EGI-InSPIRE

INTEGRATING RESOURCES INTO THE EGI PRODUCTION INFRASTRUCTURE

EU DELIVERABLE: MS407

Document identifier:	EGI-MS407-V0-95.odt
Date:	**21/10/2010**
Activity:	**SA1**
Lead Partner:	**KTH**
Document Status:	**DRAFT**
Dissemination Level:	**PUBLIC**
Document Link:	https://documents.egi.eu/document/111

Abstract

This document describes and defines the operational interfaces that must be supported for resources to be integrated into the EGI production infrastructure. This includes operational tools provided by activity EGI-JRA1 and procedures and policies defined together by O6, OE-13 and OE-11.

Delivery Slip

	Name	Partner/Activity	Date
From	Michaela Lechner	KTH/SA1	
Reviewed by	**Moderator:** **Reviewers:**		
Approved by	**AMB & PMB**		

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
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		First complete draft	

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 the 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 the ESFRI projects. 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 − structured international user communities − that are grouped into specific research domains. VRCs are formally represented within EGI at both a technical and strategic level.

TABLE OF CONTENTS

Index of Tables

Index of Figures

1. INTRODUCTION

1.1. PURPOSE

In order to add new resources into the EGI production infrastructure a basic set of operational interfaces that must be supported by the newcomers 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 EGEE project series experiences. However this might change in the future. Still we take this as a starting point when comparing the interoperability of different middleware components for each operational tool in our current horizon.

Operational procedures and policies are needed as well 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 may be adapted to the EGI era, while new requirements will have to be identified and turned into new procedures and policies. Special focus shall be laid on security.

1.2. 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.

1.3. REFERENCES

Table 1: Table of references

R 1	MS405: Operational Security procedures https://documents.egi.eu/secure/ShowDocument?docid=47
R 2	EGI Wiki https://wiki.egi.eu/
R 3	EGI SSO: https://www.egi.eu/sso/
R 4	EGI Mail manager https://mailman.egi.eu/mailman/listinfo
R 5	GOCDB requests and wish list https://savannah.cern.ch/support/?group=gocdb
R 6	GOCDB general documentation index: https://wiki.egi.eu/wiki/GOCDB_Documentation_Index

R 7	dCache http://www.dcache.org/
R 8	LFC catalogue service http://goc.grid.sinica.edu.tw/gocwiki/How_to_set_up_an_LFC_service
R 9	VOMS http://hep-project-grid-scg.web.cern.ch/hep-project-grid-scg/voms.html
R 10	WLCG http://lcg.web.cern.ch/lcg/public/
R 11	Globus Meta Data Service, Globus MDS
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 http://infnforge.cnaf.infn.it/glueinfomodel/ Glue Schema specifications http://www.ogf.org/documents/GFD.147.pdf
R 15	gLite WMS http://glite.web.cern.ch/glite/packages/R3.0/deployment/glite-WMS/glite-WMS.asp
R 16	EGEE Accounting Portal http://www3.egee.cesga.es/
R 17	Real Time Monitor http://gridportal.hep.ph.ic.ac.uk/rtm/
R 18	UNICORE bug tracker http://sourceforge.net/tracker/?group_id=102081&atid=633902 UNICORE feature tracker http://sourceforge.net/tracker/?group_id=102081&atid=633905
R 19	SGAS http://www.sgas.se
R 20	SGAS to APEL Byrom R et al. http://www.gridpp.ac.uk/abstracts/allhands2005/apel.pdf
R 21	Grønager M et al escience, pp.493-500, 2008 Fourth IEEE International Conference on eScience, 2008
R 22	Towards Sustainability: An Interoperability Outline for a Regional ARC based infrastructure in the WLCG and EGEE infrastructures
R 23	Operations Portal New Home Page https://operations-portal.in2p3.fr
R 24	Lavoisier Home page http://grid.in2p3.fr/lavoisier
R 25	SAGA Service Discovery API http://www.ggf.org/documents/GFD.144.pdf
R 26	Common Information Service (CIS) for UNICORE Grids http://www.unicore.eu/community/development/CIS/cis.php http://www.d-grid.de/fileadmin/user_upload/documents/MonitoringWorkshop/Memon.pdf
R 27	Common Information Model Home Page http://www.dmtf.org/standards/cim/
R 28	UNICORE support mailing lists for EMI related and general issues:

	emi-support@unicore.eu and unicore-support@lists.sourceforge.net.
R 29	Google maps CIS web client demo http://omiiei.zam.kfa-juelich.de:6001/web/Index
R 30	UNICORE 6 Monitoring with Nagios http://www.d-grid.de/fileadmin/user_upload/documents/MonitoringWorkshop/Rambadt.pdf
R 31	PL-Grid UNICORE Monitoring System http://www.unicore.eu/summit/2010/presentations/18_Bala_Monitoring.pdf
R 32	UNICORE architecture http://www.unicore.eu/unicore/architecture.php
R 33	Relational Grid Monitoring Architecture http://www.r-gma.org/
R 34	APEL Home http://goc.grid.sinica.edu.tw/gocwiki/ApelHome
R 35	http://forge.ggf.org/sf/docman/do/listDocuments/projects.ur-wg/docman.root.current_drafts.aggregate_ur_schema
R 36	GGUS Documentation https://gus.fzk.de/pages/docu.php
R 37	NGI Creation Process https://wiki.egi.eu/wiki/Operations:NewNGIs_creation
R 38	Nagios http://www.nagios.org/documentation
R 39	MyEGI Portal https://grid-monitoring.egi.eu/myegee/
R 40	Ops-monitor Nagios instance https://ops-monitor.cern.ch/nagios
R 41	https://rt.egi.eu/rt/Ticket/Display.html?id=79
R 42	Nagios Probe Documentation and Description https://twiki.cern.ch/twiki/bin/view/LCG/SAMProbesMetrics
R 43	Accounting portal http://www3.egee.cesga.es/gridsite/accounting/CESGA/egee_view.php
R 44	WS J. Ainsworth, S. Newhouse, and J. MacLaren. Resource Usage Service (RUS) based on WS-I Basic Profile 1.0. UR, August 2005
R 45	Grid Policy on the Handling of User-Level Job Accounting Data https://edms.cern.ch/document/855382
R 46	HEP-SPEC06 https://hepix.caspur.it/benchmarks/doku.php http://hepix.caspur.it/afs/hepix.org/project/ptrack/#SPEC_CPU2006
R 47	EGI Trust Anchor distribution https://wiki.egi.eu/wiki/EGI_IGTF_Release_Process
R 48	Integration of EMI support units into GGUS https://twiki.cern.ch/twiki/bin/view/EMI/MilestoneMSA11 EMI software maintenance and support plan https://twiki.cern.ch/twiki/bin/view/EMI/DeliverableDSA11

1.4. 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

1.5. TERMINOLOGY

A complete project glossary is provided in the EGI-InSPIRE glossary:

http://www.egi.eu/results/glossary/.

