

# EGI-InSPIRE

## Integrating Resources into the EGI Production Infrastructure

### EU MILESTONE: MS407

---

Document identifier:	EGI-MS407-final
Date:	<b>07/01/2011</b>
Activity:	<b>SA1</b>
Lead Partner:	<b>KTH</b>
Document Status:	<b>FINAL</b>
Dissemination Level:	<b>PUBLIC</b>
Document Link:	<a href="https://documents.egi.eu/document/111">https://documents.egi.eu/document/111</a>

---

#### Abstract

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 partner grids, the adoption of best practices and compliance with service level agreements.

## I. COPYRIGHT NOTICE

Copyright © Members of the EGI-InSPIRE Collaboration, 2010. See [www.egi.eu](http://www.egi.eu) for details of the EGI-InSPIRE project and the collaboration. EGI-InSPIRE (“European Grid Initiative: Integrated Sustainable Pan-European Infrastructure for Researchers in Europe”) is a project co-funded by the European Commission as an Integrated Infrastructure Initiative within the 7th Framework Programme. EGI-InSPIRE began in May 2010 and will run for 4 years. This work is licensed under the Creative Commons Attribution-Noncommercial 3.0 License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nc/3.0/> or send a letter to Creative Commons, 171 Second Street, Suite 300, San Francisco, California, 94105, and USA. The work must be attributed by attaching the following reference to the copied elements: “Copyright © Members of the EGI-InSPIRE Collaboration, 2010. See [www.egi.eu](http://www.egi.eu) for details of the EGI-InSPIRE project and the collaboration”. Using this document in a way and/or for purposes not foreseen in the license, requires the prior written permission of the copyright holders. The information contained in this document represents the views of the copyright holders as of the date such views are published.

## II. DELIVERY SLIP

	Name	Partner/Activity	Date
<b>From</b>	Michaela Lechner	KTH/SA1	26/07/2010
<b>Reviewed by</b>	<b>Moderator:</b> Steven Newhouse <b>Reviewers:</b> Martin Radecki, Zoltan Balaton, Jaroslaw Marek	EGI.eu  EGI, EDGI External	20/08/2010
<b>Approved by</b>	<b>AMB &amp; PMB</b>		20/12/2010

## III. DOCUMENT LOG

Issue	Date	Comment	Author/Partner
0.0	13/07/2010	Incomplete Placeholder	Michaela Lechner / KTH
0.1	30/07/2010	ToC after input from kickoff meeting	Michaela Lechner / KTH
0.2	05/08/2010	Added input on GOCDB	Gilles Mathieu / RAL-STFC
0.3	06/08/2010	Input on ARC and UNICORE	Michaela Lechner / KTH, Rebecca Breu / FZJ and Michael Grønager / DSMU
0.4	19/08/2010	Input Operations Portal, Globus and more UNICORE	Cyril L'orphelin / IN2P3  Rebecca Breu /FZJ  Anton Frank /LRZ

Issue	Date	Comment	Author/Partner
0.5	20/08/2010	First internal reviews	Michaela Lechner / KTH Tiziana Ferrari / EGI.eu Mario David / LIP
0.6	27/08/2010	Input on Nagios and many comments Functionality descriptions	Mathilde Romberg / FZJ Michaela Lechner / KTH Emir Imamagic
0.7	18/10/2010	Input on GGUS and support functionality description	Torsten Antoni / KIT Michaela Lechner / KTH
0.8	18/10/2010	Input on Accounting	John Gordon / RAL Michaela Lechner / KTH
0.9	19/10/2010	Internal reviews	Steven Newhouse / EGI Michaela Lechner / KTH
0.95	21/10/2010	Internal reviews	Michaela Lechner / KTH John Gordon / RAL
1.0	29/10/2010	First complete draft	Michaela Lechner / KTH
1.1	16/12/2010	Final version	Michaela Lechner, Steven Newhouse

#### 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/>.

## 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**

This document describes and defines the operational interfaces that must be supported for resources to be integrated into the European Grid Infrastructure (EGI).

For each of the operational tools we describe the steps necessary to integrate a new middleware stack into the production infrastructure. This is followed by a detailed analysis of each middleware stack from the EMI and IGE projects and their immediate future plans for operational interoperability.

An overview table [Table 3] shows the general picture outlining the current status of each middleware in relation to the currently existing operational tools. Although, the actual operational tools used within EGI might change in the future.

Based on this table we define a set of requirements to each middleware provider, so that sites running only this specific middleware stack will still be able to make full use of all relevant operational tools in order to be fully integrated with EGI. Requirements can also stem from a more general interoperability point of view.

Finally, this document will give an overview of the status of operational procedures and policies needed for the integration of new resources and conclude with some future plans.

## TABLE OF CONTENTS

<b>1</b>	<b>INTRODUCTION.....</b>	<b>8</b>
<b>2</b>	<b>INTEGRATION OF MIDDLEWARE ON OPERATIONAL TOOL LEVEL.....</b>	<b>9</b>
2.1	Overview Interoperation Status for Operational Tools – Middleware.....	9
2.2	Interoperation at an Infrastructure Level.....	10
2.3	Definition and Description of a Management Interface.....	11
2.3.1	Functionality.....	11
2.3.2	Requirements.....	12
2.3.3	Integration of new Resources into GOCDB.....	12
2.4	Definition and Description of a Monitoring interface.....	17
2.4.1	Functionality.....	17
2.4.2	Requirements.....	18
2.4.3	Interoperability of different MW stacks with Nagios.....	19
2.4.4	Procedure to integrate new Nagios probes .....	22
2.5	Definition and Description of an Accounting Interface .....	22
2.5.1	Functionality.....	22
2.5.2	Requirements.....	23
2.5.3	Current Status .....	23
2.5.4	Integration with other infrastructures.....	24
2.6	Definition and Description of a Support Interface.....	26
2.6.1	Functionality.....	26
2.6.2	Requirements.....	27
2.6.3	Integration of new resources into GGUS.....	27
2.7	Definition and Description of a Dashboard Interface .....	28
2.7.1	Dashboard Interface Functionality.....	28
2.7.2	Requirements.....	29
2.7.3	Operations Portal.....	29
2.8	User management, Authentication and Authorisation .....	33
2.8.1	User management in gLite and ARC .....	34
2.8.2	User management in UNICORE .....	34
2.8.3	User management in Globus.....	34
<b>3</b>	<b>INTEROPERATION AT PROCEDURES AND POLICY LEVEL .....</b>	<b>35</b>
3.1	Scope .....	35
3.2	Current status of EGI procedures and policies .....	35
3.3	Future procedures.....	36
<b>4</b>	<b>OUTLOOK AND FUTURE PLANS.....</b>	<b>38</b>
4.1	Operational requirements coming from NGIs .....	38
4.1.1	Integration of UNICORE and Globus resources.....	38
4.1.2	Integration of desktop Grids .....	38
4.1.3	Integration of cloud services.....	38
4.1.4	Integration of new resources into accounting .....	39
4.2	Requirements coming from Collaborations with other distributed infrastructures .....	39
<b>5</b>	<b>REFERENCES .....</b>	<b>41</b>



## 1 INTRODUCTION

In order to add new resources into the EGI production infrastructure a basic set of operational interfaces that must be supported by the new resources has to be defined and described in their basic functionality.

Different resources will use different middleware components. EGI-InSPIRE will support the Unified Middleware Distribution (UMD) for deployment on the production infrastructure. The UMD integrates middleware components provided by the European Middleware Initiative project (EMI), by the Initiative for Globus in Europe (IGE) project, and other external sources called “Community Contributions”. Services from the gLite, ARC and UNICORE middleware stacks will be included in the EMI release. Within the scope of this document middleware stacks collected in the UMD are taken into account.

Operational tools such as the GOC Database (GOCDB) or the Nagios monitoring tools, are key software components for a reliable and stable operation and monitoring of the infrastructure. The current set of what is considered to be basic operational tools is inherited from the experiences within the EGEE projects. Although, these operational tools may change in the future, they provide a starting point when comparing the interoperability of different middleware components for each operational tool currently in use.

Operational procedures and policies are needed to enforce the application of the agreed basic set of operational interfaces to be supported by all resources. Some of the old EGEEIII procedures and policies are being adapted to the EGI era, while new requirements will have to be identified and turned into new procedures and policies. Special focus shall be made on security.



## 2 INTEGRATION OF MIDDLEWARE ON OPERATIONAL TOOL LEVEL

The EGI-InSPIRE project continues to evolve the blueprint on how to successfully run a federated European Grid Infrastructure as inherited by the EGEE series of projects. A certain amount of rationalization and optimization is necessary to pick up best practice within the community and to create a sustainable model for operating a growing pan-European grid infrastructure that builds on nationally and regionally funded grid initiatives who want to work together.

