and authentication services.

In reality, though, it might be equally important to more directly connect to prevalent third-party information providers. A dashboard design that can effectively handle commonly used information services, especially those already established within EGI, while at the same time providing a well defined standard interface for interactions is the preferred solution.

2.7.3 The Operations Portal

The Operations Portal [here] content is based on information which is retrieved from several different distributed static and dynamic sources – databases, Grid Information System, web services, etc. – and gathered onto the portal. Interlacing this information has enabled us to display relevant views of static and dynamic information of the EGI production grid.

Integrating different technologies and different resources creates high dependencies to the data provided. Consequently, our technical solution is organized around a web service implementation that provides a transparent integration of each of these resources. The web service in question is named Lavoisier [here].

The goals of Lavosier are to provide:

- a web layer as independent as possible from the mechanisms technology used to retrieve the original information,
- intermediate information usable in the same format in order to cross-query it and
- information which is independent from the availability of the data provider.

This solution design means that the web application does not need to know the exact location of the

data provider and neither which kind of technology has provided the information initially. All these concerns are already taken into account by Lavoisier.

Lavoisier has been developed in order to reduce the complexity induced by the various technologies, protocols and data formats used by its data sources. It is an extensible service for providing a unified view of data collected from multiple heterogeneous data sources. It enables us to easily and efficiently execute cross data sources queries, independently of used technologies. Data views are represented as XML documents and the query language is XSL.

The global architecture of the Operations Portal is presented in Fig. 1.

By using a plug-in schema, information can be retrieved from heterogeneous data providers (on the left side of the schema in Fig. 1). These plug-ins transform information in various formats extracted from different technologies (i.e. RDMS, JSON, JMS, ldap, http, web service) into a standard format XML. At this stage it is easy to execute cross data sources queries by using XSLT transformation. In the end the web application is using all information in the same format (XML).

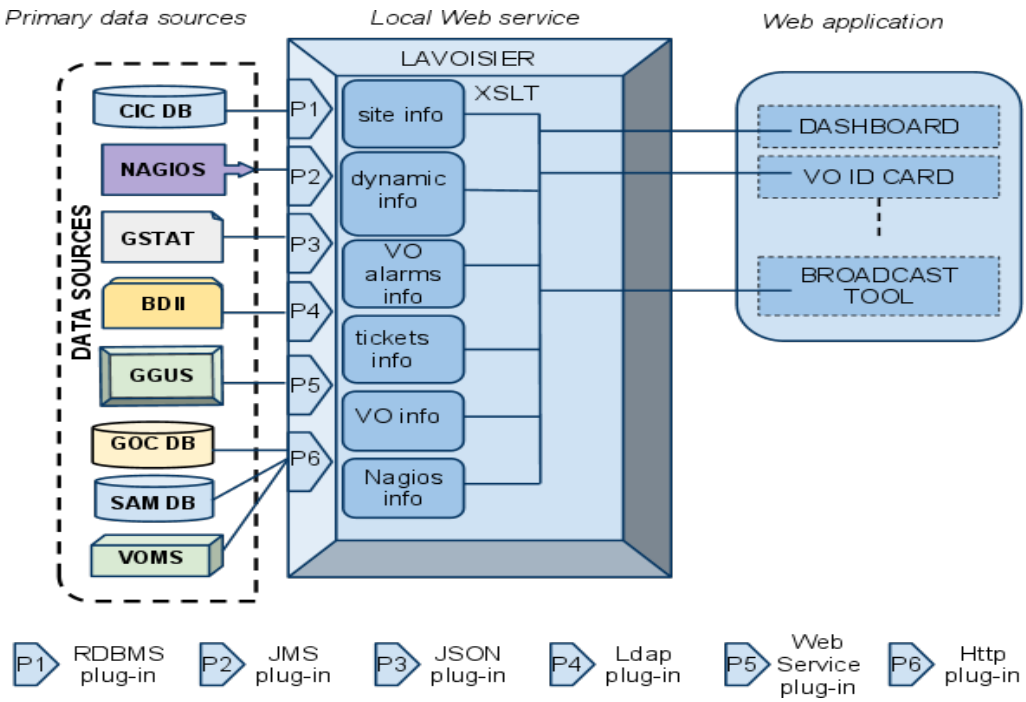

Fig. 1: Global architecture of the Operations Portal.