The table below contains further terminology not provided in the previous location:

ARC	Advanced Resource Connector (middleware stack)
BCP	Best Common Practices
BDII	Berkeley Database Information Index
CIS	Common Information Service
GIIS	Grid Index Information Server
gLite	Lightweight Middleware for Grid Computing (middleware stack)
Globus	Globus Toolkit Grid Middleware (middleware stack)
GRIS	Grid Resource Information Service
GOCDB	Grid Operations Centre DataBase
MPI	Message Passing Interface
OSG	Open Science Grid
R-GMA	Relational Grid Monitoring Architecture
RUS	Resource Usage Service
UNICORE	Uniform Interface for Computing Resources (middleware stack)
XUUDB	UNICORE User Database

Table 2: Glossary of terms.

2. EXECUTIVE SUMMARY

This document describes and defines the operational interfaces that must be supported for resources to be integrated into the EGI production infrastructure.

For the each of the operational tools we describe the steps necessary to integrate a new middleware stack into the production infrastructure, followed by a detailed analysis for each middleware stack in the UMD and IGE and the immediate future plans for operational interoperability.

An overview table shows the general picture outlining the current status of each MW in relation to the currently existing operational tools in our scope. The actual operational tools might change though in the future.

Based on this table we gather a requirement list of our suggestions to each MW provider, so that sites running only this specific MW stack will still be able to make full use of all relevant operational tools features to be fully integrated in the EGI infrastructure. Requirements can also stem from a more general interoperability point of view.

~~For completeness we also include a section describing the level of interoperation level between the different MW stacks, since this will get more important in the future when sites want to support several operational/middleware stacks in parallel, or for the interoperability of sites running different MW stacks.~~

An initially thought section describing the interoperation level between different MW stacks was skipped in the editing process for this version of the milestone. Such a section will get more important in the future when sites want to support several operational/middleware stacks in parallel, or for the interoperability of sites running different MW stacks. Issues around the interoperability of the middleware itself from a user point of view will always remain out of scope for the context of this milestone, though.

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.

3. 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 project series. A certain amount of rationalisation 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 relies on operational tools already developed in the framework of the EGEE project series. Tool development is an ongoing effort 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.

The EGI infrastructure will need to deploy several middleware stacks. 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 to be implemented. Comprehensive integration is a short-term objective 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 the EGI infrastructure will be integrated with new types of resource, such as virtualisation, digital libraries and repositories, desktop grids, High Performance Computing, etc.

3.1. OVERVIEW INTEROPERATION STATUS FOR OPERATIONAL TOOLS - MW

	gLite	ARC	UNICORE	Globus
GOCDB	completed	completed	to be done	to be done
Nagios - Definition of critical tests	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
Accounting	completed	completed	not (yet) available	not (yet) available
3rd level support in GGUS (access to expert teams via DMSU and ev. EMI)	completed	completed	completed by 29/09/2010	to be done

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

3.2. INTEROPERATION AT AN INFRASTRUCTURE LEVEL

The basic operational interfaces that must be supported for resources to be integrated into EGI's production infrastructure 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.

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 the current EGI production infrastructure 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.

The next step is to monitor the resources in some way using the OGF GLUE2 standard schema [R 14] enabling a unified view of Grids and resources per infrastructure, computing centre or federation. One possible monitoring tool fulfilling our requirements is for example Nagios, where critical services and probes have to be defined for these new services. Such a test execution and notification environment is needed for the fast identification and consequently fast resolution of eventually arising problems. The collected information can then be plugged in in the Operational Portal to give a detailed overview of operational status and the possibility to contact the sites as stored in GOCDB. General monitoring is also needed to produce Availability & Reliability figures.

Besides that a responsible distributed computing infrastructure like EGI has to insure the quality of service by providing 3rd level support and by being able to account for these resources to provide planning and usage information.

EMI has set up a 3rd level structure within GGUS for its various middleware stacks and services since GGUS has been adopted as a common infrastructure to exchange trouble tickets between different stakeholders due to its adoption during the EGEE era.

Accounting is important as well since the key feature of an operational infrastructure is that the resources have high availability and reliability, and that we can measure their usage. The summary data of the amount of actually delivered computing resources is relevant for VOs and project communities as well as on site level to check if all agreements have been fulfilled.

3.3. DEFINITION AND DESCRIPTION OF A MANAGEMENT INTERFACE

3.3.1. Functionality

A management interface is an operational interface which allows sites to store, maintain and view the topology of the whole EGI infrastructure and the basic information of its resources.

Such an EGI management interface contains information and a placement in the topology order on:

- Participating National Grid Initiatives (NGI) and possible other groups (Countries, ROCs) 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 regionalised or otherwise separated instances with their own local inputs.

3.3.2. Requirements

The EGI management interface has to support the functionality described above. System and security contacts and higher level organisational 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 kind of service types, groups or sites within the management interface.

We expect to have a role based interaction model to such a database, 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 number of production states. Decision on that has to be taken with hindsight to the fulfilment of practical use cases.

A management interface shall enable us to follow 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.

Furthermore, we expect a plug-in to an approved operational portal interface to be in existence or easily implementable due to using canonical standards.

Even though the information is mostly of static kind, 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.

3.3.3. 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.

3.3.3.1. Integration of new MW service types

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

In the first case, the proposed procedure is as follows:

EGI-JRA1 gets from middleware providers (e.g. EMI) the information about new services that have been added to an existing middleware stack.