Availability and reliability measurement, registration of services, information indexing, monitoring, accounting, user and operational support in EGI currently rely on operational tools already developed in the framework of the EGEE project series. Tool development is an ongoing activity and is part of the EGI-InSPIRE JRA1 work programme [R2].

While different middleware stacks are supported by EGI for deployment in the resource centres, the central and distributed instances of the operational tools are operated by a small number of partners committed to provide such services for National or Regional Grid Initiatives, or even for the whole EGI.

EGI will need to deploy several middleware stacks according to the requirements of users and site managers. Presently, as a result of the EGEE and WLCG projects, only gLite is fully integrated into all the operational tools, whilst ARC has been partially integrated, and for Globus and UNICORE operational integration is still incomplete. Comprehensive integration is a short-term objective of the first phase of the project.

In a second phase, it is expected that site administrators and user communities will provide requirements for the interoperability between different middleware stacks, and that EGI will need to integrate new types of resources, such as virtualization, digital libraries and repositories, desktop grids, High Performance Computing, etc. into production.

### 2.1 Overview Interoperation Status for Operational Tools – Middleware

	gLite	ARC	UNICORE	Globus
GOCDDB	completed	completed	to be done	to be done
Nagios - Defining probes which will generate an alarm in the dashboard	completed	completed	to be selected (available in NGI- DE, NGI-PL)	to be selected (available in NGI- DE, IGE)
Nagios - Probes	completed	completed	to be selected (available in NGI- DE, NGI-PL)	to be selected (available in NGI- DE, IGE)
Operational Dashboard	completed	completed	to be done	to be done

	gLite	ARC	UNICORE	Globus
Accounting	completed	completed	not (yet) available	not (yet) available
3 <sup>rd</sup> level support in GGUS (access to expert teams via DMSU and ev. EMI)	completed	completed	completed	completed

**Table 3: Outlining the current status of interoperation for each MW stack relative to the current set of operational tools**

## 2.2 Interoperation at an Infrastructure Level

The basic operational interfaces that must be supported for resources to be integrated into EGI consist of a management interface, a monitoring interface, an accounting interface, a support interface and an additional graphical dashboard interface which collects and presents the information provided by the others and ties them together in a meaningful way to facilitate daily oversight grid monitoring duties.

### MANAGEMENT INTERFACE

An important operational interface of a resource is the capability to be put in downtime if under maintenance, the capability to undergo a certification process and thereby reach production status, and the capability to be monitored to assess its operational security level. Within EGI the GOCDB is the tool of choice for fulfilling these management tasks. It portrays what services are running where and who to contact on a management and technical level as well as in case of security issues.

A first step towards integration of resources is therefore to enable the registration of new types of services provided by these new resources in GOCDB.

### MONITORING INTERFACE

The next step is to describe and advertise the resources in some way using the OGF GLUE2 standard schema [R 14]. This enables a unified view of grids and their resources across each infrastructure, computing centre or federation and thereby enables general monitoring. One possible monitoring tool fulfilling these requirements is for example Nagios, which allows all relevant services to be probed at regular intervals to assess their operation. Such a test execution and notification environment is needed for the fast identification and consequently fast resolution of any problems that may arise. General monitoring of services is also needed to produce availability and reliability figures.

### ACCOUNTING INTERFACE

Accounting ensures the quality of service by providing useful planning and usage information. After all, the key feature of an operational infrastructure is that the resources have high availability and reliability. The summary data of the amount of actually delivered computing resources is relevant for both VOs and project communities as well as for each site to check if their service level agreements have been fulfilled. Current EGI accounting is



based on APEL which is moving to embrace the common messaging interface used by other operation tools and to provide a standard web service interface to read records published by partner grids.

#### SUPPORT INTERFACE

The provision of 3<sup>rd</sup> level support is equally important as the quality of the provided services.

EMI has set up a 3<sup>rd</sup> level structure within the EGI Helpdesk (GGUS) for its various middleware stacks and services. The EGI Helpdesk has been adopted as a common infrastructure to exchange trouble tickets between different stakeholders following its use during the EGEE projects.

#### DASHBOARD INTERFACE

The collected monitoring information is presented through the Operations Portal to give a detailed overview of the operational status. It is possible to contact the sites (as described in GOCDB) and to submit tickets (via the EGI helpdesk) to the site. This dashboard interface thereby eases daily operation and provides templates adapted to the operational procedures that are currently in use.

## ***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 Resources into GOCDB

Resources are stored in GOCDB using the following two basic concepts:

1. **"Service types"**, which represent generic components deployable on the Grid infrastructure. They can be middleware components (e.g. CE, WMS, SRM...) or components specific to the operational infrastructure (e.g. MessageBroker, RegionalNagios...).
2. **"Service endpoints"**, which represent deployed instances of a service type.

In order for new resources to be integrated into GOCDB, the type of these resources has to be integrated first as "service type", and then the deployed instances of this service type can be declared.

Because of that model, integrating new resources to GOCDB does not require any development effort.

It is a matter of adding the proper set of information to the existing system as described in the following sections.

#### ***2.3.3.1 Integration of new MW service types***

New middleware service types can either be new services from an already deployed middleware, or services from a new middleware stack.

In the first case, the proposed procedure is as follows: EGI-InSPIRE JRA1 gets informed by known middleware providers (e.g. EMI) about new services that have been added to an existing middleware stack that is deployed and used within EGI.

If the new services belong to a previously undeclared middleware stack then strategic decision has to be made about the validity of the request. This is to ensure that only officially supported middleware stacks are actually integrated into GOCDB.

Requests for adding new service types in GOCDB will first go through OTAG who will collect, discuss and refine the requests before approving integration. OTAG consists of all relevant members able to take a technical decision on the integration of new service types. New service types are then added to GOCDB and are made available to declare new resources as described in 2.3.3.3.

A more flexible and abstract definition of a service type depending on its function and not the actual middleware used would be desirable.

The naming scheme for new middleware service types should be kept consistent with the service type name as used in the information service of the technology provider.

#### ***2.3.3.2 Integration of new non-MW resources types***

There is a need to store and present information about different types of services, such as those deployed for operations or at the application level. Application services are deployed by grid sites to support certain VOs without belonging to a specific middleware distribution.

After approval of the Operations Management Board (OMB), new non-middleware services are integrated into GOCDB in a similar way to the middleware services, apart from the fact that the initial information does not come from the software provider but from either the EGI-InSPIRE developers' community, JRA1 itself (in case of this being a service for operations management) or from a user community (in case of an application specific service).

The way to deal with the request and eventually integrate the new type to GOCDB is similar to what is described in section 2.3.3.1.

### **2.3.3.3 Declaration of new resources of an already available resource type in GOCDB**

Once a service type is integrated into GOCDB, instances of this service can be declared as service endpoints. This is done by the resource providers (i.e. administrators of the site hosting the endpoint, regional managers, operations staff). A complete description of the process is described in the GOCDB user documentation [R 6].

### **2.3.3.4 Regular review of the list of available service types**

The normal evolution of any infrastructure and middleware stack means that some service types will become obsolete with time. To ensure the accuracy of information available in GOCDB, a regular review of the list of available service types will be made. This task will be under the responsibility of GOCDB developers, who will consult the Technical Coordination Board (TCB) (consisting of the Software Providers like EMI, IGE, EGI-JRA1, etc.) together with the OMB to get a list of service types that are candidates for decommissioning.

### **2.3.3.5 Summary of the complete procedure**

The complete procedure to have new resources integrated to GOCDB is as follows:

- If the service type is already available in GOCDB, service endpoints can be added following the established procedures [R 6].
- If the service type is not available, a request has to be made in the OTAG queue and if approved it is communicated to the GOCDB developers to add the new service type.
- Notification of the requesting party whether the request was disapproved or completed.

### **2.3.3.6 Integrating gLite resources in GOCDB**

Current gLite resources are integrated. New resources are added according to the procedure as described above.

