EGI-InSPIRE

Integrating Resources into the EGI Production Infrastructure

EU MILESTONE: MS414

Document identifier:	EGI-MS414-V0-8.doc
Date:	**19/07/2011**
Activity:	**SA1**
Lead Partner:	**KTH**
Document Status:	**DRAFT**
Dissemination Level:	**PUBLIC**
Document Link:	https://documents.egi.eu/document/650

Abstract

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

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

I. COPYRIGHT NOTICE

II. DELIVERY SLIP

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

III. DOCUMENT LOG

Issue	Date	Comment	Author/Partner
0	05/07/2011	Incomplete placeholder	Michaela Barth /KTH
1	06/07/2011	ToC	Michaela Barth /KTH
2	12/07/2011	Input on Operations Portal	Cyril L'orphelin / IN2P3
3	19/07/2011 27/07/2011 29/07/2011 01/08/2011	Input on GOCDB, Input on Argus in general, Argus and gLite Input on GOCDB Input on Argus and ARC Comments	David Meridith / STFC Alvaro Simon / CESGA Torsten Antoni / KIT Ali Gholami / Nordugrid Michaela Barth / KTH
4	02/08/2011	Input on Argus and UNICORE	Krzysztof Benedyczak / UWAR
5	02/08/2011	Some corrections	
6	04/08/2011	Input on Accounting	John Gordon / STFC
7	04/08/2011	Going through general parts	Michaela Barth / KTH
8	05/08/2011	Going through accounting input Input on Monitoring	Cristina del Cano Novales / STFC Emir Imamagic / SRCE

IV. APPLICATION AREA

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

V. DOCUMENT AMENDMENT PROCEDURE

Amendments, comments and suggestions should be sent to the authors. The procedures documented in the EGI-InSPIRE "Document Management Procedure" will be followed: https://wiki.egi.eu/wiki/Procedures

VI. TERMINOLOGY

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

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

VII. PROJECT SUMMARY

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

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

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

The objectives of the project are:

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

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

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

VIII. EXECUTIVE SUMMARY

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

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

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

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

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

~~Finally~~Then, this document will give a short~~n~~ overview of the status of operational procedures and policies

needed for the integration of ne~~w resources~~w resources.

Finally, we discuss further integrational requirements coming from different sources, like NGIs, other DCIs and above all from our successful integration task forces- and conclude with some future plans.

TABLE OF CONTENTS

1 INTRODUCTION

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

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

Different resources will use different middleware components. EGI-InSPIRE will support the Unified Middleware Distribution (UMD) for deployment on the production infrastructure.

The UMD integrates middleware components provided by the European Middleware Initiative project (EMI), by the Initiative for Globus in Europe (IGE) project, and other external sources called "Community Contributions". Services from the gLite, ARC and UNICORE middleware stacks ~~will be~~are included in the EMI releases. 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 SAM/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 partly inherited from ~~the~~previous experiences within the EGEE projects. ~~Although, these~~The exact operational tools used may change in the future, but the current valid set ~~they~~ provides a good starting point when comparing the interoperability of different middleware components for each of the operational tools currently in use.

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

2 INTEGRATION OF MIDDLEWARE ON OPERATIONAL TOOL LEVEL

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

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

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

EGI will need to deploy several middleware stacks according to the requirements of users and site managers. Presently, gLite and ARC can be viewed as~~as a result of the EGEE and WLCG projects, only gLite is~~ fully integrated into all the operational tools, whilst some smaller adaptations are still needed due to changed and more standardized interfaces of the operational tools enabling broader access to other types of middleware. ~~ARC has been partially integrated, and for~~ ~~–~~Globus and UNICORE operational integration is in full progress also thanks to the specialised integration task forces~~still incomplete~~. Comprehensive integration is a short-term objective of the first phase of the project.

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

2.1 *Interoperation at an Infrastructure Level*

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

MANAGEMENT INTERFACE

An important operational interface of a resource is the capability to be put in downtime if under maintenance, the capability to undergo a certification process and thereby reach production status, and the capability to be monitored to assess its operational security level.

Within EGI the GOCDB is the tool of choice for fulfilling these management tasks. It portrays what services are running where and who to contact on a management and technical level as well as in case of security issues.

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

MONITORING INTERFACE

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

ACCOUNTING INTERFACE

Accounting ensures the quality of service by providing useful planning and usage information. After all, the key feature of an operational infrastructure is that the resources have high availability and reliability. The summary data of the amount of actually delivered computing resources is relevant for both VOs and project communities as well as for each site to check if their service level agreements have been fulfilled. Current EGI accounting is based on APEL which is ~~moving to~~now embracing~~e~~ the **common messaging interface used by other operation tools and to provide a standard web service interface to read records published by partner grids.**

SUPPORT INTERFACE

The provision of 3rd level support is equally important as the quality of the provided services. EMI has set up a 3rd level structure within the EGI Helpdesk (GGUS) for its various middleware stacks and services. The EGI Helpdesk has been adopted as a common infrastructure to exchange trouble tickets between different stakeholders following its use during the EGEE projects.

DASHBOARD INTERFACE

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

USERMANAGEMENT

Although not explicitely being an operational tool per se, the used user management interfaces for authentication and authorization influence the work with all the other operational tools. A common standard based interface to a top layer enabling general user management is therefore necessary as well.

2.2 Overview Status of Middleware Integration for each Operational Tool

	gLite	ARC	UNICORE	Globus
GOCDB	completed	completed	First services have been registered	First services have been registered
Nagios - Probes written, Probes integrated, Definition of an operational set	completed	completed (integration into SAM release 7)	Probes have been written, integration foreseen for SAM release 13	Probes have been written and will be supported by IGE in the future, integrated in SAM release 11, definition of an operational set problematic without certified sites.
Operational Dashboard	completed	completed	to be done (should work automatically after definition of an operational set of Nagios probes)	to be done (should work automatically after definition of an operational set of Nagios probes)
Accounting	completed	completed	In progress	In progress
3rd level support in GGUS	completed	completed	completed	completed

(access to expert teams via DMSU and ev. EMI)				

Table 1: Outlining the current status of interoperation for each MW stack relative to the current set of operational tools

2.3 Definition and Description of a Management Interface

2.3.1 Functionality

A management interface is an operational interface which allows sites to store, maintain and view the topology of the production infrastructure and the basic information about the respective resources within it.