In the second case, the request of adding a set of services belonging to a previously undeclared MW stack implies that strategic decision has been made about the validity of the request. This is to ensure that only officially supported MW stacks are actually integrated to GOCDB.

Request for adding the new service types in GOCDB should be made through the official request submission channel [R5].

A validation board described in 3.3.3.4 discusses the request and gives its green light for the integration. New service types are then added to GOCDB and are made available to declare new resources as described in 3.3.3.3.

3.3.3.2. Integration of new non-MW resources types

There is a need to store non-middleware service types in GOCDB since services used for Grid operations are declared within the repository. Also, there might be a need to store and present information about application services, deployed by Grid sites to support certain VOs without belonging to a specific middleware distribution.

New non-middleware services are integrated into GOCDB in a similar way to MW services, apart from the fact that the initial information doesn't come from the software provider but from either EGI-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 3.3.3.1.

3.3.3.3. 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].

3.3.3.4. 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 avoid filling up GOCDB with unused services, 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 get information from the Software Providers (e.g. EMI, IGE, EGI-JRA1, etc.) before producing a list of service types that are candidates for decommissioning.

3.3.3.5. 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 [R 6].

- If the service type is not available, a request to GOCDB developers has to be made in [R 5]. The case will then be discussed as described above and eventually result in the new service type being added.

3.3.3.6. 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 MW 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 EGEE 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-BDIIs 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 EGEE central/backbone messaging system.
- **Site-NAGIOS**: [Site service] site-level Nagios monitoring box
- **National-NAGIOS**: [Regional Service] NGI-level Nagios monitoring box
- **Regional-NAGIOS**: [Regional Service] ROC-level Nagios monitoring box
- **Project-NAGIOS**: [Central Service] project-level Nagios monitoring box
- **gLite-APEL**: [Site service] The gLite-APEL hosts the site Accounting client (3.2 replacement of the MonBox)

3.3.3.7. Integrating ARC resources in GOCDB

ARC resources were already added into GOCDB as early as 2007. This has happened even though the Nordic infrastructure using the ARC middleware was not formally an EGEE partner. ARC integration could 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 storage elements (dCache), catalogue service (LFC), VOMS, 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. A virtual site was created for NDGF in GOCDB so that the ARC-CEs could be registered there.

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. In order for these resources to be visible for EGEE services a special BDII has been set up for the virtual NDGF 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.

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

3.3.3.8. Integrating UNICORE resources in GOCDB

The needed MW service types haven't 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 need to be.

- Gateway (Sits in front of one or more UNICORE services as a gateway to the internet. Normally one Gateway per site.)
- Registry (All UNICORE services register here; clients ask the registry for available services in the Grid. Normally there's one Registry per Grid infrastructure. Backups can occur. The Registry works like a phone book and collects URLs of services.)
- Workflow Engine (Needed to add workflow functionality to UNICORE. 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. One Service Orchestrator per Grid infrastructure)
- UNICORE/X (Hosts the XNJS, which handles job submission, file transfer, job monitoring etc., and the CIP. One UNICORE/X per supercomputer/cluster.)

- CIS (Information service. Standalone service which collects information from the UNICORE/X. One per grid.)
- XUUDB (User database. Maps certificates or DNs to user logins, roles etc. Services like the Workflow Engine and the UNICORE/X query the XUUDB for authorisation. Pretty flexible how many there are per Grid; each site running their own XUUDB seems to be the most common setting.)
- UVOS - Serves the same function as XUUDB but is much more advanced and flexible by supporting arbitrary attributes, groups, advanced authorization, and more. Usually one per grid, 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].

3.3.3.9. Integrating Globus resources in GOCDB

The three most important service types for Globus which need to be 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.

3.4. DEFINITION AND DESCRIPTION OF A MONITORING INTERFACE

3.4.1. Functionality

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

Critical tests to monitor all mission-critical infrastructure components have to be defined and implemented as probes. 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. 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. Users and operators are informed about the state of the Grid.

The design of the monitoring interface is scalable and a fail-over concept is in existence.

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

3.4.2. 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 avoids scalability issues 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. Otherwise problems at a central location would reflect on the whole grid, and any changes would require consultations with a central body under the control of a single team. If something goes wrong in a regionalized scenario it can be quickly solved locally without asking a central party to perform actions. 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 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.
- The generic probe profile, which also works as a basis for availability calculation, has to be checked at regular intervals to ensure it is up-to-date. In particular, if the current set of probes fulfils all the needs or has to be extended or reorganized. New probes shall be identified and provided as required. This should happen in coordination with the software providers.
- Information shall be exchanged according to a given template or prevalent open standard, e.g. the use of ActiveMQ (or an alternative) transport protocol is recommended.
- It shall work as an input plugin for the Operational Portal.
- Additionally it would be desirable to add an additional level and to not only monitor the resources put also the availability of needed operational tools, like the different regional monitoring instances.

We continue with referring to the Service Availability Monitor (SAM) in the regionalized Nagios monitoring framework based version where each region runs its own instance. This Nagios monitoring framework based solution was redesigned and chosen in favour of the former centralised SAM submission framework by the WLCG Grid Service Monitoring

working group which was deployed at CERN during the EGEE project series to monitor the infrastructure's resources before being decommissioned on June 23th 2010.

3.4.3. Interoperability of different MW stacks with Nagios

Nagios [R 38] is a well-known and mature 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 extendible 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 critical tests to monitor further mission-critical infrastructure components can be defined and deployed with freshly written probes for them.

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

The current design was finished within the Operations Automation Team (OAT) group.

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. Further details can be found in the R 41.

Analysis of fail-over configuration of centralized tools was performed. SAM/Nagios instances are supposed to be deployed at each NGI. Each NGI is responsible for fault tolerance implementations. Certain procedures ensuring that a NGI's Nagios is not down for a longer period of time are still needed.

To integrate a new MW stack into Nagios, critical tests for the service types defined in the management interface for this MW have to be defined and then Nagios probes for them have to be written. Possibly it is also sufficient to just have a compatible Nagios reporter from a different kind of monitoring tool which can be integrated in regional and central instances.