2.7.3.1 Integration of a new resource

The architecture of the portal has been designed to propose a standard access to information from an extended number of data sources. The integration of new data sources is eased by the use of the Lavoisier web service.

In the case of a known technology we will create and add a new view by using an existing plug-in out of the wide-range of plug-ins already available.

If a site and its resources are already integrated in all the other operational tools through existing information providers (e.g. registered in GOCDB, monitored by Nagios, publishing their information

via BDII and having a tree in GGUS), existing plug-ins can be reused and no additional integration effort for the usage of the Operations Portal is needed.

For new providers, we will develop new plug-ins to be able to retrieve information from a new provider.

The integration of different information systems present in different middlewares such as ARC, UNICORE, or Globus can be done via an abstraction layer.

One such a possible abstraction layer could be to integrate the SAGA Service Discovery specification [here] (OGF) into a Lavoisier plug-in which will permit to access information using different services (like the information service of UNICORE – CIS [here]) and different schemas like CIM [here] or GLUE Schema [here] standards.

Lavoisier's flexibility allows us to be ready to integrate almost any kind of new information. Such an integration is certainly needed and meaningful for the new resource types entering EGI, such as HPC systems, virtualized resources or desktop resources. As long as these resources are monitored, it is possible to integrate them via plug-ins inside Lavoisier.

The integration will be done step-by-step during the whole project. The difficulty will be to identify the priorities in the components to integrate.

2.7.3.2 Alternative possibilities to integrate new information providers

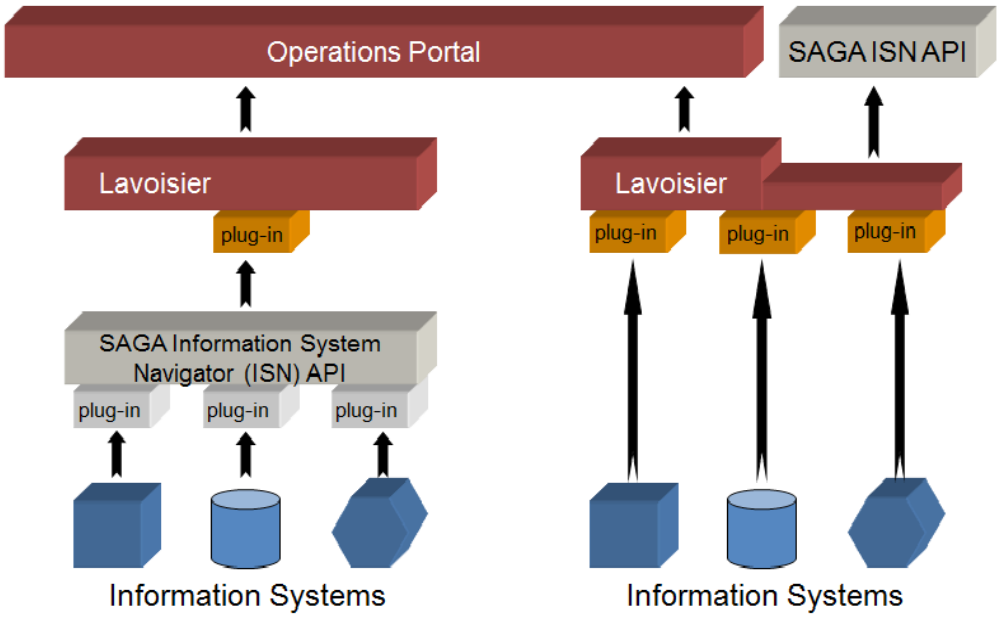

Fig. 2: Integration of new information systems into the Operations Portal

So far, no clear recommendation has been given yet on how to best include new information providers to the dashboard developers. The alternative depicted on the left side of the picture above might seem more work at first, but part of this work could probably be outsourced to the information providers and reused for other purposes. On the other hand, a Lavoisier to SAGA Information System Navigator (ISN) link might be needed anyway. The two possible alternatives are not mutually exclusive and might be combined.