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

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

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

2.3.2 Requirements

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

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

Such a database needs a role based interaction model, so that people responsible for certain sites, services or resources can update and maintain the various entries representing the entities under their responsibility within typical daily operations scenarios. In particular, basic service status information shall be easily viewable and changeable. It shall be easily possible to register a service of a known service type, to edit system administration information and put whole sites or single resources in and out of downtime according to predefined procedures. It shall be easy to identify whether a resource is monitored or not by the corresponding monitoring system. This monitoring bit can be set separately or implicitly within the different production states.

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

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

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

2.3.3 Integration of new Services into GOCDB

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

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

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

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

2.3.3.1 Procedure for registering new Service Types

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

1. If the service type is already registered in GOCDB, service endpoints can be added by users of GOCDB following the established procedure.
2. If the service type is not registered, a request should be made to the OTAG for its inclusion in GOCDB (e.g. by the new middleware provider or JRA1 community). If the new service type belongs to a previously undeclared middleware stack, then a strategic decision is required to ensure only officially supported middleware is integrated into GOCDB. If the request is approved, it is communicated to the GOCDB developers to add the new service type.
3. The requesting party is notified (either the request is rejected or completed).

2.3.3.2 Regular review of the list of available service types

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

2.3.3.3 Integrated operational tool and core service types

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

- **egi.GSTAT**: [Central service] The central GStat portal
- **ngi.OpsPortal**: [Regional service] NGI-level regional operations portal instance

2.3.3.4 Integrated gLite service types

- **CE**: [Site service] The LCG Compute Element. Currently the standard CE within the gLite middleware stack. Soon to be replaced by the CREAM CE.
- ~~**gLite-CE**: [OBSOLETE Site service] The gLite Compute Element is now obsolete and is not supported. Please avoid using this middleware service.~~
- **CREAM-CE**: [Site service] The CREAM Compute Element is the new CE within the gLite middleware stack.
- **APEL**: [Site service] This is a "dummy" Service Type to enable the monitoring tests for APEL accounting. All sites must have one instance of this Service Type, associated with a CE.
- ~~**MON**: [OBSOLETE Site service] The gLite MonBox hosts the site R-GMA services.~~
- **UI**: [User service] The User Interface. Can be installed by users but more commonly installed by a site.
- **SRM**: [Site service] Storage Resource Manager. Mandatory for all sites running an SRM enabled storage element.
- ~~**Classic-SE**: [OBSOLETE Site service] The Classic Storage Element is now obsolete and is not supported. Please avoid using this middleware service.~~
- **Central-LFC**: [Central service] An instance of the gLite file catalogue which holds entries for all files owned by a particular VO. NOTE: An LFC can be both Central and Local.
- **Local-LFC**: [Site service] An instance of the gLite file catalogue which holds entries for files owned by a particular VO, at your site. NOTE: An LFC can be both Central and Local.
- **WMS**: [Central service] gLite Workload Management Service. Acts as the broker for matching user jobs to available computing resources.
- ~~**RB**: [OBSOLETE Central service] The LCG Resource Broker is now obsolete and is not supported. Please avoid using this middleware service.~~
- **VOMS**: [Central service] VO Management System. Part of the authentication and authorization system. This service only needs to be installed on the request of a VO.
- **LB**: [Central service] gLite Logging and Bookkeeping. Usually installed by sites running a WMS. One LB service can support several WMS instances.
- **AMGA**: [Central service] gLite metadata catalogue. This service only needs to be installed on the request of a VO.
- **FTM**: [Site service] gLite File Transfer Monitor. Monitors the FTS service at a site.
- **FTS**: [Central service] The gLite File Transfer Service manages the transfer of files between sites. This service only needs to be installed on the request of a VO.
- **VO-box**: [Site service] The gLite VO box allows a VO to run their own services at a site. This service only needs to be installed on the request of a VO.
- **gLite-APEL**: [Site service] The gLite-APEL hosts the site Accounting client (3.2 replacement of the MonBox)
- **gLExec**: [Site service] A light-weight gatekeeper to authenticate and authorize credentials according to local site policy and execute commands.

2.3.3.5 Integrated ARC service types

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

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

2.3.3.6 Integrated UNICORE service types

- **unicore6.Registry**: [Central service] All UNICORE services register here; clients ask the registry for available services in the Grid. Normally one Registry per Grid infrastructure which collects URLs of services.
- **unicore6.Gateway**: [Site service] Sits in front of one or more UNICORE services as a gateway to the internet. Normally one Gateway per site.
- **unicore6.TargetSystemFactory** [Site service] used as an entry-point for submitting single jobs. It can create Target System Services (TSSs) and submit jobs to those TSSs.
- **unicore6.StorageFactory** [Site service] Creates StorageManagement instances. A user can create dynamic storage management services for own purposes with it. Often used to provide file space during workflow execution.
- **unicore6.StorageManagement** [Site service] Provides an abstract file system-like view on a storage resource. A Storage Management Service (SMS) can be created by a Storage Factory or can be configured statically way by a configuration file.
- **unicore6.ServiceOrchestrator** [Site service] Handles dispatching of a workflow's atomic jobs, and brokering. Normally there is one per grid infrastructure.
- **unicore6.WorkflowFactory** [Site service] Used as an entry point for submitting workflow jobs. The Workflow factory is creating workflow instances and can submit workflows to them. It is the workflow submission equivalent to the Target System Factory used for single job submission.
- **unicore6.UVOSAssertionQueryService** [Site service] Provides data and user information via the SAML standard as needed for authorization and environment customization.