3.4.3.1. Critical tests and Nagios probes for gLite resources

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

- BDII (top and site BDII)
- CE
- CREAM-CE
- FTS_oracle
- LB
- LFC_mysql/oracle
- MON

- PX
- SE_dcache
- SE_dpm_disk
- SE_dpm_mysql
- VOBOX
- VOMS

Regarding these probes, further documentation and descriptions are found on the dedicated twiki page[R 42].

3.4.3.2. Critical 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 3rd 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 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 is ensured.

The transition towards Nagios monitoring was done during EGEE III together with gLite.

3.4.3.3. Critical tests and Nagios probes for UNICORE resources

UNICORE does have Nagios reporters which make use of the UNICORE monitoring tool SIMON. The Site Monitor for UNICORE resources (SIMON) [R 30] submits various kinds of UNICORE test jobs to check the availability of the UNICORE stack. One could integrate those into the EGI Nagios. SIMON acts as a user, thus needs its own certificate, login and entry in the UNICORE User Database. PL-Grid defined a number of critical tests and their dependencies [R 31].

UNICOREs 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. A small demo of a Google maps CIS web client can be found under [R 29].

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 critical.

3.4.3.4. Critical tests and Nagios probes for Globus resources

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

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

Currently the following Nagios probes for critical tests 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 if these Nagios probes can be used as is or if they need to be adjusted to the EGI requirements.

3.4.4. Procedure to integrate new Nagios probes

Xxxxx (see Emir)

3.5. DEFINITION AND DESCRIPTION OF AN ACCOUNTING INTERFACE

3.5.1. Functionality

The EGI Accounting Infrastructure collects CPU accounting records from sites and/or grid infrastructures and summarises 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 partner Grids.

Accounting is necessary to demonstrate that delivered computing resources to a specific project are in accordance with expectations, e.g. within signed Memorandum of Understanding agreements. 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 utilised by their users.

When looking at the accounting interface as the interface between the accounting services of different interoperating infrastructures ~~T~~he main aim ~~of interoperation~~ 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.

3.5.2. Requirements

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

- Access to accounting data needs to respect all relevant policy and legal requirements. It is expected that this is controlled by the standard user authentication and authorisation 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.
- ~~Many national states do not allow for accounting info on the person level to be exported outside country borders. Hence a federated infrastructure, only accounting information suitably aggregated and anonymised will be submitted to the central database. Regional versions of the accounting portals are therefore 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.~~
- ~~Accounting of MPI jobs as well as accounting of virtual resources (grid-cloud integration) should be possible.~~

3.5.3. Current Status

The EGI Accounting Infrastructure is based on APEL [R 34]. The collected CPU accounting records and the data summarised 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 partner Grids.

The bulk of existing sites collect data from their batch systems (list those supported), 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 transport layer from R-GMA [R 33] to ActiveMQ already used by other EGI Operational Tools. Other partner Grids (list), and a few sites, with their own accounting services, publish summaries of data in the form described above to 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 are 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 since 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 adoption [R 35].

In addition to the ActiveMQ route for receiving data, the APEL development roadmap plans to have a RUS [R 44] interface to receive data only.

3.5.4. 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 transform.

c) Register the publisher with APEL (host DN). 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.

d) Publish the records into EGI's ActiveMQ Message Bus with a destination of (queue/topic tbc). 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.

e) From the master summary table, the data are then exported to CESGA where they can be viewed in the accounting portal.

3.5.4.1. 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 recognised 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.
- 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 ~~Policy for Storage of Accounting Data~~ Grid Policy on the Handling of User-Level Job Accounting Data ~~ck the name and ref)(che~~[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 reevaluated again when exchanging accounting data with other infrastructures like e.g. DEISA.
- Current accounting is only of CPU~~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 normalisation of CPU~~cpu~~ time in EGEE, EGI, and WLCG is HEPSPEC06 hours ~~(ref)~~[R 46]. For ~~I~~interoperation with an infrastructure that does not collect this value from the resources running jobs, some conversion factor must be negotiated.

3.5.4.2. 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 infrastructure to use a standard web services interface to publish records. This will replace item (d) in the integration list above.

3.5.4.3. 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) and an automatic script for exporting the accounting info gathered in SGAS to APEL was set up. Currently only LHC VOs are published to APEL but this could easily be extended to other international VOs.

3.5.4.4. 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 formated and provided according to XUUDB access control. (should mention more about RUS here as well?)

3.5.4.5. Globus resources

OGF-UR is available and used in DEISA (and soon also in PRACE). However, currently it is not integrated in the Globus tools. UR should be able to send UR to APEL, though. 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).

If DGAS is used then publishing to APEL should already be possible.

3.6. DEFINITION AND DESCRIPTION OF A SUPPORT INTERFACE

3.6.1. Functionality

The user support infrastructure in use within EGI is a distributed one consisting of various topical and regional helpdesk systems that are linked together through a central integration platform, the GGUS helpdesk. This central helpdesk enables formalised communication between all partners involved in user support by providing an interface to which all other tools can connect and thus 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 the ticket systems in place throughout the project enables a passing of 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 exits to ensure the quality of service.

These are documented in the GGUS documentation [R 36].

For EGEE, and now EGI, an own functional institution has been introduced 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 that part of the tickets that have to be assigned manually, so that they get forwarded to the right persons and the right units. The TPM teams act as a 1st line support chain and have also to keep track of long-term trouble tickets and help to solve

them with their very good general grid knowledge. In this way, a problem submitted to GGUS can be quickly identified as either a grid problem or a VO specific problem and addressed to the appropriate second line specialized support units or the dedicated VO support teams whose members have specific VO knowledge.

The second line support is formed by many support units. Each support unit is formed from members who are specialists in various areas of grid middleware, or ROC 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.

3.6.2. 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 initial problem. This transparency together with the independence from the actual ticket system used by the experts from the different areas who get assigned to the ticket can be seen as the main requirements that ensure that information flows between different parts of the EGI support network.

This is especially important since the support interface is not only used for 3rd level support dedicated to the end user, but also for 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.

3.6.3. Integration of new resources into GGUS

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

3.6.3.1. 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.

3.6.3.2. 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 small upstarting NGIs is to use GGUS directly. This has the limitation of just one support unit for the whole NGI. Tickets cannot be assigned to specialised 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. Customisation can be done vie 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].