2.7.3.3 Integration of a gLite resources

Plug-ins for all relevant information providers in the case of a site's gLite resources (Nagios, GOCDB, GGUS, BDII) exist and gLite resources can therefore be operated from within the Operations Portal.

2.7.3.4 Integration of a ARC resources

Plug-ins for all relevant information providers in the case of a site's ARC resources (Nagios, GOCDB, GGUS, BDII) exist and gLite resources can therefore be operated from within the Operations Portal.

2.7.3.5 Integration of a UNICORE resources

The UNICORE resources are registered in GOCDB and starting to be monitored by SAM/Nagios, GGUS trees exist. Hardware GLUE information could be taken from the Central Information Service CIS over the SAGA
ISN API link.

2.7.3.6 Integration of a Globus resources

Globus GT5 resources are registered in GOCDB and starting to be monitored by SAM/Nagios, GGUS trees exist.

Taking into account that LCG-CE is very similar to Globus GRAM, lcg-ce information providers can be reused for the BDII. With that Globus resources should be able to be directly integrated into the operational dashboard.

2.8 User Management, Authentication and Authorization

The actual way users are administrated and authenticated effects many operational interfaces that have been defined so far. This might be especially true for accounting, but is equally relevant for monitoring or when using a high level tool like the operational portal.

The basic information on who is authorized to access a site's resources can be stored in different ways within different distributed infrastructures interested to join or collaborate with EGI.

Within the EGI production infrastructure one primary authentication token is the X.509 certificate and its proxy derivatives. A user would e.g. request a X509 credential with VOMS extensions from a national or organizational Certificate Authority (CA) which is recognized by the International Grid Trust Federation (IGTF) (see also [R 11]). Resources within the production infrastructure are made available to controlled collaborations of users represented in the infrastructure through e.g. Virtual

Organizations (VOs). Access to such a VO is governed by a VO Manager who is responsible for managing the addition and removal of users and the assignment of users to groups and roles within the VO.

On site authorization information could be translated via native VOMS support or grid-mapfile equivalents.

In EGI there are resource providers who are not willing to offer pool accounts on their resources in order to enforce proper access control. Users have to apply for a personal account first and have a certificate mapped to it.

Ideally, EGI would provide a central service where users apply for an EGI user account (within a VO) and then the accounts are created at the resource providers sites.

2.8.1 Desired Functionality of a user authorization system

- Providing a consistent approach for Identical DN/UID mapping which is not dependent to shared file systems
- Support for accounting of pilot jobs
- Global banning and unbanning of users over sites and services
- Providing an administrative tool to maintain and control DN and policies

2.8.2 Requirements on a user authorization system

We have different requirements:

- Identical user mapping functionality
    - It should be possible to use a centralized approach to do the DN/UDI mapping in a consistent approach. Solutions based on shared file system or shared pool directory are not acceptable as they add dependency to the middleware since they are not all POSIX compliant. Besides in case of multiple users, each try to overwrite a shared entity which cause inconsistency.
- User authorization
    - This feature enables site administrators to ban users based on DNs, CAs, VOs for a whole site or over multiple services.
    - The banning list and other policies can be created using SPL (a language to create and customized policies).
- Support for pilot jobs
    - Pilot jobs are submitted through pilot submitter and the real owner of the jobs until they start execution on the worker nodes are not known which is important in the case of accounting. Using e.g. Argus as a centralized service, it should be possible to map users to a particular POSIXUID/GID.
    - Pilot jobs are two types: single-user and multi-user jobs. It's only multi-user pilot jobs which create authorization problems.
    - This feature is to be done as further work.

2.8.3 Argus

EMI has selected the ARGUS authorization framework as general approach for user authorization based on the common SAML profile which shall be supported over all middleware stacks.