Currently listed and not yet obsolete service types used by the gLite middleware stack in GOCDB:

- **CE:** [Site service] *The LCG Compute Element. Currently the standard CE within the gLite middleware stack. Soon to be replaced by the CREAM CE.*
- **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 EGI sites must have one instance of this Service Type, associated with a CE.*
- **MON:** [Site service] *The gLite MonBox hosts the site R-GMA services.*
- **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-BDII's and publish the data. Only a few instances per region are required.*



- **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.*
- **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.*
- **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.*
- **MyProxy:** [Central service] *The My Proxy service is part of the authentication and authorization system. Often installed by sites installing the WMS service.*
- **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.*
- **RGMA-IC:** [Central service] *This is the Registry for an R-GMA service. There will only ever be a few of these per grid.*
- **MSG-Broker:** [Central service] *A broker for the EGI central/backbone messaging system.*
- **Site-NAGIOS:** [Site service] *A site-level Nagios monitoring box*
- **National-NAGIOS:** [Regional Service] *An NGI-level Nagios monitoring box*
- **Regional-NAGIOS:** [Regional Service] *A ROC-level Nagios monitoring box*
- **Project-NAGIOS:** [Central Service] *A project-level Nagios monitoring box*
- **gLite-APEL:** [Site service] *The gLite-APEL hosts the site Accounting client (gLite 3.2 replacement of the MonBox)*

### 2.3.3.7 Integrating ARC resources in GOCDB

ARC resources have been part of GOCDB since 2007. This has happened even though the Nordic infrastructure using the ARC middleware was not formally an EGEE partner. The ARC integration can therefore serve as a role model on how to integrate other middleware stacks. In the beginning a lot of services were already common to gLite, such as dCache storage elements [R 7]), LFC catalogue service [R 8], VOMS [R 9], etc.

However, the ARC method of dynamic service indexing, the ARC GIIS and the ARC-CE were not supported in GOCDB. The ARC-CE was added as a new Compute Element service type. Afterwards the first ARC-CEs could be registered under the NDGF-T1 site in GOCDB.

For the indexing of services another solution was chosen. ARC had applied the Globus Meta Data Service consisting of top level GIIS and site level GRIS services [R 12]. In order for these resources to be visible a special BDII [R 13] had been set up for the NDGF-T1 site which dynamically collected the content of the GRIS'es of the ARC-CEs based on the list of CEs provided by the GIIS'es. 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].

Nowadays new resources are simply added according to the procedure as described above.

#### **2.3.3.8 Integrating UNICORE resources in GOCDB**

Not all needed middleware service types have been defined yet in GOCDB.

A list of service types that need to be defined follows. The different service types are typically installed on separate machines, but don't necessarily need to be.

- **Gateway:** Sits in front of one or more UNICORE services as a gateway to the internet. Normally one Gateway per site. Has been added as a service type.
- **Registry:** All UNICORE services register here; clients ask the registry for available services in the grid. Normally there's one Registry per grid infrastructure with optional backup instances. The Registry works like a phone book and collects URLs of services. This has already been added as a service type.
- **Workflow Engine:** Needed to add workflow functionality to UNICORE. It is not needed if only single jobs are submitted within a grid infrastructure. Normally there's one Workflow Engine per grid infrastructure.
- **Service Orchestrator:** Handles dispatching of a workflow's subjobs, and brokering. Normally there's one Service Orchestrator per grid infrastructure.
- **UNICORE/X:** Hosts the XNJS, which handles job submission, file transfer, job monitoring etc., and the Common Information Provider CIP. One UNICORE/X per supercomputer/cluster. Has been added as a new service type in GOCDB, but requires that an additional attribute, namely the endpoint URL can be added to the service within GOCDB. A new feature in GOCDB has been requested to support this.
- **CIS:** Information service. Standalone service which collects information from the UNICORE/X. One per grid infrastructure.
- **XUADB:** User database. Maps the X.509 certificates to a user's login accounts and roles. Services like the Workflow Engine and the UNICORE/X query the XUADB for authorization. There is great flexibility in how many instances are needed per grid infrastructure; each site running their own XUADB seems to be the most common deployment option.



- UVOS: UNICORE VO Service. Serves the same function as XUADB but is much more advanced and flexible by supporting arbitrary attributes, groups, advanced authorization, and more. Usually one per grid infrastructure, but may be replicated.
- Target System Interface (TSI): The actual interface to the local batch system; submits jobs and goes with the UNICORE/X.
- SIMON: Standalone service which monitors UNICORE sites, mainly by periodically sending test jobs.

Some of these services are quite tightly coupled, and are not visible as separate services to clients, nor can they be tested separately. Thus it might not make sense to separate them when integrating them. A more detailed view on UNICORE architecture can be found in [R 32].

### **2.3.3.9 Integrating Globus resources in GOCDB**

The three most important service types for Globus which have been registered into the GOCDB are:

1. job submission service for Globus version 4.0.x, 4.2.x (WS-GRAM) and 5.x (GRAM5).
2. storage endpoint and data transfer service for the Globus middleware stack (GridFTP).
3. certificate based interactive login service (gsisshd).

Used ports can differ from the default, thus the registration of the port must be possible as well.

Other non-middleware specific service types that are already defined and can be used together with gLite and ARC like e.g. MyProxy are not listed. Further Globus services that are candidates to be supported in GOCDB are:

- MDS: Monitoring and Discovery Service for Globus Resources
- WebMDS: Web interface to MDS information
- RLS: Replica Location Service
- RFT: Reliable File Transfer

## **2.4 Definition and Description of a Monitoring interface**

### **2.4.1 Functionality**

A monitoring interface monitors the resources presented within EGI to ensure the infrastructure's reliability and to quickly find causes of failure. Ideally, actual failure is avoided by fine tuning the tests so that warnings about any required maintenance can be sent before failure actually occurs.

Tests to monitor all mission-critical infrastructure components have to be defined and implemented as probes. A subset of probes will be able to raise alarms in the dashboard and are flagged accordingly. In the event of failure, notifications of the possible problem together with hints on how to solve the problem are sent to the technical staff and other relevant

people allowing them to work on the problem before outages affect production and availability.

Alerts and warnings are delivered to IT staff via email and SMS, depending on the site managers' choice. Multi-user notification escalation capabilities ensure alerts reach the attention of the right people.

The execution of probes can be rescheduled to test the solution of a problem.

Statistical data is collected to provide input for the availability and reliability figures to see if OLAs are fulfilled and production level is reached. Only the subset of test results creating alarms in the dashboard are considered for the computation of monthly availability and reliability statistics. Users and operators are informed about the state of the grid.

The design of the monitoring interface is scalable and some fail-over support is possible.

A good monitoring system monitors not only the network and the resources, but also the accessibility and functionality of the used operational tools.

The set of Nagios-based monitoring services necessary at NGI and central level provided within EGI is called the Service Availability Monitor (SAM). This Nagios based monitoring framework has replaced the former centralized SAM submission framework developed by the WLCG Grid Service Monitoring working group. In addition to the monitoring infrastructure it is necessary to have documented procedures that can be followed by the support teams to maintain a reliable infrastructure. One such aspect is the generic probe profile (used as the basis for availability calculation) which has to be checked at regular intervals to ensure it is up-to-date. This includes regularly reviewing the current set of probes to check that they fulfil all the operational needs or to see if they need to be extended, reorganized or new probes need be provided.

## 2.4.2 Requirements

- Regionalization is an important factor since grid in its nature is a distributed system. Monitoring should therefore be split into various instances running in each region and a central instance collecting results. From the technical perspective the distributed system contributes to increased scalability as each instance covers a smaller number of sites than a single central instance. From the operational perspective, the NGI teams get much more control and responsibility over the whole monitoring process since customization of the national monitoring infrastructure is under the responsibility of the NGI. This way, central problems no longer impinge local monitoring and response time should decrease by shortening the length of the reaction chain and removing a possible bottleneck. Finally, a distributed system enables individual instances to tune the monitoring by introducing extended custom probes to monitor custom services not covered by the generic profile. Also, individual instances can benefit from additional functionalities of the monitoring system such as direct email or text message notifications, extending monitoring on uncertified sites or direct scheduling of tests via a web interface.

- Status and historical data should be accessible in a centralized portal. These historical records of outages, notifications, and alert response are relevant for later analysis.
- The monitoring interface should also provide a component to calculate resource availability – a figure that makes allowances for notified downtimes.
- Information shall be exchanged according to a given template and using a common transport mechanism (ActiveMQ).
- It shall work as an input plug-in for the Operations Portal.
- Additionally it would be desirable to not only monitor the resources but also the availability of the needed operational tools, such as the different regional monitoring instances.

### 2.4.3 Interoperability of different MW stacks with Nagios

Nagios [R 38] is a well-known and mature general purpose monitoring system that enables organizations to identify and resolve IT infrastructure problems.

Out of the box, Nagios can already monitor many different infrastructure components - including applications, services, operating systems, network protocols, system metrics and network infrastructure. Furthermore, its extensible architecture allows easy integration with in-house and third-party applications. Hundreds of community-developed add-ons extend core functionality to ensure a faultless functioning of the entire infrastructure. New tests to monitor further mission-critical infrastructure components can be defined and deployed with freshly written probes for them.

Within EGI the central instance of Nagios collects the results and provides a centralized MyEGI portal [R 39] to access status and historical data.

A special Nagios box was established at CERN with the purpose of monitoring the ActiveMQ Brokers network and Nagios instances. CERN developed probes for monitoring these two services. CERN committed to run this instance during the EGI-InSPIRE project. The ops-monitor Nagios instance can be found on the address provided in [R 40]. Other operational tools developers were requested to provide probes for monitoring their tools as well. Once the probes are provided, they will be integrated into the ops-monitor Nagios instance.