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

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

2.3.3.7 Integration of Globus resources

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

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

2.4 *Definition and Description of a Monitoring Interface*

2.4.1 Functionality

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

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

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

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

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

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

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

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

2.4.2 Requirements

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

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

2.4.3 Interoperability of different MW Stacks with SAM/Nagios

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

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

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

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

SAM/Nagios instances are ~~supposed to be~~now deployed ~~at each~~on all NGIs.

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

Since SAM Update-07 release (30th November 2010) SAM fully supports using ATP as a topology provider. ATP is currently fed with information from both GOCDB and BDII. ATP extracts VO mappings from the BDII as those are not present in GOCDB. When EMI provides a single information system which will integrate all supported middlewares, it will be integrated into ATP.

2.4.3.1 Tests and Nagios probes for gLite resources

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

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

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

2.4.3.2 Tests and Nagios probes for ARC resources

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

~~Nagios ARC probe developers prepared a set of Nagios probes for ARC which they tested in an independent local Nagios instance and started integration with SAM. The problem at the start of the integration was that these procedures had not yet been prepared. Since the integration of ARC probes was already approved beforehand, an RT ticket was created directly in the JRA1 queue and further progress was followed through the developers ticketing system [R 47].~~

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

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

~~The biggest issue encountered during integration was the lack of a multiple middleware UI.~~

~~Currently, it is not possible to install both glite UI and ARC Client by using packages on the same machine. Therefore an alternative approach based on an ARC Client standalone package was chosen and implemented. This requires Nagios administrators to perform additional steps described in [12].~~

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

ARC probes are fully integrated with SAM starting from the release Update-7 and they monitor the ARC-CE service. The ARC GridFTP service will be monitored with probes for standard GridFTP service.

Maintenance of ARC-CE probes is done by the EMI ARC product team.

The ARC monitoring tests became operational on 7.04.2011.

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

2.4.3.3 Tests and Nagios probes for UNICORE resources

UNICORE probes are provided by NGI_PL. The probes monitor the following services:

- Gateway
- UVOS
- Registry
- ~~UNICORE/X~~
- Workflow Service
- Service Orchestrator
- Global Storage

More details on those probes can be found at (add ref to http://alfred.studmat.umk.pl/%7Eszczeles/PL-Grid/UMI-Probes.html).

Integration of probes with the SAM is ongoing and planned for the release Update-13.

Maintenance of UNICORE probes will be done by the respective UNICORE EMI product teams.

2.4.3.4 Tests and Nagios probes for Globus resources

Globus probes are fully integrated with SAM starting from the release Update-12. The probes monitor the following services:

- GSI-SSH
- GridFTP
- GRAM
- MyProxy.

Other services (e.g. LDAP) and basic checks (e.g. port checks and certificate lifetime) are covered by the same tests used for gLite services.

Maintenance of Globus probes will be done by the IGE project.

2.4.4 Procedures to integrate new Nagios Probes

There are some procedures in the Availability and Monitoring area. For the integration of new resources namely two of them are relevant:

- “Adding new probes to SAM”, [https://wiki.egi.eu/wiki/PROC07] approved by the OMB in March 2011, a procedure for adding new OPS Nagios probes to the SAM release.
- “Setting a Nagios test status to OPERATIONS”, [https://wiki.egi.eu/wiki/PROC06] , approved by the OMB in November 2010: A Nagios probe is set to OPERATIONS when its results are used to generate notifications for the Operations Dashboard. This procedure details the steps to turn a Nagios test to OPERATIONs.

2.5 Definition and Description of an Accounting Interface

2.5.1 Functionality

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

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

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

2.5.2 Requirements

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

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

2.5.3 Current Status

The core EGI Accounting Infrastructure is based on APEL [R 34]. Other systems interface to APEL to collect data in one central place. The collected CPU accounting records and the data summarized by site, date, VO, and user are displayed in the Accounting Portal [R 43] and can be visualised by dynamic queries on the parameters above at any level of the hierarchical tree structure which defines EGI and its partner grids.

The bulk of existing sites collect data from their batch systems (LSF, Torque; SGE, Condor), which are joined with the job's user grid credentials and published to the central APEL repository. At the time of writing the EGI infrastructure is in transition of the transport layer from a private ActiveMQ broker to the production broker network already used by other EGI Operational Tools. The new system uses STOMP and Python to define a messaging model with encryption, verification, and acknowledgements. Other partner grids (Open Science Grid and NDGF), and a few sites with their own accounting services, currently publish summaries of data in the form described above directly into the APEL central repository. Sub-grids of EGI (e.g. Italian Grid Infrastructure IGI) publish all of their VOs data. Partner grids (e.g. Open Science Grid OSG) publish selective VOs. In particular the LHC VOs are all published to APEL so that there is a single worldwide repository for LHC. These alternative publishers will move to the new publishing method so that there will no longer be any direct database insertion.

CPU data is published in the form of either: job level records (JR) containing data from a single batch job; or summary aggregate job records (SJR) containing totals for a number of jobs run at a single site for a single user and VO in a given month. The Job Usage Record (UR) schema is a plain text version of the OGF-UR v1.0 with some common extensions. For example, the original UR did not have the concept of a site, which is so crucial to the grid. The summary record has been submitted to OGF's UR-WG for possible adoption as a community standard [R 35].

The OGF UR Working Group (UR-WG) is considering a proposal from EMI for a UR for storage accounting. It is anticipated that this will be integrated into the same APEL infrastructure once implemented on the relevant storage products.

EMI also has a group reviewing the implementations of the OGF UR for compute [https://twiki.cern.ch/twiki/bin/view/EMI/ComputeAccounting] to agree on the semantics of the existing UR and existing common extensions and possibly propose further extensions.