In a VO, the VO admin assigns attributes and membership to people and this is controlled by the VOMS, but the sites can not influence this information. However a site sometimes wants to control access in more fine grained detail: like to ban one user from a certain VO, or limit the access to some of the resources. With the help of Argus this is no longer a problem.

Argus is a authorization system for distributed services such Compute Elements, Portals and Worker Nodes and it replaces the Site Central Authorization Service (SCAS). In order to achieve this consistency a number of points must be addressed. Argus consists in several distinct components. The first component is the Policy Administration Point (PAP for short) service. Second, authored policies must be evaluated in a consistent manner, this task is performed by the Policy Decision Point (PDP). And finally, the data provided for evaluation against policies must be consistent, this is done by the Policy Enforcement Point (PEP).

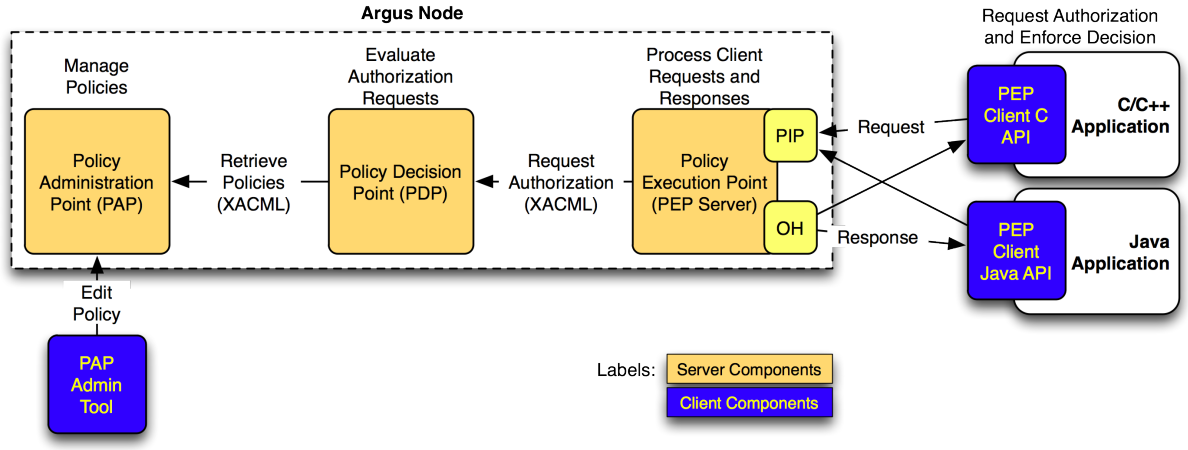

Fig. 3: Argus Components

2.8.3.1 Argus and gLite

Several services can interact with Argus in gLite, eventually every service that uses SCAS for users validation can be migrated to use Argus. Basically Argus is designed to answer questions in the form of *Can user X perform action Y on resource Z at this time?.* If so, Argus gives a response to the PEP java client and the user can perform the action. If the request does not match to any appropriate access control policy then the access is rejected.

Several gLite services will be integrated with the ARGUS EMI authorization system:

- CREAM: Argus policies will grant access to grid users to access CREAM-CE computing resources. When a new user job is submitted to CREAM the site Argus instance is requested to accept or deny the job submission based on the site Argus policy.
- WN/gLExec: Pilot jobs can be mapped to a specific grid user based on Argus policy reponse instead of SCAS. Pilot jobs are mapped to grid user into WN following the Argus site security policy.