SAM/Nagios instances are supposed to be deployed at each NGI.

To integrate a new middleware stack into Nagios, sensible tests for the service types defined in the management interface for this middleware have to be developed. These tests need to be included in new Nagios probes so that they cover the important functionality in the middleware stack. For that the subset of probes which should raise alarms and have an influence on the reported availability and reliability metrics has to be defined. It may be sufficient to just have a compatible Nagios reporter from a different kind of monitoring tool which can be integrated in regional and central instances.

Since SAM Update-07 release (30<sup>th</sup> November 2010) SAM fully supports using ATP as a topology provider. ATP is currently fed with information from both GOCDB and BDII. ATP

extracts VO mappings from the BDII as those are not present in GOCDB. When EMI provides a single information system which will integrate all supported middlewares, it will be integrated into ATP.

#### **2.4.3.1 Tests and Nagios probes for gLite resources**

Currently, the Nagios probes for the following service types needed by gLite are implemented:

- APEL
- BDII (top and site BDII)
- CE
- CREAM-CE
- FTS
- gRB/WMS
- LB
- Local-LFC/Central-LFC
- MPI
- myProxy
- RGMA-IC/MON
- SRM
- VO-box
- VOMS

Further documentation and descriptions on these probes can be found on [R 42].

#### **2.4.3.2 Tests and Nagios probes for ARC resources**

Historically Nagios' predecessor the former Service Availability Monitoring framework, SAM, was the first EGEE infrastructure service to interact with ARC services. Every hour SAM executed tests against the different sites registered in the GOCDB by querying the individual services listed in the site BDII. SAM tests for index, storage, catalogue could be run right from the start. A new sensor suite in the modular SAM was developed for the new Compute Element service type ARC-CE. The WLCG Management Board and an extra working group made sure that the tests for the different CE types compare and a fair and balanced translation between the different CE tests was ensured. The transition towards Nagios based monitoring was done during EGEE III for both ARC and gLite.

Nagios ARC probe developers prepared a set of Nagios probes for ARC which they tested in an independent local Nagios instance and started integration with SAM. The problem at the start of the integration was that these procedures had not yet been prepared. Since the

integration of ARC probes was already approved beforehand, an RT ticket was created directly in the JRA1 queue and further progress was followed through the developers ticketing system [R 47].

The first task was to prepare RPM packages consistent with other probes integrated in SAM [R 48]. Probes were prepared and built by ARC developers after they were given remote access to the required build environment.

The second task was to integrate probes into SAM. The description and details for the set of ARC probes had to be added as a profile to the Metrics Description Database (MDDb) and the configuration generator component (NCG).

The biggest issue encountered during integration was the lack of a multiple middleware UI. Currently, it is not possible to install both glite UI and ARC Client by using packages on the same machine. Therefore an alternative approach based on an ARC Client standalone package was chosen and implemented. This requires Nagios administrators to perform additional steps described in [12].

The SAM Update-07 released on November 30<sup>th</sup> 2010 contains ARC probes. The release has passed staged roll-out on December 6<sup>th</sup> and once approved by operators these probes will be used for availability and reliability calculation. Until then availability and reliability of ARC sites will be calculated based on results from tests run by the traditional SAM system.

#### **2.4.3.3 Tests and Nagios probes for UNICORE resources**

The Site Monitor for UNICORE resources (SIMON) [R 30] is a standalone service which submits various kinds of UNICORE test jobs to check the availability of the UNICORE stack. This UNICORE monitoring tool SIMON acts as a user, thus needs its own certificate, login and entry in the UNICORE User Database. SIMON can also report to Nagios. PL-Grid is doing the work on integrating UNICORE SIMON probes into EGI Nagios/SAM. PL-Grid defined a number of useful tests and their dependencies [R 31], as well.

UNICORE's Common Information Service (CIS) [R 26] provides detailed information about the underlying system, e.g. the number of CPUs, memory, number of running jobs etc. according to the OGSA standard GLUE2 information model [R 14] for representing resource information.

Nagios is already used in D-Grid (the German e-Science grid) for testing UNICORE resources. All UNICORE services except the CIS are considered as mission-critical.

#### **2.4.3.4 Tests and Nagios probes for Globus resources**

For Globus the availability of the servers for central services (RFT, MyProxy, MDS/WEBMDS) and of the services at the resources (GSI-SSH, GridFTP, WS-GRAM, GRAM, etc.) are considered as mission-critical.

Various Nagios probes have been developed in the scope of D-Grid/NGI-DE and DEISA.

Currently, the following Nagios probes which should raise alarms are available:

- Globus service availability (GSI-SSH, GridFTP, (WS-)GRAM)
- GridFTP server availability test
- WS-GRAM (Globus v. 4.0.x) job submission test
- GridFTP file transfer test
- Globus container certificates (availability, lifetime)
- Globus container memory consumption
- RFT PostgreSQL DB
- RFT transfer test
- Globus WebMDS status
- Globus WebMDS HTTP response
- Version check of IGTF CA distribution
- Host certificate validity life-time check

It has to be checked to see if these Nagios probes can be used as is or if they need to be adjusted to the EGI requirements. A first GRAM5 Nagios probe is now ready and about to be tested.

#### **2.4.4 Procedure to integrate new Nagios probes**

A new procedure on how to integrate new Nagios probes is currently in development outgoing from the current experiences with the tickets in RT in the 'inspire-jra1' queue about the integration of ARC, UNICORE, GLOBUS5 and EDGI probes into SAM/Nagios:

- <https://rt.egi.eu/rt/Ticket/Display.html?id=201>
- <https://rt.egi.eu/rt/Ticket/Display.html?id=306>
- <https://rt.egi.eu/rt/Ticket/Display.html?id=390>
- <https://rt.egi.eu/rt/Ticket/Display.html?id=461>

The Experience with the integration of ARC probes as described in 3.4.3.2 will be especially helpful in defining the final procedure for further integration. Furthermore a comprehensive list of all tests a site should pass is being collected in [R 42].

## **2.5 Definition and Description of an Accounting Interface**

### **2.5.1 Functionality**

The EGI Accounting Infrastructure collects CPU accounting records from sites and/or grid infrastructures and summarizes the data by site, date (especially by month), VO, and user. This summary data can be displayed in a dedicated Accounting Portal by dynamic queries on the parameters above at any level of the hierarchical tree structure which defines EGI and its partner grids.



Accounting is necessary to demonstrate that the usage of resources by user communities is in accordance with expectations. Site administrators are able to check actual usage of CPU resources against scheduling policies implemented at the site. VO resource managers are able to understand how CPU resources are utilized by their users.

When looking at the accounting interface as the interface between the accounting services of different interoperating infrastructures the main aim is to enable all the accounting data of a VO to be collected in one place. This is assumed to be delivered by the exchange of accounting data at the appropriate level.

### 2.5.2 Requirements

An accounting interface has to fulfil the functionality described above. Further requirements are:

- Access to accounting data needs to respect all relevant policies and legal requirements. It is expected that this is controlled by the standard user authentication and authorization framework.
- Data identifying an individual should not be sent across the wide area network in plain text.
- As data from different grids is to be combined, the units of measurement should be understood and manipulated appropriately.

### 2.5.3 Current Status

The EGI Accounting Infrastructure is largely based on APEL [R 34]. The collected CPU accounting records and the data summarized by site, date, VO, and user are displayed in the Accounting Portal [R 43] by dynamic queries on the parameters above at any level of the hierarchical tree structure which defines EGI and its partner grids.

The bulk of existing sites collect data from their batch systems (LSF, Torque; SGE, Condor), which are joined with the job's user grid credentials and published to the central APEL repository. At the time of writing the EGI infrastructure is in transition of the transport layer from R-GMA [R 33] to ActiveMQ already used by other EGI Operational Tools. Other partner grids (Open Science Grid and NDGF), and a few sites with their own accounting services, publish summaries of data in the form described above directly into the APEL central repository. Sub-grids of EGI (e.g. Italian Grid Infrastructure IGI) publish all of their VOs data. Partner grids (e.g. Open Science Grid OSG) publish selective VOs. In particular the LHC VOs are all published to APEL so that there is a single worldwide repository for LHC. At the time of writing, summary publishing is done by remote database insertion but an ActiveMQ summary publisher is under development.

CPU data is published in the form of either: job level records containing data from a single batch job; or summary aggregate records containing totals for a number of jobs run at a single site for a single user and VO in a given month. The Job User Record (UR) schema is a plain text version of the OGF-UR v1.0 with some common extensions. For example, the original UR did not have the concept of a site, which is so crucial to the grid. The summary

record has been submitted to OGF's UR-WG for possible adoption as a community standard [R 35].