3.6.3.3. Integration of a new technology provider into the support infrastructure

Should EGI decide to utilise 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 who 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. UNICORE. For details refer to the EMI Milestone 17 on the integration of EMI support units into GGUS or the EMI software maintenance and support plan [R 48]. E.g. in the case of 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].

3rd level support for Globus will be provided by IGE. IGE provides a support infrastructure for the European Globus users in all European, national, and regional e-Infrastructures with EGI and DEISA/PRACE being the most important ones. GGUS will contain a queue to forward 3rd level support tickets directly to the IGE user support team. Further details will be clarified shortly after the project start of IGE.

3.7. DEFINITION AND DESCRIPTION OF A DASHBOARD INTERFACE

3.7.1. 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 1st 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 (1st line support, regional, central). Sites in the dashboard scope can be regional, central or predefined out of a list and can be sorted and displayed 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.

3.7.2. Requirements

A dashboard interface has to fulfil the functionality described above. Further requirements are:

- access to a harmonized information service for...,
- access to a harmonized user authentication service for ... etc.

3.7.3. Operational Dashboard 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 EGEE, now 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 Lavosier are to provide:

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

This solution design means that the web application doesn't 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 we are able to retrieve information 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).

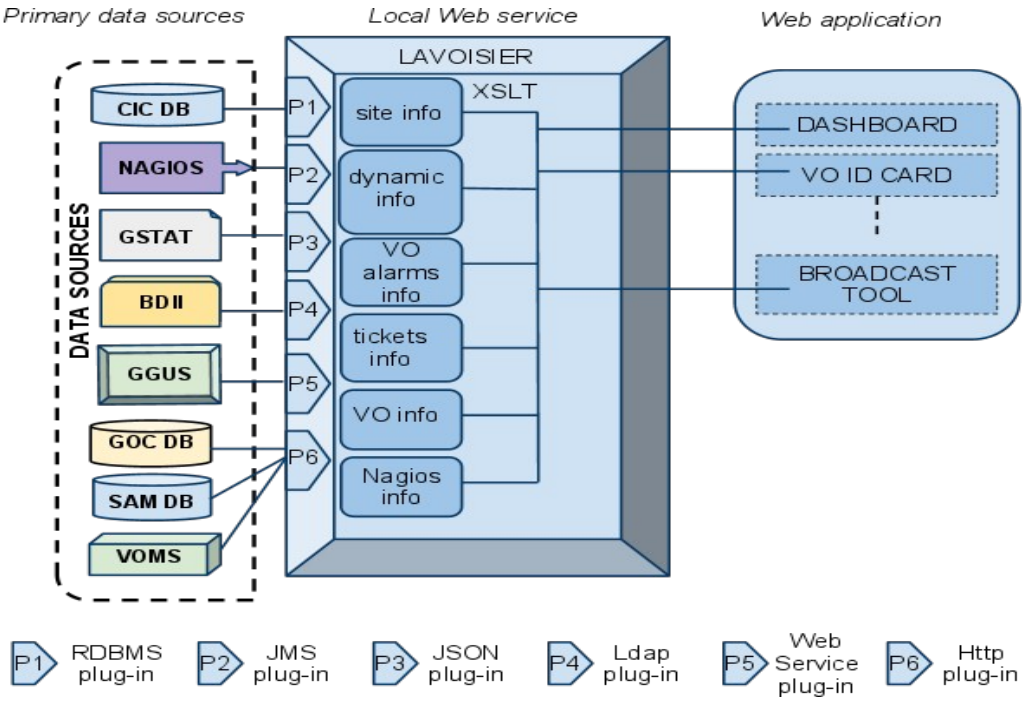

Fig. 1: Global architecture of the Operations Portal.

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

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 will 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 coming into the EGI production infrastructure, such as HPC systems, virtualized resources or desktop resources. As long as these resources are monitored we are able 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.

3.7.3.2. gLite resources in the Operational Dashboard

gLite resources are Nagios monitored and therefore already integrated. (true?)

3.7.3.3. ARC resources in the Operational Dashboard

ARC resources are Nagios monitored and therefore already integrated. (true?)

3.7.3.4. UNICORE resources in the Operational Dashboard

Xxxxx (depends on other things not finished yet, (list of services needed and integrated).

(Indirect over SIMON Nagios or directly from CIS over SAGA Link?)

3.7.3.5. Globus resources in the Operational Dashboard

With the Nagios probes available the operational alarms from Globus resources or central servers can be directly integrated in the operational dashboard.

3.8. USER MANAGEMENT, AUTHENTICATION AND AUTHORISATION

The actual way on how users are administrated and authenticated affect many operational interfaces as 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 a sites 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 organisational Certificate Authority (CA) which is recognised by the International Grid Trust Federation (IGTF) (see also [R 47]). Resources within the production infrastructure are made available to controlled collaborations of users represented in the infrastructure through Virtual Organisations (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 authorisation 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 to allow 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 an 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 authorisation information in a unified way in a large Grid infrastructure. In D-Grid i.e. 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 these informations a service is preparing mapping files for Globus, gLite, dCache, and UNICORE per site which then are used to feed e.g. the UNICORE User Database XUUDB.

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 visualised 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.

3.8.1. User management in gLite and ARC

VOMS is used for VO and user administration.

3.8.2. User management in UNICORE

The UNICORE User Database (XUUDB) stores the mapping of user certificates/DN's to local userIDs and roles at a single UNICORE site. The XUUDB is a site local authorization component, maintained by each site. These XUUDBs have to be filled with the information of those users who are authorized to use the site's resource(s). Proxy certificates are not used in UNICORE. Technically, it doesn't matter who manages the XUUDB user database. Every site can set up their own XUUDB and an independent way of managing it, or there could be a central XUUDB, or a central service that generates input for each site's XUUDB like it is done within D-Grid.

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 informations are collected in LDAP servers at the different sites that get synchronized once per day. Each site generates the input for its XUUDB from its local LDAP server. DEISA user management is described in detail in http://www.deisa.eu/services/user-related#usermngt.