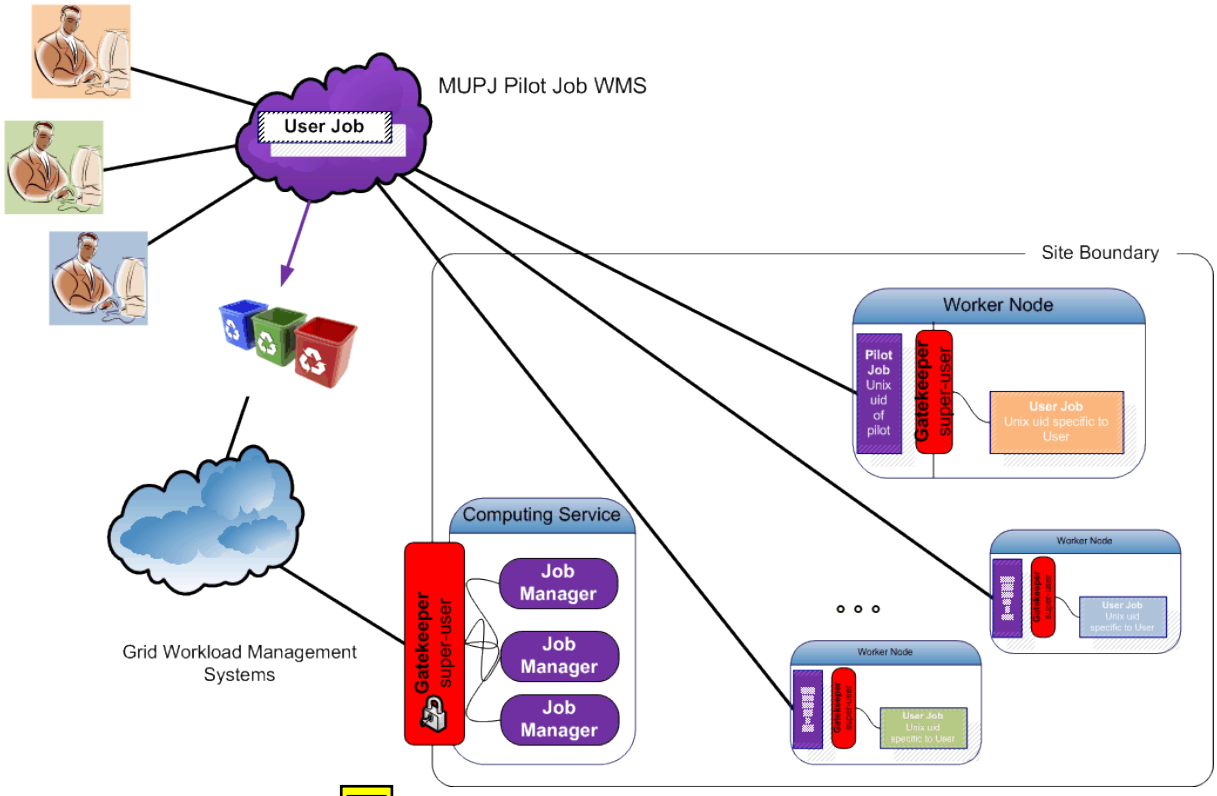

Fig. 4: gLExec Infrastructure

2.8.3.2 Argus and ARC

Nordugrid ARC middleware requires a consistent mechanism to provide authorization based on user DNs. Existing ARC releases don't provide coherent solutions to address issues such as identical DN/UID mapping, DNs and policy maintenance, Global banning and unbanning of users over sites or specific services and support for accounting of pilot jobs. To overcome these issues, the Argus authorization framework has been opted as an effective solution to be integrated with the Hosting environment daemon (HED) component in ARCv1.

HED is in charge of authorization requests for incoming user jobs. During the user ID mapping process the HED component initiates the authorization client which then communicates with the PEP daemon in Argus. As a first step, the ID mapper within HED collects the Grid credentials and tries to configure the HED authorization client so it can establish a communication channel between the HED client and the Argus authorization framework to send and receive the eXtensible Access Control Markup Language (XACML) requests/responses. XACML is a declarative access control policy language based on XML and can be used as a processing model which describes how to interpret the policies http://docs.oasis-open.org/xacml/2.0/access_control-xacml-2.0-core-spec-os.pdf .

By default an ARC authorization and authentication request is composed of a XACML subject, resource, action and and an additional XACML environment element which differs from the response structure received by Argus with these attributes: XACML decision element and obligation. The HED authorization client uses the gLExec LCMAPS plug-in to send and receive these requests and eventually parse the XACML response decision to authorize the user and the obligations to map a

user to a local account.

Currently as proof of concept an Argus provided client is in charge of send/receive messages to the PEP daemon.