2.5.4 Integration with other Infrastructures

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

a) Define a structure for their grid in GOCDB (or equivalent) that can be used by the accounting portal to display the data. The minimum requirement is a flat set of site names, used in the accounting records. (e.g. for OSG these data are obtained from MyOSG)

b) Extract data from their accounting system grouped data by site/VO/User/FQAN/month and create each group into a 'summary record' meeting the APEL definition. Experience shows that for accounting systems using the OGF-UR this is a simple transformation.

c) Other infrastructures running a gLite CE (lcg-CE or CREAM) could run our software to aid collecting accounting data. Infrastructures running other middleware stacks who run one of the currently supported batch systems listed above can take our data collectors to parse the raw accounting data collected by the batch system to which they will then need to add the CPU speed and user/VO credentials, before publishing.

d) Register the publisher with APEL (by providing the host DN to the EGI APEL support unit). The APEL Repository only accepts accounting records from registered sites. For APEL client sites this is defined by the glite-APEL service type in in GOCDB. An equivalent mechanism will be developed for summary publishing sites/grids.

e) Publish the records into EGI's ActiveMQ Message Bus using the agreed encryption framework. The APEL repository will accept the records into a holding container from where they will be merged with the summaries from other grids and the summary produced by APEL from the job records it has received. Currently, the master summary is rebuilt from scratch several times per day. Each time it uses the last set of summaries received from each grid.

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

2.5.4.1 Issues

- For the aggregation of user data it is assumed that all interoperating infrastructures use a user identity based on X.509 certificates signed by IGTF recognized Certificate Authorities.
- While a worldwide community management service like VOMS makes the aggregation of VO accounting data from different infrastructures simple, it would be feasible to implement a VO name transformation to combine the data from infrastructures who have named the same VO differently.
- Another issue is the unambiguous mapping of user accounts to VOs. In some cases users might belong to more than one VO in which case identifying to which VO the utilization results would go is not possible. Extra effort will be needed to check the fulfilment of arranged pledges.
- The issue of exchanging data identifying a user has been a contentious one. It is frequently asserted that this is illegal under the laws of certain countries. Extensive research was undertaken by the Joint Security Policy Group (JSPG) in EGEE-III during the development of the Grid Policy on the Handling of User-Level Job Accounting Data [R 45] with the result that legal advice was given that with the appropriate acceptable use policy and the agreement signed by the user and by the site running the accounting repository, then the collection, storage and restricted display of data identified by UserDN is acceptable. This issue might have to be re-evaluated again when exchanging accounting data with other infrastructures like e.g. DEISA.
- Current accounting is only of CPU of batch jobs but the interfaces between infrastructures should also allow the integration of other types of accounting record as they are developed. JRA1 will start work on new accounting types in year 2.
- The currently agreed unit for normalization of CPU time in EGEE, EGI, and WLCG is HEPSPEC06 hours [R 46]. For interoperation with an infrastructure that does not collect this value from the resources running jobs, some conversion factor must be negotiated.

2.5.4.2 Future Work

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

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

Should also refer to JRA1 MS706 describing the Operational Tools accounting work plan here!

2.5.4.3 ARC resources

Accounting integration was performed already during EGEE III. The aim was to gather and export accounting from the Nordic T1 and T2s, which for the compute part were based on ARC, and send data for selected VOs to the APEL central repository so they can be viewed with the EGI Accounting Portal. ARC-CE supports accounting via SGAS (SweGrid Accounting System, [R 19]) and an automatic script for exporting the accounting info gathered in SGAS to APEL was set up [R 20]. Currently, only LHC VOs are published to APEL but this could easily be extended to other international VOs.

The SGAS-APEL interface should be changed to the new once discussed above. This should be straightforward as the extraction and selection phase will not change, only the transport layer from JDBC to ActiveMQ.

2.5.4.4 UNICORE resources

Currently no means of collecting accounting and usage records are directly implemented within UNICORE. Instead, this is done directly via the underlying batch system, see for example as in the DEISA project, where the accounting data is converted into OGF-UR format and provided according to XUUDB access control.
Accounting services for UNICORE have been developed by NGI_PL and NGI_BY. These are being reviewed within the UNICORE community. D-Grid within the NGI_DE is also building a regional service to collect accounting data from UNICORE and other clients. Whichever one or more of these is deployed should add the common interface to publish data onwards to the EGI central repository. Discussions have started with the developers on this.

2.5.4.5 Globus resources

IGE has adopted GridSAFE as its accounting solution. It is currently under test. GridSAFE was designed as a site accounting repository to collect data locally but it has the interfaces to accept data over the WAN too so it could act as a regional repository receiving data from a number of sites.
From the specification it does not have the ability to publish data on to higher levels in a hierarchy of repositories. It relies on others pulling data from it through an OGF RUQI interface rather than the EGI push model. However a proof of concept was carried out in NGI_UK to use their Globus RUS client as a backend to GridSAFE to push data on to a remote RUS. This implies that data can be extracted so the APEL publishing model could be made to work.

2.6 Definition and Description of a Support Interface

2.6.1 Functionality

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

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

For EGEE, and now EGI, a functional body has been defined to keep track of the ticket processing management (TPM). The TPM keeps a global overview of the state of all tickets and is responsible for those tickets that have to be assigned manually, i.e. so that they get forwarded to the correct support units. The TPM teams act as a 1st line support chain and have also to keep track of long-term trouble tickets and help to solve them with their very good general grid knowledge. In this way, a problem submitted to GGUS can be quickly identified as either a grid problem or a VO specific problem and addressed to the appropriate second line specialized support units or the dedicated VO support teams whose members have specific VO knowledge.

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

2.6.2 Requirements

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

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

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

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

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

2.6.3 Integration of new Resources into GGUS

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

2.6.3.1 Integration of a new Resource Centre into the infrastructure

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

2.6.3.2 Integration of a new NGI into the infrastructure

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

- The simplest way, which might be suitable for a small new NGI is to use GGUS directly. This has the limitation of just one support unit for the whole NGI. Tickets cannot be assigned to specialized groups or specific resource centres within the NGI. This further processing of the tickets is done independently from the EGI support infrastructure.

- The NGI can make use of xGUS a customisable slimmed-down regional instance of GGUS. xGUS is hosted and maintained by the GGUS team. Customization can be done via an administrative web interface, which enables creating and managing support units and defining special workflows. xGUS comes with the interface to GGUS built in.