3.8.3. User management in Globus

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

3.9. ~~INTEROPERATION BETWEEN OPERATIONAL STACKS~~

~~Requirements on the interoperation between different operational stacks have to come from the sites and NGIs which run more than one MW. Issues around the interoperability of the middleware itself from a user point of view remain out of scope for the context of this milestone, though.~~

3.9.1. ~~Job Submission~~

~~Cross Grid job submission is not strictly needed for infrastructure interoperation. It does, however, promote a more seamless integration between different infrastructures and middleware stacks and might therefore be desirable.~~

~~(reference to OGSA-BES and that through it UNICORE can submit Jobs to e.g. ARC.)~~

3.9.1.1. ~~Direct submission from gLite-WMS to an ARC-CE~~

~~Unlike gLite based Computing Elements (CEs), ARC-CEs are accessed directly by the ARC client. No intermediate resource broker like the gLite-WMS is needed. Brokering is performed by the ARC client. Various schemes have been explored for submitting a job from a gLite-~~

based infrastructure to an ARC environment. Already in EGEE-II-SA3 [R] direct submission from gLite-WMS to ARC-CEs got implemented and is now part of the standard gLite-WMS.

3.9.1.2. Direct submission from ARC to a gLite-CREAM-CE

Version 0.8 of the ARC client also supports submission directly to the gLite-CREAM-CE hence making cross grid submission possible also in the ARC->gLite direction. A more detailed analysis on how ARC-CE compares to gLite flavoured CEs has been written in [R]..

3.9.2. Data Management and Storage Infrastructure

(dCache for ARC and gLite)

3.9.3. Logging

Collecting detailed logging information in a common accessible and exchangeable format is clearly not the most essential high priority task when integrating new resources. A detailed analysis of this might follow in a later milestone.

3.9.3.1. Real Time Monitoring of ARC resources

ARC-CE supports logging calls to the gLite LB server. This means that detailed job states can also be obtained by ARC-CE sites, enabling advanced real time monitoring of the production flow like in the Real Time Monitor [R.] and for debugging scenarios.

Instead of installing the logging clients on all the ARC-CEs a hook in ARC for directly exporting the detailed job states to the gLite Logging and Bookkeeping server was chosen. (reference to OGSA-BES and that through it UNICORE can submit Jobs to e.g. ARC.)

3.10. REQUIREMENT LISTS TO THE MIDDLEWARE PROVIDERS

Xxxxx (outgoing from points in 3.2, was thought as a kind of conclusion, we won't get any requirements from the NGIs yet.)
(possibly related: https://rt.egi.eu/rt/Ticket/Display.html?id=231)

3.10.1. gLite

Xxxxx

3.10.2. ARC

Xxxxx

3.10.3. UNICORE

Xxxxx

3.10.4. Globus

Xxxxx

4. INTEROPERATION AT PROCEDURES AND POLICY LEVEL

4.1. SCOPE

After describing the technical set up in the previous sections we will now focus on the operational set up allowing researchers to enter European collaborations.

When integrating new resources we have to make sure that they do not compromise the reputation we have for our production infrastructure. In order for seamless interoperation it is extremely important to have OLAs and a high degree of communication between the different project partners and operations teams. The importance of having procedures and best practices that are valid for all project partners can not 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 the actual MW stack used and unified and collected to a common core that can be further extended to more explicit versions for specific MW stacks and to a certain level also adapted by all NGIs. On a smaller scale this approach is already applied successfully in the security context where several infrastructure providers agreed on a common procedure document which will be kept in sync and EGI has its own add-ons for it. (AUP Reference!!, Dave Kelsey presentation on Interoperability in case of incident response at PRACE helsinky security meeting)

(check overlap with MS 405 and 408!)

4.2. REQUIREMENTS

There are some important general requirements on procedures, policies and related documentation we would like to see fulfilled:

- Core documents should be as general as possible and not refer to any specific instance of MW or operational tool used.
- They should be fully collected in one place with no external links in order to provide data loss in case of changing the document format .
    - In the case of EGI this means that documentation should only be on the wiki to avoid confusion regarding the freshness of a document. There can only be one current version, which is generally accessed and a work in progress version which is only available to the people working on the document.
- There should be a well defined valid procedure in integrating new procedures. All the relevant players should have the possibility to suggest new best practices and procedures or improvements to already existing ones. In the case of EGI, new procedural documents should at least be approved by the Operations Management Board (OMB) before release.
- Procedure manuals should have a release schedule of around 3 times a year to keep them updated and functional.
- Request on consistency: A procedures should always refer to where the rest of the most actual version of the procedures is found. Higher level links to come to the current collection have to be integrated.

4.3. CURRENT STATUS OF EGI PROCEDURES AND POLICIES

This will give a quick overview of the procedures, policies and best practices inherited and already improved, changes needed and the adoptions and ideas for obtaining high standards in all aspects of the infrastructure.

4.3.1. Procedures taken over from EGEE

The heritage of the EGEE projects, which were not so much oriented towards technical experiments but quality of service, provide a solid platform on which we stand and from which we can create an also in this aspect exceptional grid infrastructure.

4.3.1.1. Operational Procedures Manual

The Operational Procedures Manual (OPS Manual) defines the procedures and responsibilities of the various parties involved in the running of the EGI infrastructure, namely the resource centres consisting of local support and sites administrators, the staff of the NGIs, the regional operations team consisting of the regional Operator on Duty and the 1st Line Support and the oversight grid monitoring operators.

The version valid under EGEE III of 9th April 2010 can be found on https://wiki.egi.eu/wiki/Operations:Manuals . This version has been transferred by 1st May 2010 to the EGI wiki (https://wiki.egi.eu/wiki/Operations:OD) and the three documents are the OPS Manual is also available in pdf format on the EGI Document server. (https://documents.egi.eu/public/ShowDocument?docid=15). There are many inconsistencies in this current format. Re-Branding; the renaming of EGEE references to EGI is needed. All relevant links have to be updated. Wiki know-how has to be collected, i.e. on how to include common sections. New procedures have to be incorporated and references to tools that are not yet fully integrated like NAGIOS have to be incorporated.

OE-13 has recently started some kick-off meetings (https://www.egi.eu/indico/conferenceDisplay.py?confId=140) to coordinate the efforts of the involved key players, namely the NGIs and other involved parties like COD and operational tool providers in keeping the documentation and the training guides up-to-date and integrate the already active new procedures (see below).