In addition to the ActiveMQ route for receiving and transporting data, the APEL development roadmap plans to also implement a RUS [R 44] standard interface to receive data.

## 2.5.4 Integration with other infrastructures

Other grid infrastructures who wish to publish accounting data need to:

- a) Define a structure for their grid in GOCDB (or equivalent) that can be used by the accounting portal to display the data. The minimum requirement is a flat set of site names, used in the accounting records. (e.g. for OSG these data are obtained from MyOSG)
- b) Extract data from their accounting system grouped data by site/VO/User/FQAN/month and create each group into a 'summary record' meeting the APEL definition. Experience shows that for accounting systems using the OGF-UR this is a simple transformation.
- c) Other infrastructures running a gLite CE (lcg-CE or CREAM) could run our software to aid collecting accounting data. Infrastructures running other middleware stacks who run one of the currently supported batch systems listed above can take our data collectors to parse the raw accounting data collected by the batch system to which they will then need to add the CPU speed and user/VO credentials, before publishing.
- d) Register the publisher with APEL (by providing the host DN to the EGI APEL support unit). The APEL Repository only accepts accounting records from registered sites. For APEL client sites this is defined by the glite-APEL service type in in GOCDB. An equivalent mechanism will be developed for summary publishing sites/grids.
- e) Publish the records into EGI's ActiveMQ Message Bus. The APEL repository will accept the records into a holding container from where they will be merged with the summaries from other grids and the summary produced by APEL from the job records it has received. Currently, the master summary is rebuilt from scratch several times per day. Each time it uses the last set of summaries received from each grid.
- f) From the master summary table, the data are then exported to CESGA where they can be viewed in the accounting portal.

### 2.5.4.1 Issues

- For the aggregation of user data it is assumed that all interoperating infrastructures use a user identity based on X.509 certificates signed by IGTF recognized Certificate Authorities.
- While a worldwide community management service like VOMS makes the aggregation of VO accounting data from different infrastructures simple, it would be feasible to implement a VO name transformation to combine the data from infrastructures who have named the same VO differently.
- Another issue is the unambiguous mapping of user accounts to VOs. In some cases users might belong to more than one VO in which case identifying to which VO the utilization



results would go is not possible. Extra effort will be needed to check the fulfilment of arranged pledges.

- The issue of exchanging data identifying a user has been a contentious one. It is frequently asserted that this is illegal under the laws of certain countries. Extensive research was undertaken by the Joint Security Policy Group (JSPG) in EGEE-III during the development of the Grid Policy on the Handling of User-Level Job Accounting Data [R 45] with the result that legal advice was given that with the appropriate acceptable use policy and the agreement signed by the user and by the site running the accounting repository, then the collection, storage and restricted display of data identified by UserDN is acceptable. This issue might have to be re-evaluated again when exchanging accounting data with other infrastructures like e.g. DEISA.
- Current accounting is only of CPU of batch jobs but the interfaces between infrastructures should also allow the integration of other types of accounting record as they are developed. New accounting types should ideally be developed by all the infrastructures working together.
- The currently agreed unit for normalization of CPU time in EGEE, EGI, and WLCG is HEPSPC06 hours [R 46]. For interoperation with an infrastructure that does not collect this value from the resources running jobs, some conversion factor must be negotiated.

#### **2.5.4.2 Future Work**

At the time of writing the ActiveMQ interface into APEL only accepts a single type of job record for the CPU used by a batch job. The summary development mentioned above will include handling multiple types of record. As well as the summary record this will allow the repository easily to be extended to support other types of accounting, such as storage, as well as allowing evolution of the CPU UR. New accounting types should ideally be developed by all the infrastructures working together.

The RUS interface planned in APEL will allow other grid infrastructures to use a standard web services interface to publish records. This will replace item (e) in the integration list above.

#### **2.5.4.3 ARC resources**

Accounting integration was performed already during EGEE III. The aim was to gather and export accounting from the Nordic T1 and T2s, which for the compute part were based on ARC, sorted per VO to the EGEE Accounting Portal. The EGEE Accounting Portal used the APEL database as back-end, and direct DB insertion is provided per site. ARC-CE supports accounting via SGAS (SweGrid Accounting System, [R 19]) and an automatic script for exporting the accounting info gathered in SGAS to APEL was set up [R 20]. Currently, only LHC VOs are published to APEL but this could easily be extended to other international VOs.

#### **2.5.4.4 UNICORE resources**

Currently no means of collecting accounting and usage records are directly implemented within UNICORE. Instead, this is done directly via the underlying batch system, see for

example as in the DEISA project, where the accounting data is converted into OGF-UR format and provided according to XUUDB access control.

#### **2.5.4.5 Globus resources**

Globus currently has no accounting software. Accounting statistics for Globus can be obtained indirectly through the underlying batch system. There were efforts of adopting DGAS for Globus in the scope of D-Grid. It was also planned to use OGF-UR there (which was unfortunately not yet provided by DGAS at that time). Nowadays DGAS is part of the EMI project which is aiming to harmonize components to a common standard set. EMI has now in their roadmap agreed on a common usage record based on the OGF-UR standard with extensions.

## **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 1<sup>st</sup> 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 3<sup>rd</sup> 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 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 Integrating 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 Integrating a new NGI in 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 support 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. No general rule how this will be done 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 GGUS for its various services, including e.g. ARC or UNICORE. For details refer to the EMI Milestone 17 [R 3] on the integration of EMI support units into GGUS or the EMI software maintenance and support plan [R 21]. For example, in UNICORE, problems that can't be solved within EGI or EMI will be relayed to UNICORE's bug and feature tracker [R 18] or to the support mailing lists [R 28].

3<sup>rd</sup> 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. GGUS already contains a queue to forward 3rd level support tickets directly to the IGE user support team.

## ***2.7 Definition and Description of a Dashboard Interface***

### **2.7.1 Dashboard Interface Functionality**

In order to operate a distributed infrastructure, management and monitoring information has to be collected and presented to ease the work of the operators of the infrastructure. The dashboard interface combines and harmonizes different static and dynamic information and enables the operators to react on alarms, interact with the sites, and provide 1<sup>st</sup> line

support, as well as to really operate the sites and to supervise the creation and the work on problem tickets on a regional and central level.

The dashboard allows predefined communication templates and is adaptable to different operational roles (1<sup>st</sup> 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 after several severity criterions to give an impression of not only one service put over needed actions 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 [R 25] (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 like e.g. GLUE2 based BDII, independently of the middleware stack used.
- Access should be regulated by a harmonized user authentication service like VOMS or something better (see also detailed discussion in section 3.8).

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 Operations Portal

The Operations Portal [R 23] 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 [R 24].

The goals of Lavoisier 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).

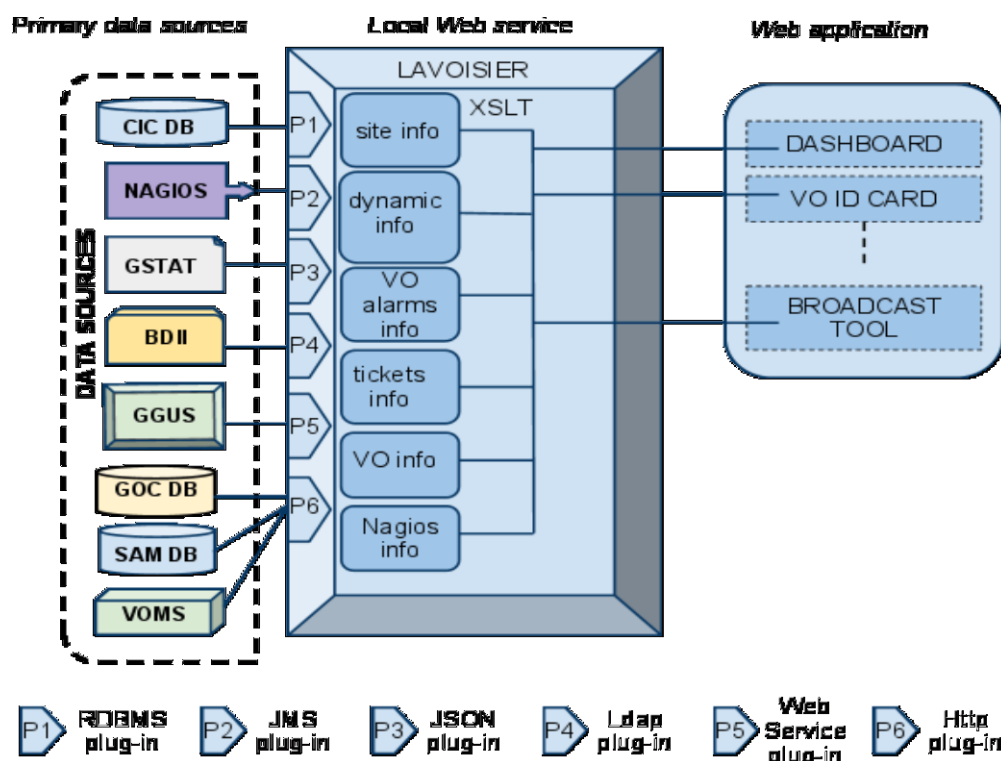


Fig. 1: Global architecture of the Operations Portal.