- The NGI can set up its own ticket system. In this case the NGI has to make sure that their ticket system fulfils the requirements of the interface definition to GGUS. The NGI ticket system needs to be interfaced to GGUS and the NGI is responsible for maintaining this interface. This for example includes testing the interface after releases of the GGUS portal.

- Details on the NGI creation process can be found on a dedicated page in the wiki [R 37].

2.6.3.3 Integration of a new Technology Provider into the infrastructure

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

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

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

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

For details on the Technology Helpdesk refer to MS410

2.7 Definition and Description of a Dashboard Interface

2.7.1 Functionality

In order to operate a distributed infrastructure, management and monitoring information has to be collected and presented in a labour saving way to assist the operators of the infrastructure in their daily work. The dashboard interface combines and harmonizes different static and dynamic information and therewith enables the operators to react on alarms, to interact with the sites, to provide 1st line support and/or to really operate the sites by creating and supervising problem tickets on regional as well as central level.

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

2.7.2 Requirements

A dashboard interface has to fulfil the functionality described above.

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

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

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

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

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

2.7.3 The Operations Portal

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

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

The goals of Lavosier are to provide:

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

This solution design means that the web application does not need to know the exact location of the data provider and neither which kind of technology has provided the information initially. All these concerns are already taken into account by Lavoisier.

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

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

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

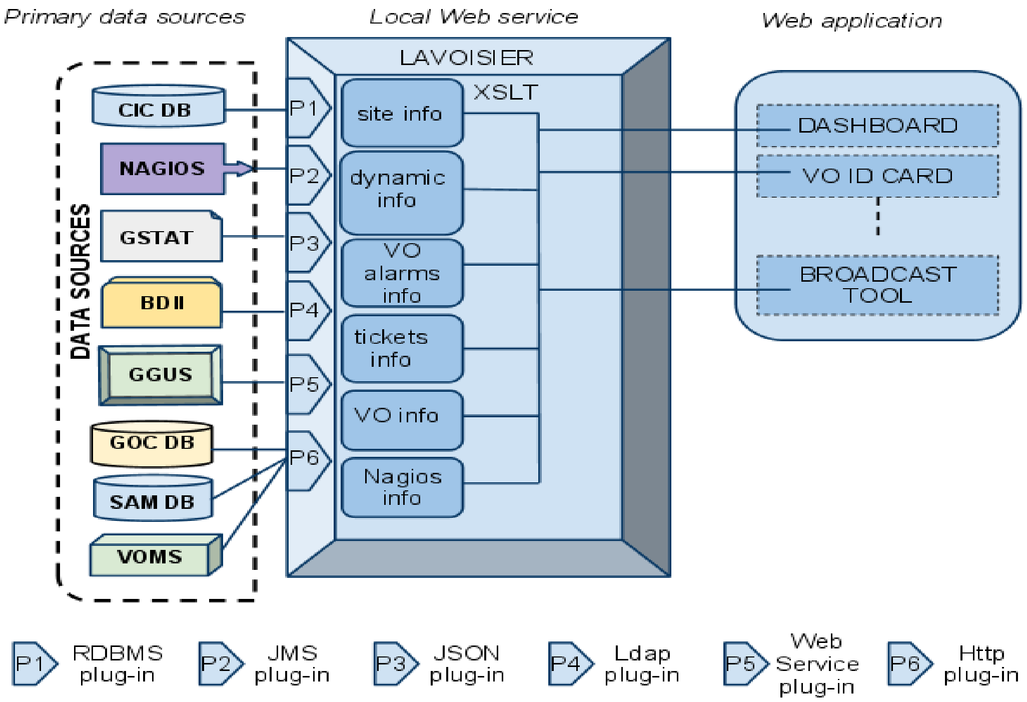

Fig. 1: Global architecture of the Operations Portal.

2.7.3.1 Integration of a new resource

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

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

If a site and its resources are already integrated in all the other operational tools through existing information providers (e.g. registered in GOCDB, monitored by Nagios, publishing their information via BDII and having a tree in GGUS), existing plug-ins can be reused and no additional integration effort for the usage of the Operations Portal is needed.

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

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

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

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

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

2.7.3.2 Alternative possibilities to integrate new information providers

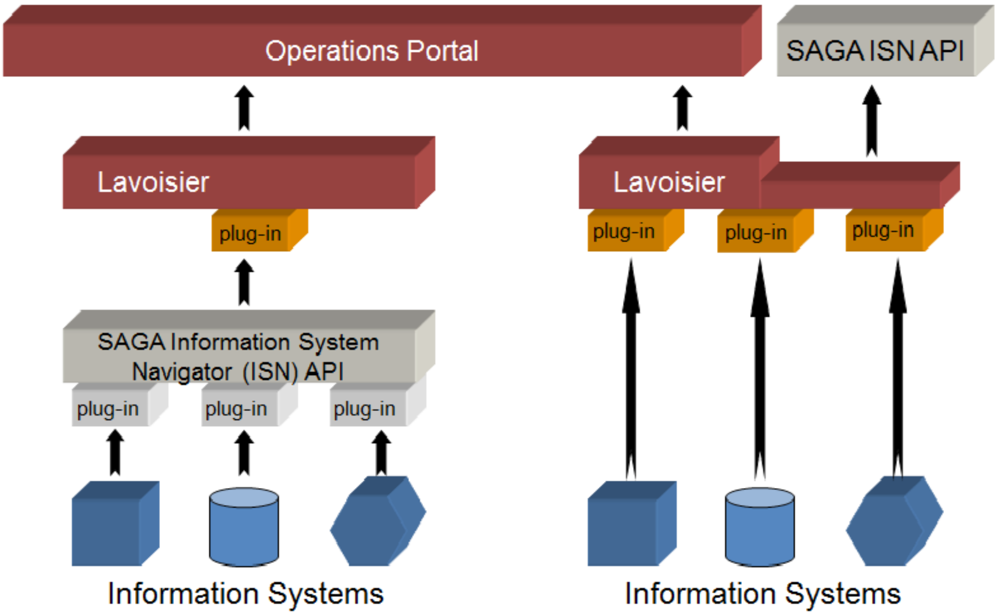

Fig. 2: Integration of new information systems into the Operations Portal

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

2.7.3.3 Integration of a gLite resources

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

2.7.3.4 Integration of a ARC resources

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

2.7.3.5 Integration of a UNICORE resources

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

2.7.3.6 Integration of a Globus resources

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

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

2.8 User Management, Authentication and Authorization

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

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

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

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

Normally in a VO, the VO Manager assigns attributes and membership to people and this is controlled by the VOMS, but the sites can not influence this information. However a site sometimes wants to control access in more fine grained detail: like to ban one user from a certain VO, or limit the access to some of the resources.