The new current draft versions of the operational procedure manuals, the NGIs and Sites operations manual, the COD operations manual, ROD operations manual and the common section for them are collected in: https://wiki.egi.eu/wiki/Operations:Authors

The Operational Procedure Manual includes several procedures:

- Procedure for Changing or Implementing a new procedure and/or changing the OPS Manual
    - SA1-Pole2 (Gridops-procedures) collected suggestions for new procedures or redesigns inefficient procedures. Best Common Practices are taken into account. New procedures had to be discussed in a plenum with all ROC managers, who had to declare their consent to the new procedures. Afterwards the new procedures were integrated into the OPS manuals. Ops manuals were updated, and new versions were released in semi-periodic intervals. (This procedure needs to be reworked.)

- Actions needed for people to get started and introducing a new site. (needs reworking according to this MS.)
- Description of the duties and obligations of the relevant players within operations
- Policy on allowed and recommended site status flow transitions
- Procedures for site downtime scheduling
- Service intervention procedures
- Notification Mechanism documentation
- Incident reporting procedure
- 1st line support's handling of new incidents procedure
- Procedure on creating new entries in the knowledgebase
- Procedures on creating and changing, closing, escalting and reopening tickets
- Removing problematic sites and there especially emergency suspension
- Removing resources
- Procedure to add a test to the ROC_OPERATORS profile (accounting)
- Workflow and escalation procedure
- Site suspension procedure
- Handover procedures
- COD policy to collate knowledge sharing contributions, upgrade the OPS Manual and Make recommendations on criticality of tests and reporting on problematic procedures.
- Procedure on security incidents handling and interaction with OSCT-DC
- Grid security vulnerability handling process

Furthermore the Operational Manuals describe the communication channels to be used for contacting NGIs. New procedures need to make sure to keep this crucial information updated as well.

The Operational Manuals are backed up by Best Common Practices (BCP) and custom-tailored training guides.

- The Best Practices are found at https://wiki.egi.eu/wiki/Operations:Best_Practices . Like the OPS Manuals they have to be re-branded and all relevant links have to be updated. New practices have to be collected and included.
- The Training Guides are currently available through the CIC portal and EDMS (https://edms.cern.ch/document/1015741). GOCDB, GGUS documentation for the user (operations) will be transfered to the EGI wiki (https://wiki.egi.eu/wiki/GOCDB_Visualisation_Portal_User_Documentation ; https://wiki.egi.eu/wiki/EGI_Helpdesk).

Ticket Processing Management (TPM) procedures are not part of the OPS manual but can be found under https://gus.fzk.de/pages/support.php.

4.3.2. New procedures already in effect and passed through OMB

Those procedures are already defined and approved.

- New mechanism to collect statistics for availability and reliability: https://wiki.egi.eu/wiki/Availability_and_reliability_monthly_statistics#Description_of_the_process
- New vailability and reliability internal procedure for COD (a availability/reliability statistics followup procedure) https://wiki.egi.eu/wiki/Availability_and_reliability_internal_procedure_for_COD
- New procedure for creation and validition of a new NGI

  https://wiki.egi.eu/wiki/Operations:NewNGIs_creation

  Slides: https://www.egi.eu/indico/materialDisplay.py?contribId=5&materialId=slides&confId=75
- OLA between NGI and Site

  https://documents.egi.eu/public/ShowDocument?docid=31 defines the following requirements:

  - minimum tolerated availability: 70%,
  - minimum tolerated reliabilty: 75%.

4.3.3. Procedures currently under discussion

Draft versions of these procedures have already been presented to the OMB at least once.

- Site Registration and Certification

  https://wiki.egi.eu/wiki/Operations:Site_Certification (driven by NGI_PL, NGI_IT and NGI_GR)
- Retiring Grid Components

  https://wiki.egi.eu/wiki/Operations:RetiringGridComponent
- NGI/ROC Decommission

  https://wiki.egi.eu/wiki/Operations:EGEE_ROC_decommission
- Updated COD escalation procedure

  Slides by Ron Trompert at the OMB https://www.egi.eu/indico/contributionDisplay.py?contribId=3&confId=124

  https://wiki.egi.eu//wiki/Operations:COD_Escalation_new

Connected to the implementation in the Operations Portal:

- Collection of dashboard requirements regarding COD work (draft) https://www.egi.eu//wiki/Operations:COD_Dashboard_requirements
- COD Dashboard escalation procedure (draft)

  https://www.egi.eu//wiki/Operations:COD_Dashboard_escalation_procedure

4.3.4. Other Procedures in Development and Draft Status

- New procedure for middleware rollout to the infrastructure (first draft presented in MS402 https://documents.egi.eu/secure/ShowDocument?docid=5)
- New procedures on how to integrate resources in EGI-Production infrastructure (Will be created as a MS407 spinoff https://documents.egi.eu/secure/ShowDocument?docid=111, first of all for Nagios and GOCDB: tickets in RT in jra1queue and/or otag queue)
- EGI IGTF Release Process https://wiki.egi.eu/wiki/EGI_IGTF_Release_Process
- UMD Release Process https://wiki.egi.eu/wiki/Middleware:Release_Process
- COD Improvements to availability procedure

  https://wiki.egi.eu/wiki/Operations:COD_Improvements_to_availability_procedure
- Collection of GOC DB requirements regarding COD work (draft)

  https://wiki.egi.eu/wiki/Operations:COD_GOCDB_requirements
- Setting Nagios tests critical procedure

  https://www.egi.eu//wiki/Operations:Setting_Nagios_tests_critical_procedure
- A/R fixing procedure (draft) https://wiki.egi.eu/wiki/Operations:A/R_fixing_procedure
- Suspension procedure to manage sites that fail to provide justifications https://rt.egi.eu/rt/Ticket/Display.html?id=238

  (This will be reviewed when https://rt.egi.eu/rt/Ticket/Display.html?id=247 is done. It will be covered by general escalation procedure for sites that fail to handle operational issues in a timely manner.)
- Policy for keeping relevant contact information in GOCDB up-to-date (in preparation)

4.3.5. Security Procedures

All operational security procedures are collected in MS 405 [R 1].