### **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 [R 25] (OGF) into a Lavoisier plug-in which will permit to access information using different services (like the information service of UNICORE – CIS [R 26]) and different schemas like CIM [R 27] or GLUE Schema [R 14] 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

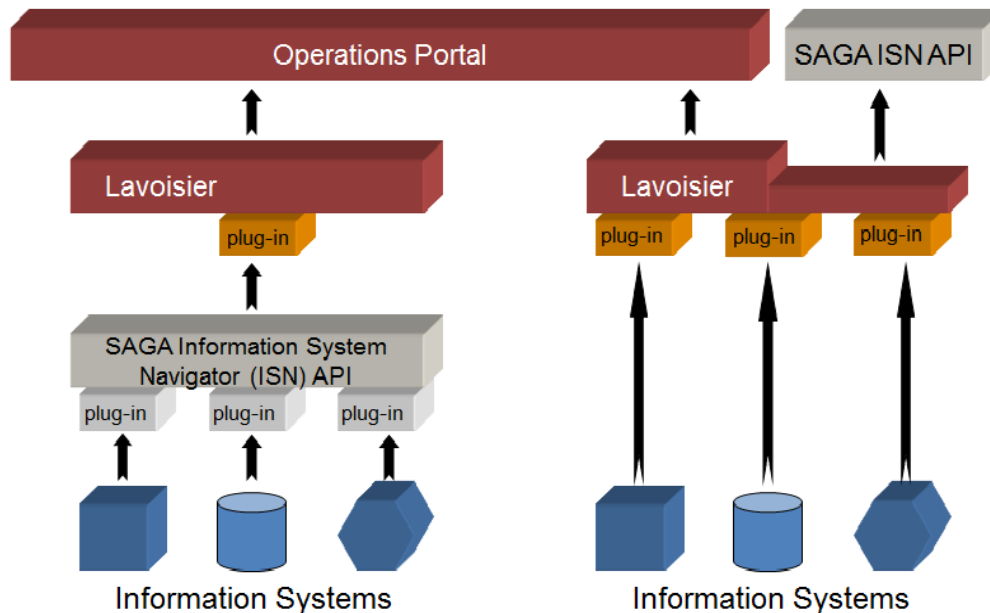


Fig. 2: Integration of new information systems into the Operations Portal

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 gLite resources in the Operational Dashboard

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 ARC resources in the Operational Dashboard

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

### 2.7.3.5 UNICORE resources in the Operational Dashboard

There are different ways to integrate UNICORE resources. Currently, UNICORE resources have to first be registered in GOCDB and monitored by Nagios with the EGI topology and the hardware GLUE information taken from the Central Information Service CIS over the SAGA ISN API link.





An alternative to collecting monitoring information from the EGI Nagios could be to collect it from SIMON Nagios output instead.

#### **2.7.3.6 Globus resources in the Operational Dashboard**

With the eventual deployment and use of Nagios probes the operational alarms from Globus resources or central servers can be directly integrated into the operational dashboard.

### **2.8 User management, Authentication and Authorisation**

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 also 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 the primary authentication token is the X.509 certificate and its proxy derivatives. Every user requests 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 Virtual Organizations (VOs). Access to 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 is 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. To make life easier for the users within EGI a central service would be needed where users apply for an EGI user account (within a VO) and then the accounts are created at the resource providers sites. Otherwise user would have to apply at each site for a user account and each site would have to generate the proper mappings. On the other hand, this new requirement might create clashing userID and adherence problems to different universities'/centres' naming schemes.

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 XUADB.

In EGI, for comparison, information about which VOs are allowed on which resources is published by the sites' BDII via GLUE. The resulting GlueVO\* attributes in the LDAP stream of the BDIIs are collected and visualized by different tools like GSTAT. GOCDB just has a reference to GSTAT and the full GIIS LDAP links as needed to get information from the site, but is not directly collecting and showing this information.

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.

### **2.8.1 User management in gLite and ARC**

VOMS is used for VO and user administration.

### **2.8.2 User management in UNICORE**

The X.509 certificate based UNICORE User Database (XUADB) stores the mapping of user certificates/DN's to local userIDs and roles at a single UNICORE site. The XUADB is a site local authorization component, maintained by each site. It is a Web service in itself and thus can be used from multiple UNICORE installations. These XUADBs have to be filled with the information of those users who are authorized to use the site's resource(s). Full X.509 certificates are used as base line, while the access control is based on XACML policies. Proxy certificates are not used in UNICORE, they are optionally supported in UNICORE6 to e.g. use GridFTP. Technically, it doesn't matter who manages the XUADB user database. Every site can set up their own XUADB and an independent way of managing it, or there could be a central XUADB, or a central service that generates input for each site's XUADB like it is done within D-Grid, which might be a good example for a more embracing authentication scheme.

In DEISA, on the other hand, users who have been granted compute time on a specific subset of DEISA resources apply at one of the DEISA sites for an account with their certificate. The DEISA user information is collected in LDAP servers at the different sites that get synchronized once per day. Each site generates the input for its XUADB from its local LDAP server. DEISA user management is described in detail in [R 29].

As an alternative to the XUADB, a VO service can be used for user authorization. The UNICORE VO Service (UVOS) uses the SAML standard and offers a wide variety of features.

### **2.8.3 User management in Globus**

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

### 3 INTEROPERATION AT PROCEDURES AND POLICY LEVEL

#### 3.1 Scope

The technical set up of individual infrastructures has been described in the previous sections. In order for researchers to enter into European collaborations which require reliable, predictable and consistent access to other infrastructures it is necessary to define further operational procedures.

Compliance to procedures and policies is therefore important to ensure seamless interoperation of operations across EGI. The importance of having procedures and best practices that are valid for all project partners and operations teams cannot be overemphasized. Precise definitions are needed to guarantee that OLAs are fulfilled, which in turn is a precondition for a high quality and stable production environment.

We have to make sure that the actual procedures that guarantee the aspired quality of service are independent from any actual operational tool used as well as middleware agnostic. The procedures should be unified and collected to a common core that can be completed with further, more explicit extensions including adaptations to specific environments and needs of different NGIs. On a smaller scale such an approach is already applied successfully in the security context where several infrastructure providers agreed within the EGEE Joint Security Policy Group (JSPG), [R 5] on a common procedure documents which were to be kept in sync. Different infrastructure providers like e.g. DEISA adopted them and eventually amended them with several add-ons. Especially successful was the Acceptable Use Policy (AUP). It should be in the scope of the Infrastructure Policy Group (IPG) to regularly update these documents and ensure a high degree of communication between the different project partners.

#### 3.2 Current status of EGI procedures and policies

An overview of the procedures, policies and best practices inherited and already improved, changes needed can be found in MS408 [R 10]. The milestone lists procedures taken over from EGEE, new procedures already passed through OMB and in effect and procedures in various draft stages.

One procedure in MS408 explicitly worth mentioning, since it has a great impact on the integration of new resources into the monitoring interface and the quality assurance of those new production resources, is the procedure for turning a Nagios test into an operations test. This procedure defines which tests are able to generate a notification in the dashboard in case of error and which are used to calculate the availability league table.

Security procedures are handled separately.

Potentially new players when adding new resource types have to be aware and follow the procedures published by the Security Policy Group (SPG), [R 41], namely the Grid Incident Response Policy referenced in MS405 [R 1] and the operational security procedures,

especially the security incident procedure and the software vulnerability issue handling process collected therein.

In the deployed EGI infrastructure all problems concerning security should be dealt with between the EGI Computer Security Incident Response Team (CSIRT) and the EGI Software Vulnerability Group (SVG), [R 41]. CSIRT advises the sites on security matters and has the power to suspend sites from the infrastructure if they fail to apply critical security patches. EGI Incident Response Task Force (IRTF) makes sure that incidents are handled according to the Incident Response Procedure. The SVG should ensure that the software available for installation on the EGI infrastructure is sufficiently secure and contains as few vulnerabilities as possible, thus reducing the likelihood of incidents.

When introducing a resource of a new software provider the contact details of the support of that software provider have to be pointed out to the SVG. Those new software providers should become part of EGI UMD and sign the corresponding SLA [R 17]. All these issues are covered in detail in the software vulnerability issue handling process part of MS405.