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

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

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

2.8.1 Desired Functionality of a user authorization system

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

2.8.2 Requirements on a user authorization system

We have different requirements:

- Identical user mapping functionality
    - It should be possible to use a centralized approach to do the DN/UDI mapping in a consistent approach. Solutions based on shared file system or shared pool directory are not acceptable as they add dependency to the middleware since not all of them are POSIX compliant. Besides in case of multiple users, each try to overwrite a shared entity which cause inconsistency.
- Policy based user access
    - site administrators should be able to ban users based on DNs, CAs, VOs for the whole site or over multiple services.
    - The banning list and other policies can be created and written down in a well defined way, e.g. by using a language to create and customize policies like the Simplified Policy Language (SPL) as used in Argus.
- Support for single-user and multi-user pilot jobs
    - Pilot jobs are submitted through pilot submitter and the real owner of the jobs until they start execution on the worker nodes are not known which is important in the case of accounting. Using e.g. Argus as a centralized service, it should be possible to map users to a particular POSIXUID/GID.
    - This requirement is possibly not equally urgent as the other two, since authorization problems are only expected for multi-user pilot jobs.

2.8.3 Argus

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

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

The interfaces to the PAP and PDP daemons are standardized and well defined.

The EMI XACML working group is aiming at standardizing the XACML attributes https://twiki.cern.ch/twiki/bin/view/EMI/EmiJra1T4XACML used in the ~~policies~~requests.

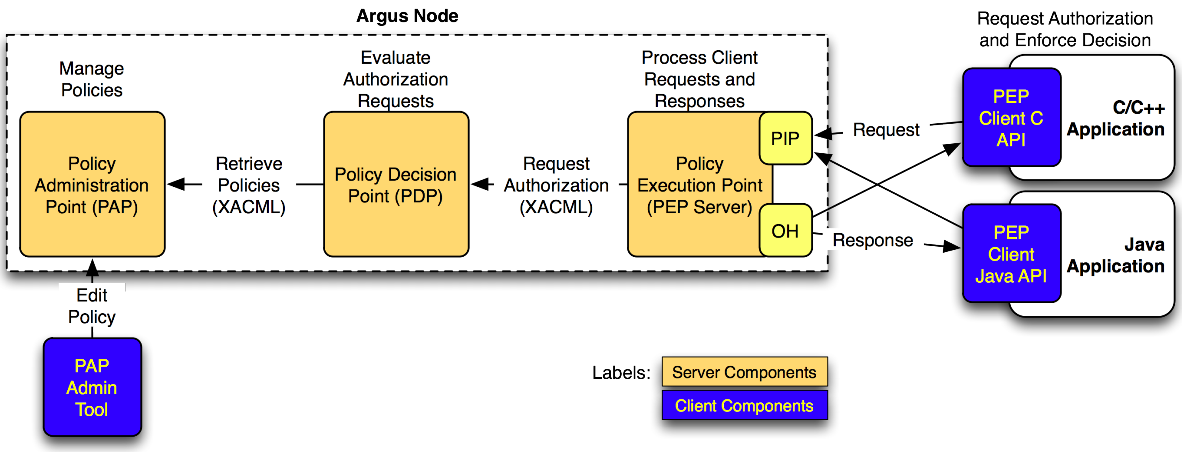
Argus Node
Manage Policies
Policy Administration Point (PAP)
Retrieve Policies (XACML)
Evaluate Authorization Requests
Policy Decision Point (PDP)
Request Authorization (XACML)
Process Client Requests and Responses
Policy Execution Point (PEP Server)
PIP
OH
Request
Response
Request Authorization and Enforce Decision
PEP Client C API
C/C++ Application
PEP Client Java API
Java Application
Edit Policy
PAP Admin Tool
Labels:
Server Components
Client Components

Fig. 3: Internal Argus Components

The three so far presented Argus components (PAP, PDP, PEP) are responsible for authorization. Argus-EES is the component which maps DNs to particular POSIXUID/GID. It is normally contacted by the PEP. But not all middleware stacks are using PEP. It has also to be noted that Argus supports hierarchical policies since a PAP can use another PAP.

2.8.3.1 Argus and gLite

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

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

- CREAM: Argus policies will grant access to grid users to access CREAM-CE computing resources. When a new user job is submitted to CREAM the site Argus instance is requested to accept or deny the job submission based on the site Argus policy.

- WN/gLExec: Pilot jobs can be mapped to a specific grid user based on Argus policy reponse instead of SCAS. Pilot jobs are mapped to grid user into WN following the Argus site security policy.