Further details on implementation, deployment and configuration examples can be found under Into references: http://wiki.nordugrid.org/index.php/Argus_integration

2.8.3.3 Argus and UNICORE

Ask Krzysztof Benedyczak

Still something like that?:

There are exemplary ways to distribute the authorization information in a unified way in a large grid infrastructure. In D-Grid, the central Grid Resource Registration Service (GRRS) knows about resources and which VOs are allowed to use them. Each VO has a VO management registration service (VOMRS) server where users are registered with their certificate and D-Grid userID after they have applied for a userID and the VO membership. From this information a service is preparing mapping files for Globus, gLite, dCache [R 7], and UNICORE for each site which then are used by the relevant local services, e.g. the UNICORE User Database XUUDB.

In Argus users can be banend. In UNICORE with the user attributes there is a default policy, people with the corresponding user attribute can access this site and use its services. The site admin can overwrite this user attribute in order to say that this user is banned.

So the bottomline here is, that in the case of UNCIORE Argus is not a crucial service, since authorization can already be controled in another way, but it might be fruitful to use it anyway: like in the scenarios where we are having two middlewares deployed together and Argus can control both of them.

The currently planned status is that the UNICORE release in EMI 1 will use authorization by using Argus. So theoretically it will be possible for site admins to regulate access. Unfortunately this won't be useful at all, since currently the Argus policy language is quite simple. And therefore it is not yet possible to decide a possible useful policy for Argus. That part will be implemented later and afterwards Argus will definitely be a useful solution, and it will be possible to use ARGUS for gLite and for UNICORE at the same time.

In detail UNICORE will not use Argus for authorization, but only the PAP daemon to ask for the authorization policy, then UNICORE can use that directly: The format of the PAP is the same as the one used in UNICORE in general, so it won't be required to do a network call to Argus and like this it will also work if Argus is shut down. But this depicted solution is not implemented at all currently, it is not even in the orginal plans and Argus people don't advertise this PAP daemon. For them the only entry point is just another Argus. I' have to understand how Argus really works internally, so we could do this and implement it in EMI 1.

PS: PAP allows to download all policies in the service.

KB: So we can just query PAP, this is trivial to be done in UNICORE.

PS: Within Argus there is a specific authorisation service, that uses those policies.

KB: you mean PDP, it is not specific service, it is part of any UNICORE installation container.

PS: So this component is already part of every UNICORE service, every service has to download the Argus policy and implement it according to Argus?

KB: The current approach to use Argus within EMI is to choose PDP to use for each UNICORE service container, so each single service will access Argus. This takes time and can fail, since Argus can be down or the network can be overloaded, and the protocols used for supporting things like policies are changed regulary.

ML: So but your proposed solution which sounds very effective is not yet in EMI roadmap?

KB: EMI roadmap is still very general, there it is only written that UNICORE should be prepared to use Argus. In gLite for example the approach is already very different: an extra PIP is used which covers PDP.

ML: We will have to make sure that interfaces are standarized and stable and clearly defined.

KB: The Argus daemons are talking to each other using standardized protocols. The PIP to PDP on the other hand used in the case of gLite is propiatary. So it is clearly preferable and better to use PDP and PAP directly since those two have standardized interfaces. I was able to configure and ask PDP and PAP from UNICORE directly. And it is probable that we'll aim for the simplest and easiest development solution within EMI.

ML: Who, besides you is involved in the UNICORE development part for Argus?

KB: for UNICORE is just me, maybe Piotr, too. Valery is maybe also relevant for this work since he is leading the XACML working group (an EMI working group), which is aiming at standardizing XACML attributes used in the policies.

[11:18:44] Krzysztof Benedyczak https://twiki.cern.ch/twiki/bin/view/EMI/EmiJra1T4XACML

ML: Are there tendencies to go bigger? Are there plans to make this into an international standard later on by example in OGF?

KB: Not yet, it needed to define it more clearly first.

KB: The version from EMI 1 can talk to Argus: But you are not able to define the default authorization policy in Argus, some simplified policy language is used. The same story is valid for ARC, afaik. They also still have attributes which are unsupported by Argus. Argus is going to be updated by the end of this year to for EMI release 2 to implement and include all these attributes as it has been agreed between UNICORE, ARC and gLite.