When adding new resource types to the infrastructure, new people with expertise in these resources will be asked to join the Risk Assessment Team (RAT). This is the group of people within the SVG who carry out the issue handling process of the SVG, and are party to information on vulnerabilities which have not been disclosed publicly. The RAT members typically consist of developers from the various software provider teams whose software is included in the EGI UMD, NGIs and experienced site administrators.

### **3.3 Future procedures**

The operational procedures used within EGI have been evolved from those developed within the EGEE series of projects and now have broad community support and adoption. For example, all procedures and related operational work flows are directly reflected within the Operations Portal. As a result, the portal has to be regularly updated as the procedures change to reflect the needs of the community. Input on non- or only partially working procedures, together with suggestions for improvements, are identified from many sources. One important source of feedback comes from the review of the operational activities that takes place during the weekly handover between operational support teams who use these procedures on a daily basis through their implementation in the operational tools such as the Operations Portal. The collected best practices within the community represents a further source of feedback.

Future operational procedures will continue to move away from relying on personal communication channels and instead on documented communication processes, such as those provided by mailing lists or tickets. To date within EGI the site suspension procedure has been formalised and clear guidance has been formulated on how to correctly interpret and use the AT\_RISK/warning downtimes as defined in the existing downtime procedures.



However, what is clearly needed is to provide an up to date quick reference sheet for procedures (aka cheat sheet) for site administrators and other players to keep an overview of current valid procedures and where to find them. An early result of this work (from TSA1.8) will be found in MS408 [R 10].

## 4 OUTLOOK AND FUTURE PLANS

The functionality descriptions and the respective requirements of the different operational tool interfaces described in this milestone will improve over time.

Operational requirements will continue to be collected from NGIs that are interested in integrating novel resource types into their e-Infrastructure as required. Input from infrastructure providers planning to operate different middleware stacks will be gathered. In parallel to this, the integration with other Distributed Computing Infrastructures will likely bring new requirements for the extension of the operational interfaces currently deployed in EGI for monitoring, accounting, communication, management and support. All this will be documented in the next edition of this milestone in 12 months' time.

### 4.1 *Operational requirements coming from NGIs*

#### 4.1.1 Integration of UNICORE and Globus resources

Various NGIs with short-term integration requirements will be identified and dedicated meetings will be organized. The German NGI and the Polish NGI have concrete plans of integration of UNICORE resources.

As to Globus, a survey will be conducted in collaboration with the IGE project to understand which Globus resource providers are willing to become part of existing National Grid Infrastructures, and the related timeline for their integration. This will provide the necessary input to develop a joint integration plan.

#### 4.1.2 Integration of desktop Grids

Meetings have been organized with representatives of the EDGI Project [R 16] during the EGI Technical Forum to develop a joint integration strategy. The EDGI Project will contribute to the software development of the extensions needed to ensure a seamless monitoring and accounting infrastructure. Exchange of technical information between the two projects has already started in October 2010, and it is expected that this will continue during the next months.

EDGI will not be responsible of operating an independent pan-European desktop grid infrastructure. On the contrary, it is expected that individual desktop grids will be operated under the umbrella of the EGI NGIs. For this reason, surveys will be periodically conducted to gather information about NGI plans, and to define use cases.

#### 4.1.3 Integration of cloud services

Collaboration started with the StratusLab project [R 4] during the first EGI Technical Forum. It is expected that two fully virtualized grid sites will be integrated with EGI as part of the Greek NGI. As virtualized grid sites will rely on UMD middleware components, full monitoring and accounting functionality will be granted for such sites.

However, the deployment of virtualized resources requires extensions to the current monitoring and accounting functionality. Various use cases have been identified and discussed during the “EGI Production Infrastructure” session of the first EGI Technical Forum. These need to be further refined to define a common tool development roadmap.

#### **4.1.4 Integration of new resources into accounting**

The integration of resources such as storage, MPI clusters, virtualized computing clusters, etc., will likely require extensions to the existing accounting usage record schema, to the central and regionalized repositories and portals, and possibly to the communication infrastructure used to exchange usage records.

An initial set of requirements has been gathered through the first middleware survey that was conducted during October 2010, whose output will be shared with the EGI software providers and will be reflected in the next version of the UMD Roadmap. Several NGIs contributed requirements – among these Italy and Spain. NGIs interested in prototyping extensions of the current accounting infrastructure will be involved in the definition of a set of use cases and of the related time scales.

- Integration of storage resources into accounting: Italy and possibly other NGIs that are pioneers in this field.
- MPI accounting: Italy is certainly interested in this, together with Spain. Other NGIs from SEE region such as Turkey and Bulgaria have expressed expertise and requirements as well.

This will lead to new requirements on the accounting interface which are not directly coupled to the requirements relevant for integrating resources from new middleware stacks or from other infrastructures as discussed previously. Requirements on accounting we expect to possibly arise:

- Accounting of MPI jobs as well as accounting of virtual resources (grid-cloud integration) should be possible.
- Regional versions of the accounting portal might turn out to be necessary.
- Usage Records (URs) should comply to a common standard usage record if possible.

A common transport mechanism needs to be identified to transport records across sites deploying different middleware stacks.

## ***4.2 Requirements coming from Collaborations with other distributed infrastructures***

Collaboration with DEISA and PRACE started with a dedicated meeting which was organized in September 2010.

Currently, the percentage of users that is interested in capacity as well as capability computing at the same time is rather low. However, regardless of this, infrastructure providers need to support users in directing them to the infrastructure that suits their use



case best, and need to reduce the number of barriers user may experience, so that shifting from one infrastructure to the other should be a more smooth and transparent process. This can happen through the deployment of common top-level tools, support systems and procedures.

Some milestones on the way to a more unified user experience (e.g. SSO authentication, trust in EUGridPMA, the usage of the GLUE standard for hardware descriptions, etc.) have already been achieved.

Conscious differences are currently experienced in authorization, resource allocation (project-oriented vs. VO-oriented) as well as in responsibilities and ways of user administrations (e.g. site-administrated LDAP vs. VO-administrated VOMS).

Several topics of common interest were identified:

1. Support: deployment of an integrated helpdesk system constituted by different distributed infrastructure helpdesks together with an automatic routing mechanism for trouble tickets addressed to the respective infrastructure provider. Such a common support network is offering a single entry point to get support for the users and their communities.
2. Accounting: deployment of an integrated central repository and portal providing access to accounting information from different DCIs. These tools can offer a comprehensive picture of use for large international collaboration making use of both HTC and HPC resources.
3. Resource allocation mechanisms allowing a more dynamic allocation of a resource budget to users according to their yearly grant, where applicable.
4. Operational Level Agreements: these define a baseline set of procedures and policies and the operational services shared between different infrastructure providers, availability and reliability of services offered, and other related quality parameters. Sharing of agreements, the respective templates and using a common terminology can facilitate DCI integration.