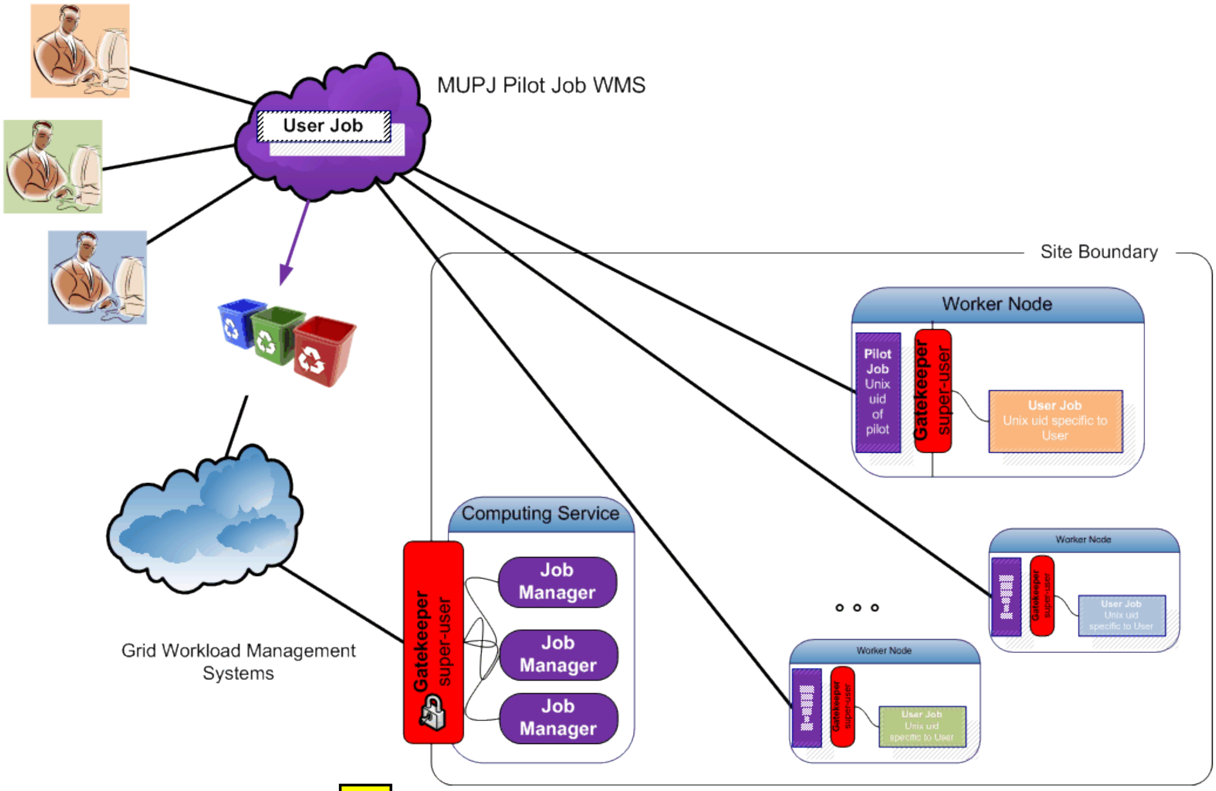

Fig. 4: gLExec Infrastructure

2.8.3.2 Argus and ARC

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

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

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

Currently as a proof of concept an Argus provided client is in charge of sending/receiving messages to the PEP daemon. However, eliminating communication to the PEP daemon from the ARC authorization client will increase the performance and can be achieved through providing a profile for ARC in Argus.

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

2.8.3.3 Argus and UNICORE

For the case of UNICORE what normally is referred to as authorization is split into two terms: "authorization" in UNICORE means the decision if a certain request is allowed or not; "incarnation" in UNICORE means to map a request to a local system (what includes more than in e.g. in the case of gLite: not only UID/GID(s), but also symbolic application names are mapped, as well as symbolic arguments, execution environments, etc.).

The current working state for authorization can be described as follows: UNICORE has a built in mechanism called PDP which is responsible for the actual authorization. The administrator can choose its ~~type of~~ implementation. The default implementation uses a file based authorization policy. This default XACML based policy ~~allows the administrator to assign a special~~predefines~~d~~ attributes to allow/ban a user. Therefore authorization is typically administrated by assigning attributes for users using tools of choice: UVOS, XUUDB, files. XACML policy is modified only ~~to ban or allow access also for~~in case of complicated use-cases (e.g. banning all users of a certain VO but only at night). So in the case of UNICORE authorization can already be controlled to the desired level - without using Argus. Argus can be seen as an intermediate solution: its usage will allow for— more flexibility than ~~it~~ is provided by assigning attributes while~~and~~ still allowing administrators not to learn a complicated XACML syntax. However a really advanced authorization problem~~s~~ w~~st~~ill st~~w~~ill require manual XACML policy editing. Argus integration may~~ be~~ also be considered if grid deployments (~~to whatever,~~because of e.g. legacy reasons) prefer to keep attribute sources very simple.

Concerning incarnation: attribute source services (UVOS/XUUDB/or even a file) define permitted and default values for users/groups of users etc. within UNICORE. As in the case of e.g. D-Grid the input and definition files for these attribute source services can be created in a more global way. Additionally a local configuration file is used for application related data. Users can express preferences to choose desired values (e.g. a desired GID) out of possible ones. Additionally the local administrator can define hooks which modify the incarnation.

So even if the current user management already fulfils our basic requirements it will be useful to integrate UNICORE with an EGI wide supported user authorization system for the sake of unified access or in scenarios where different middlewares are deployed on one site.

To integrate Argus with UNICORE there are three different integration options to be discussed:

a) Usage of Argus PDP. UNICORE can be configured not to use the local XACML file as in the default implementation, but to contact Argus PDP instead. The Argus PEP component is intentionally skipped as it is spurious, slower and using a proprietary protocol. The obvious drawbacks with this approach is that a web service call has to be made for each request which is quite slow, and in the case that Argus is unreachable (down, network overload,..) the relying services would be down. Furthermore the current implementation of the SPL is still to simple to express a default UNICORE policy, but this will be fixed soon in a later release of Argus. ~~Another problem with this approach is that it would be quite difficult to have it in a hierarchical setup since it is not well defined/configurable how the policies are merged.~~

b) Therefore the currently preferred option is to use Argus PAP directly. The according prototype is nearly ready, and will be finished until the end of September. Policies are fetched from the Argus PAP and evaluated locally. This is fast and Argus fault-tolerant. Additionally it will be implemented in such a way that the policy from Argus which is very good for expressing~~at~~ banning statements will be integrated as a part of the default policy to avoid any integration problems. For this implementation an extension of the Argus SPL is not needed.

c) A third integration possibility (independent from the two previous ones) would be to use Argus EES for ~~integration~~incarnation. To do so a refactoring of the UNICORE container would be needed. This ~~possibility~~feature ~~to contact Argus EES~~ is only in the planning state ~~for a future~~ (clearly beyond this years development plan), so for now only the UNICORE native incarnation is possible.