EI: Can you already have access to UNICORE sites with this simple policy?

KB: Now, in UNICORE it is more complicated, there are owners of resources, and this feature is currently not supported.

KB: So basically it is working but in an insecure way, so it is not useful.

KB: The EMI 2 plan is to finish most critical updates of Argus by the end of this year, but not much earlier.

2.8.3.4 Argus and Globus

Globus still all relies on the entries in the Globus grid-mapfile for authorization purposes.
VOMS of VOMRS can be used to provide the necessary entries in order to achieve a high-level VO management for Globus.

Ask oscar okoeroo@nikhef.nl on something less technical?

The following features concern the Globus gatekeeper, gridftpd and gsi-opensshd:

Features to ramp up to Argus integration, planned release UMD 1.2:
- Non-VOMS poolaccount support (legacy feature)
- VOMS-based authorization and (pool)account mapping

Feature planned before the end of the year:

- Integrate the Argus call-out as a supported plug-in

On the todo list:
- Minor development in the already existing Argus plug-in
- Ensure that the Argus protocol libraries are suitable for integration on the platforms IGE wants to be able to deploy on. Some issues need to be resolved for SLC6/CentOS6 and probably Debian6 too.

3 INTEROPERATION AT PROCEDURES AND POLICY LEVEL

3.1 Scope

3.2 Current EGI Procedures and Policies

3.3 Future Procedures

4 OUTLOOK AND FUTURE PLANS

4.1 Operational requirements coming from NGIs

4.2 Operational requirements coming from our integration taskforces

NEW!

During the last year specialized integration taskforces have been created to keep an open dialogue between all involved parties and in order to keep more transparently track of the current ongoing efforts.

In the direct future: Deeper focus on Accounting with the outcome after the EMI Computeraccounting wg within the UNICORE and Globus integration task forces. And learning from our staged-rollout experiences.

Will be able to see how good our procedures (like site certification, defining operatinoal set of SAM probes,..) work and where they have to be adapted.

4.2.1 UNICORE integration taskforce

The UNICORE integration taskforce started its work in February 2011.

https://wiki.egi.eu/wiki/UNICORE_integration_task_force

- unicore-integration-tf@mailman.egi.eu https://www.egi.eu/sso/groupView/unicore-integration-tf

One of the biggest issues with GOCDB UNICORE integration could be solved by implementing an alternative solution to enter ServiceEndPointURLs into GOCDB. After some bug reports all RFC 3986 chars can now be entered into the URL field. The first UNICORE services have been added to GOCDB. UNICORE SAM Nagios probes should now be included in SAM Release 13 which has been shifted to August.

Problems with certain not completely middleware independent formulations in the Resource Center OLA, have been brought to notice of the OMB and will be discussed there with other changes to the OLA in December 2011, January 2012.

4.2.2 Globus integration taskforce

After the success of the UNICORE integration taskforce the Globus integration taskforce has been brought to life during the Technical Forum in Vilnius in April 2011.

(refs, as above)

Globus SAM Nagios Probes have been included already by SAM Release 11. Now we will follow through the whole staged rollout-out process.

4.3 Operational requirements coming from Collaborations with other DCIs

5 REFERENCES

R 1	Operations Portal Home Page https://operations-portal.in2p3.fr
R 2	Lavoisier Home page http://grid.in2p3.fr/lavoisier
R 3	SAGA Service Discovery API http://www.ggf.org/documents/GFD.144.pdf
R 4	Common Information Service (CIS) for UNICORE Grids http://www.unicore.eu/community/development/CIS/cis.php http://www.d-grid.de/fileadmin/user_upload/documents/MonitoringWorkshop/Memon.pdf
R 5	Common Information Model Home Page http://www.dmtf.org/standards/cim/
R 6	GLUE schema http://infnforge.cnaf.infn.it/glueinfomodel/ Glue Schema specifications http://www.ogf.org/documents/GFD.147.pdf