## 5 REFERENCES

R 1	MS405: Operational Security procedures <a href="https://documents.egi.eu/secure/ShowDocument?docid=47">https://documents.egi.eu/secure/ShowDocument?docid=47</a>
R 2	JRA1 Description of Work summary <a href="https://wiki.egi.eu/wiki/WP7:_Operational_Tools_DoW_summary">https://wiki.egi.eu/wiki/WP7:_Operational_Tools_DoW_summary</a>
R 3	Integration of EMI support units into GGUS <a href="https://twiki.cern.ch/twiki/bin/view/EMI/MilestoneMSA11">https://twiki.cern.ch/twiki/bin/view/EMI/MilestoneMSA11</a>
R 4	StratusLab <a href="http://stratuslab.eu">http://stratuslab.eu</a>
R 5	JSPG <a href="http://www.jspg.org/">http://www.jspg.org/</a>
R 6	GOCDB general documentation index: <a href="https://wiki.egi.eu/wiki/GOCDB_Documentation_Index">https://wiki.egi.eu/wiki/GOCDB_Documentation_Index</a>
R 7	dCache <a href="http://www.dcache.org/">http://www.dcache.org/</a>
R 8	LFC catalogue service <a href="http://goc.grid.sinica.edu.tw/gocwiki/How_to_set_up_an_LFC_service">http://goc.grid.sinica.edu.tw/gocwiki/How_to_set_up_an_LFC_service</a>
R 9	VOMS <a href="https://twiki.cnaif.infn.it/twiki/bin/view/VOMS/WebHome">https://twiki.cnaif.infn.it/twiki/bin/view/VOMS/WebHome</a>
R 10	MS408: EGI Operational Procedures <a href="https://documents.egi.eu/secure/ShowDocument?docid=209">https://documents.egi.eu/secure/ShowDocument?docid=209</a>
R 11	EGI Trust Anchor distribution <a href="https://wiki.egi.eu/wiki/EGI_IGTF_Release_Process">https://wiki.egi.eu/wiki/EGI_IGTF_Release_Process</a>
R 12	M.Ellert et al., Future Generation Computer Systems 23 (2007) 219-240.
R 13	Field L and Schultz M W Proc. of CHEP 2004, CERN-2005-002, 2005
R 14	GLUE Schema specifications <a href="http://www.ogf.org/documents/GFD.147.pdf">http://www.ogf.org/documents/GFD.147.pdf</a>
R 15	gLite WMS <a href="http://glite.web.cern.ch/glite/packages/R3.1/deployment/glite-WMS/glite-WMS.asp">http://glite.web.cern.ch/glite/packages/R3.1/deployment/glite-WMS/glite-WMS.asp</a>
R 16	EDGI Project <a href="http://edgi-project.eu/">http://edgi-project.eu/</a>
R 17	Software Provider SLA Agreement <a href="https://documents.egi.eu/secure/ShowDocument?docid=212">https://documents.egi.eu/secure/ShowDocument?docid=212</a>
R 18	UNICORE bug tracker <a href="http://sourceforge.net/tracker/?group_id=102081&amp;atid=633902">http://sourceforge.net/tracker/?group_id=102081&amp;atid=633902</a> UNICORE feature tracker <a href="http://sourceforge.net/tracker/?group_id=102081&amp;atid=633905">http://sourceforge.net/tracker/?group_id=102081&amp;atid=633905</a>
R 19	SGAS <a href="http://www.sgas.se">http://www.sgas.se</a>
R 20	SGAS to APEL Byrom R et al. <a href="http://www.gridpp.ac.uk/abstracts/allhands2005/apel.pdf">http://www.gridpp.ac.uk/abstracts/allhands2005/apel.pdf</a>
R 21	EMI software maintenance and support plan <a href="https://twiki.cern.ch/twiki/bin/view/EMI/DeliverableDSA11">https://twiki.cern.ch/twiki/bin/view/EMI/DeliverableDSA11</a>

<b>R 22</b>	L Field et al., 2010 J. Phys.: Conf. Ser. 219 062051
<b>R 23</b>	Operations Portal New Home Page <a href="https://operations-portal.in2p3.fr">https://operations-portal.in2p3.fr</a>
<b>R 24</b>	Lavoisier Home page <a href="http://grid.in2p3.fr/lavoisier">http://grid.in2p3.fr/lavoisier</a>
<b>R 25</b>	SAGA Service Discovery API <a href="http://www.ggf.org/documents/GFD.144.pdf">http://www.ggf.org/documents/GFD.144.pdf</a>
<b>R 26</b>	Common Information Service (CIS) for UNICORE Grids <a href="http://www.unicore.eu/community/development/CIS/cis.php">http://www.unicore.eu/community/development/CIS/cis.php</a> <a href="http://www.d-grid.de/fileadmin/user_upload/documents/MonitoringWorkshop/Memon.pdf">http://www.d-grid.de/fileadmin/user_upload/documents/MonitoringWorkshop/Memon.pdf</a>
<b>R 27</b>	Common Information Model Home Page <a href="http://www.dmtf.org/standards/cim/">http://www.dmtf.org/standards/cim/</a>
<b>R 28</b>	UNICORE support mailing lists for EMI related and general issues: <a href="mailto:emi-support@unicore.eu">emi-support@unicore.eu</a> and <a href="mailto:unicore-support@lists.sourceforge.net">unicore-support@lists.sourceforge.net</a> .
<b>R 29</b>	DEISA user management <a href="http://www.deisa.eu/services/user-related#usermngt">http://www.deisa.eu/services/user-related#usermngt</a>
<b>R 30</b>	UNICORE 6 Monitoring with Nagios <a href="http://www.d-grid.de/fileadmin/user_upload/documents/MonitoringWorkshop/Rambadt.pdf">http://www.d-grid.de/fileadmin/user_upload/documents/MonitoringWorkshop/Rambadt.pdf</a>
<b>R 31</b>	PL-Grid UNICORE Monitoring System <a href="http://www.unicore.eu/summit/2010/presentations/18_Bala_Monitoring.pdf">http://www.unicore.eu/summit/2010/presentations/18_Bala_Monitoring.pdf</a>
<b>R 32</b>	UNICORE architecture <a href="http://www.unicore.eu/unicore/architecture.php">http://www.unicore.eu/unicore/architecture.php</a>
<b>R 33</b>	Relational Grid Monitoring Architecture <a href="http://www.r-gma.org/">http://www.r-gma.org/</a>
<b>R 34</b>	APEL Home <a href="http://goc.grid.sinica.edu.tw/gocwiki/ApelHome">http://goc.grid.sinica.edu.tw/gocwiki/ApelHome</a>
<b>R 35</b>	Extensions to OGF-UR V1.0 as used in APEL <a href="http://forge.ggf.org/sf/docman/do/listDocuments/projects.ur-wg/docman.root.current_drafts.aggregate_ur_schema">http://forge.ggf.org/sf/docman/do/listDocuments/projects.ur-wg/docman.root.current_drafts.aggregate_ur_schema</a>
<b>R 36</b>	GGUS Documentation on interfaces and templates for new support units <a href="https://gus.fzk.de/pages/ggus-docs/interfaces/docu_ggus_interfaces.php">https://gus.fzk.de/pages/ggus-docs/interfaces/docu_ggus_interfaces.php</a> <a href="https://gus.fzk.de/pages/ggus-docs/PDF/1800_FAQ_for_TEMPLATE.pdf">https://gus.fzk.de/pages/ggus-docs/PDF/1800_FAQ_for_TEMPLATE.pdf</a>
<b>R 37</b>	NGI Creation Process <a href="https://wiki.egi.eu/wiki/Operations:NewNGIs_creation">https://wiki.egi.eu/wiki/Operations:NewNGIs_creation</a>
<b>R 38</b>	Nagios <a href="http://www.nagios.org/documentation">http://www.nagios.org/documentation</a>
<b>R 39</b>	MyEGI Portal <a href="https://grid-monitoring.egi.eu/myegee/">https://grid-monitoring.egi.eu/myegee/</a>
<b>R 40</b>	Ops-monitor Nagios instance <a href="https://ops-monitor.cern.ch/nagios">https://ops-monitor.cern.ch/nagios</a>
<b>R 41</b>	Security Policy Group <a href="https://wiki.egi.eu/wiki/SPG">https://wiki.egi.eu/wiki/SPG</a> Software Vulnerability Group <a href="https://wiki.egi.eu/wiki/SVG">https://wiki.egi.eu/wiki/SVG</a>
<b>R 42</b>	Nagios Probe Documentation and Description <a href="https://twiki.cern.ch/twiki/bin/view/LCG/SAMProbesMetrics">https://twiki.cern.ch/twiki/bin/view/LCG/SAMProbesMetrics</a> In the future: <a href="https://wiki.egi.eu/wiki/Operations:Operations_tests">https://wiki.egi.eu/wiki/Operations:Operations_tests</a>

<b>R 43</b>	Accounting portal <a href="http://www3.egee.cesga.es/gridsite/accounting/CESGA/egee_view.php">http://www3.egee.cesga.es/gridsite/accounting/CESGA/egee_view.php</a>
<b>R 44</b>	WS J. Ainsworth, S. Newhouse, and J. MacLaren. Resource Usage Service (RUS) based on WS-I Basic Profile 1.0. UR, August 2005
<b>R 45</b>	Grid Policy on the Handling of User-Level Job Accounting Data <a href="https://edms.cern.ch/document/855382">https://edms.cern.ch/document/855382</a>
<b>R 46</b>	HEP-SPEC06 <a href="https://hepux.caspar.it/benchmarks/doku.php">https://hepux.caspar.it/benchmarks/doku.php</a> <a href="http://hepux.caspar.it/afs/hepux.org/project/ptrack/#SPEC_CPU2006">http://hepux.caspar.it/afs/hepux.org/project/ptrack/#SPEC_CPU2006</a>
<b>R 47</b>	Integrating ARC into SAM <a href="https://tomtools.cern.ch/jira/browse/SAM-751">https://tomtools.cern.ch/jira/browse/SAM-751</a>
<b>R 48</b>	Building Packages on the SA1 Koji build system <a href="https://twiki.cern.ch/twiki/bin/view/EGEE/EGEESA1BuildingPackages">https://twiki.cern.ch/twiki/bin/view/EGEE/EGEESA1BuildingPackages</a>
<b>R 49</b>	Additional steps required when supporting ARC services <a href="https://tomtools.cern.ch/confluence/display/SAM/SAM+setup+for+ARC+services">https://tomtools.cern.ch/confluence/display/SAM/SAM+setup+for+ARC+services</a>