2.8.3.4 Argus and Globus

Globus still all relies on the entries in the Globus grid-mapfile for authorization purposes.

VOMS of VOMRS can be used to provide the necessary entries in order to achieve a high-level VO management for Globus.

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

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

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

Feature planned before the end of the year:

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

On the todo list:

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

3 INTEROPERATION AT PROCEDURES AND POLICY LEVEL

3.1 *Scope*

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

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

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

3.2 *Current EGI Procedures and Policies*

An overview of the procedures, policies and best practices inherited and already improved, changes needed can be found in MS408, resp. MS415 [**R** ~~10~~]. The milestone lists procedures taken over from EGEE, newer procedures ~~already~~ passed through OMB and in effect and procedures in various draft stages. The current valid procedures are collected in the wiki [**https://wiki.egi.eu/wiki/Operational_Procedures**] and regularly updated and discussed in the OMB. Operational Manuals including prescriptive guidelines on e.g. downtime handling, Best Practices and related howtos are now also found in the wiki [https://wiki.egi.eu/wiki/Operations_Manuals, https://wiki.egi.eu/wiki/Operations_Best_Practices].

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

Security procedures are handled separately.

Potentially new players when adding new resource types have to be aware and follow the policies and procedures published by the Security Policy Group (SPG), [R 41], namely the Security~~Grid~~ Incident Response Policy [https://documents.egi.eu/public/ShowDocument?docid=82] ~~referenced in MS405 [R 1]~~ and the operational security procedures published in MS412 [R 1], especially the security incident procedure and the software vulnerability issue handling process collected therein.

In the deployed EGI infrastructure all problems concerning security should be dealt with between the

EGI Computer Security Incident Response Team (CSIRT) and the EGI Software Vulnerability Group (SVG), [R 41]. CSIRT advises the ~~sites~~resource centres on security matters and has the power to suspend them~~sites~~ from the infrastructure if they fail to apply critical security patches.

The EGI Incident Response Task Force (IRTF) makes sure that incidents are handled according to the Incident Response Procedure. The SVG should ensure that the software available for installation on the EGI infrastructure is sufficiently secure and contains as few vulnerabilities as possible, thus reducing the likelihood of incidents.

When introducing a resource of a new software provider the contact details of the support of that software provider have to be pointed out to the SVG. Those new software providers should become part of EGI UMD and sign the corresponding SLA [R 17]. All these issues are covered in detail in the software vulnerability issue handling process part of MS412~~05~~.

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

3.3 Future Procedures

The operational procedures used within EGI have been evolved and further developed from those established~~developed~~ within the EGEE series of projects and now have broad community support and adoption. ~~For example~~Correspondingly, all current procedures and related operational work flows are directly reflected within the Operations Portal. As a result, the portal has to be regularly updated as the procedures change to reflect the needs of the community. Input on non- or only partially working procedures, together with suggestions for improvements, are identified from many sources.

One important source of feedback comes from the review of the operational activities that takes place during the weekly handover between operational support teams who use these procedures on a daily basis through their implementation in the operational tools such as the Operations Portal. The collected best practices within the community represents a further source of feedback.

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

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

Certain focus will have to be laid on the more technically tied procedures, like many of the ones in the availability and monitoring area or the Resource Centre Registration and Certification Procedure [ref https://wiki.egi.eu/wiki/PROC09] to see to it that they equally to the Resource Centre OLA [https://documents.egi.eu/document/31] are totally middleware agnostic while at the same time staying specific and useful. For that purpose the middleware specific howtos which are still part of some procedures will have to be split from the actual procedures.

Also externally developed procedures and policies might be relevant for newly integrated Resource Centres. E.g. the Availability/Reliability reports are provided to EGI by the ACE team which is an WLCG effort and WLCG recently defined an availability/reliability recomputation policy/procedure https://tomtools.cern.ch/confluence/display/SAM/Availability+Re-computation+Policy .

4 OUTLOOK AND FUTURE PLANS

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

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

Recently we have made good experiences with setting up dedicated integration taskforces that bring all interested parties and otherwise involved players together and will continue with that approach whenever we see a critical mass of interest for a specific requirement coming from different partners.

Should mention also UMD release schedule here somewhere https://documents.egi.eu/public/ShowDocument?docid=526

4.1 *Operational requirements coming from our integration taskforces*

NEW!

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

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

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

4.1.1 UNICORE integration taskforce

The UNICORE integration taskforce started its work in February 2011.

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

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

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

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

4.1.2 Globus integration taskforce

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

(refs, as above)

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

4.2 *Operational requirements coming from NGIs*

ipv6 compatibility?

4.2.1 Integration of desktop Grids

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

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

4.2.2 Integration of cloud services

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

However, the deployment of virtualized resources requires extensions to the current monitoring and accounting functionality. Various use cases have been identified and discussed during the "EGI Production Infrastructure" session of the first EGI Technical Forum.

These need to be further refined to define a common tool development roadmap.

4.2.3 Integration of new resources into accounting

New repository can handle multiple record types.

This is required for schema evolution in CPU

But allows multiple types of resource to be accounted.

MS706 – explains

Potential new resources:

Storage

EMI defined StAR

Clouds (See EGI Virtualisation and Clouds Workshop output)

Applications

CPU

Local Jobs

MPI

Data Use – remote access to data, by user, not owner

Services? Suggestion that ALL services be accounted.

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

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

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

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

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

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

4.3 Operational requirements coming from Collaborations with other DCIs

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

Currently, the percentage of users that is interested in capacity as well as capability computing at the same time is rather low. However, regardless of this, infrastructure providers need to support users in directing them to the infrastructure that suits their use case best, and need to reduce the number of barriers user may experience, so that shifting from one infrastructure to the other should be a more smooth and transparent process. This can happen through the deployment of common top-level tools, support systems and procedures.

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

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

Several topics of common interest were identified:

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

5 REFERENCES

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