(Input so far from Guiseppe Misurelli)

4.3.5.1. The integration into the EGI-CSIRT group

(a large majority of people involved come from the ex EGEE ROC security contact) of security experts from other MW stacks (I'm thinking about security officers rather then developers)

2. EGI-CSIRT has it's own plan for the development of security tools and what has been developed so far is mostly based on the gLite stuff so a discussion on how to cover other MWs is needed as well (quite the same for the extension of the Nagios tool to other MW stacks)

Anyway, I'll be very happy to help you reviewing the document giving the NGI operational

center viewpoint and helping Elisa as soon as I'll be back on August 23.

(need: contact EGI-CSIRT)

4.4. FUTURE OF PROCEDURES

There is general satisfaction with certain aspects of procedures: The operations portal with its collecting and overview role has a central role in being able to fulfil operational procedures, for example. It will have to be updated regularly to fit the needs of the current valid procedures and to ease their actual enforcement and execution. COD and ROD handover procedures over it provide a good and well documented record and history of events. Together with the information provided by the metrics non-functioning procedures are reflected and can be followed up. As already applied successfully earlier the role of BCPs for future procedure development has again to be enhanced and NGIs should actively try to contribute to them.

Future procedures will try to not rely on personal communications channels but on documented communication like on well defined mailing lists or tickets.

The site suspension procedure has been handled sloppily during EGEE III, but is emphasized now in EGI.

The downtime procedure has maybe to be rewritten to clarify some points. Some challenge i.e. the usefulness of AT_RISK downtimes since they are often wrongly used for very short outages instead of for warnings and information for the user in case of vacations or other situations of reduced on-site reliability.

However what is clearly needed in the current situation to keep track of what is going on is a 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. OE13 is coordinating the efforts to create such a reference sheet.

5. 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.

5.1. OPERATIONAL REQUIREMENTS COMING FROM NGIS

Operational requirements will continue to be collected from NGIs that are interested in integrating novel resource types into their e-Infrastructure as required.
Some of these NGIs and the novel resource types that they are planning to integrate into the production infrastructure are:

- Integration of UNICORE and Globus services: NGI-DE
  Germany is the lead partner of the IGE project (http://www.ige-project.eu/ about Globus support in Europe), the lead partner is Leibniz Supercomputing Centre.
- Integration of desktop services: NGI-HU. Hungary is leading the EDGI project http://edgi-project.eu/
- Integration of cloud services: NGI-FRANCE and NGI-IBERGRID. France and Spain are involved in the StratusLab project

- Integration of storage resources into accounting: Italy and possibly other NGIs that are pioneers in this field. (Paolo Veronesi and Andrea Cristofori from the Italian Grid Infrastructure NGI), Paolo.Veronesi@cnaf.infn.it, Andrea.Cristofori@cnaf.infn.it)
- MPI accounting: Italy is certainly interested in this, together with Spain. Other NGIs from SEE region such as Turkey and Bulgaria have expertise/requirements in this, according to slide 8 of their presentation given at the SA1 kick-off meeting (https://www.egi.eu/indico/sessionDisplay.py?sessionId=9&slotId=0&confId=43#2010-06-04).

The call for participation to dedicated meetings will be open in such a way that any NGI that wishes to contribute is welcome.
We hope to present some of the gathered requirements already in the second version of this milestone. After collecting a set of requirements, the operational interfaces described in this milestone can evolve accordingly.
We would also like to see requirements for the interoperation of different operational stacks coming from NGIs and sites which run more than one MW.

5.2. REQUIREMENTS COMING FROM COLLABORATIONS WITH OTHER DISTRIBUTED INFRASTRUCTURES

When looking abroad to other infrastructures, in especially HPC oriented infrastructures like DEISA and PRACE, we can ask ourselves whether we can learn from each other and unify and improve certain aspects of our infrastructures to make the life of the user easier.
Currently the percentage of users that is interested in capacity as well as capability computing at the same time is rather low. It should be our obligation as infrastructure providers to direct the users to the infrastructure that suits their use case best and reduce the number of barriers they experience on the way, so that at least shifting from one infrastructure to the other should be a more smooth and transparent process.
Generally the user needs to be identified to get access to the infrastructure, needs to allocate the resources and the knowledge of how to make best use of the existing structure, needs support and needs some job monitoring (accounting from the users point of view).
In some cases the access application procedures for the different infrastructures are already very similar. Collaboration between the different infrastructures is even more interesting than interoperability. If the user interaction points of the different interfaces look even more similar, the user doesn't have to experience a steep learning curve, when switching infrastructure and it puts a lesser burden on the tool developers. The future will hopefully see several highly specialised infrastructures providing together one single service for the user. The right system should be chosen according to the computational needs to avoid inefficient usage of a certain infrastructure.
Different groups are currently concerned with human and technical interoperation of different distributed infrastructures. The e-Infrastructure Forum http://www.einfrastructure-forum.eu/ for example brings together networking layer, HPC, HTC, and European data providers. The OGF working group IPG (Infrastructure Policy Group) gets supported from both EGI/DEISA as well as OSG/TG and focuses more on technical details.
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 seen in authorisation, in resource allocation (project model vs VO model) as well as in responsibilities and ways of user administrations (e.g. site adminstrated LDAP vs VO administrated VOMS).

Several topics of common interest can be identified when looking at collaboration between different distributed infrastructures:

- A common mechanism for resource allocation,
- Large collaborations with integrated accounting information,
- Budget of computing allocated per year. Sometimes the peer review is bypassed if project is recognized.

Two more points are discussed in greater detail:

5.2.1. A common support network for different infrastructures

One possible identified starting point in glueing different distributed infrastructures together would be a single support helpdesk or at least having an agreed interface of exchanging trouble tickets between different infrastructures.

In the immediate future we would like to investigate the possibility to (automatically) route tickets from EGI (GGUS) to PRACE (queue based RT) and to have one single support entry point to users.

5.2.2. Core procedures and Operation Level Agreements

We would like compare the current procedures of our infrastructure with others and be able to synchronize future development similar as already done for security. (ref again)

Especially are we interested in sharing our OLAs on availability and reliability and our operational procedures and policies.