EGI-InSPIRE
Operational Tools Sustainability

Document Link	https://documents.egi.eu/document/1843
Last Modified	10/04/2013
Version	1.0
Contact Person	Diego Scardaci (INFN)
Document Status	

Abstract

This document deals with the sustainability of EGI Operational Tools after EGI-InSPIRE.



I. COPYRIGHT NOTICE

II. AUTHORS LIST

	Name	Partner/Activity/ Organisation/Function	Date
From	Diego Scardaci	INFN	10/04/2013

III. DOCUMENT LOG

Version	Date	Comment	Author/Organization
1.0	10/04/2013	**First Version**	**Diego Scardaci (INFN)**
2.0			

TABLE OF CONTENTS

INTRODUCTION

This document deals with the sustainability of EGI Operational Tools after EGI-InSPIRE.

For each tool the following information have been collected:

- technical description of the system;
- information about the amount of support effort needed after EGI-InSPIRE through GGUS (to users and operators);
- the effort needed to gather new requirements from OMB and UCB of EGI, analyse these technical requirements and prioritize them, define your yearly technical roadmap (number of releases foreseen, new functionality expected, etc.);
- the development effort needed to maintain the software:
    - the resources needed for the daily running of the system, this includes cost of hardware and licenses;
    - the effort needed for bug fixing, proactive maintenance, improvement of the system, code refactoring. This effort does not include the development of new features;
    - an estimation of the manpower needed to maintain the tool splitted in two sub-activities: running the system and testing.

The cost breakdown table in Section 9 summarizes the data collected.

1 OPERATIONS PORTAL

1.1 Technical Description of the system

The architecture is composed of three modules:

- A database – to store information related to the users or the VO - namely MySQL
- A web module – graphical user interface – which is currently integrated into the Symfony and bootstrap frameworks
- A Data Aggregation and Unification Service named Lavoisier

Lavoisier is the component used to store, consolidate and "feed" data into the web application. This module provides information from various sources without the portal being directly dependent on those information sources thanks to a caching mechanism. This indeed protects us from intermittent failures of information sources.

This portal has been conceived and built as an integration platform of different and heterogeneous sources of information. Lavoisier is used to integrate and harmonized these different data sources.

On the schema in Figure 1 you can see on the right the components of the portal and on the left the different external sources used in the portal.

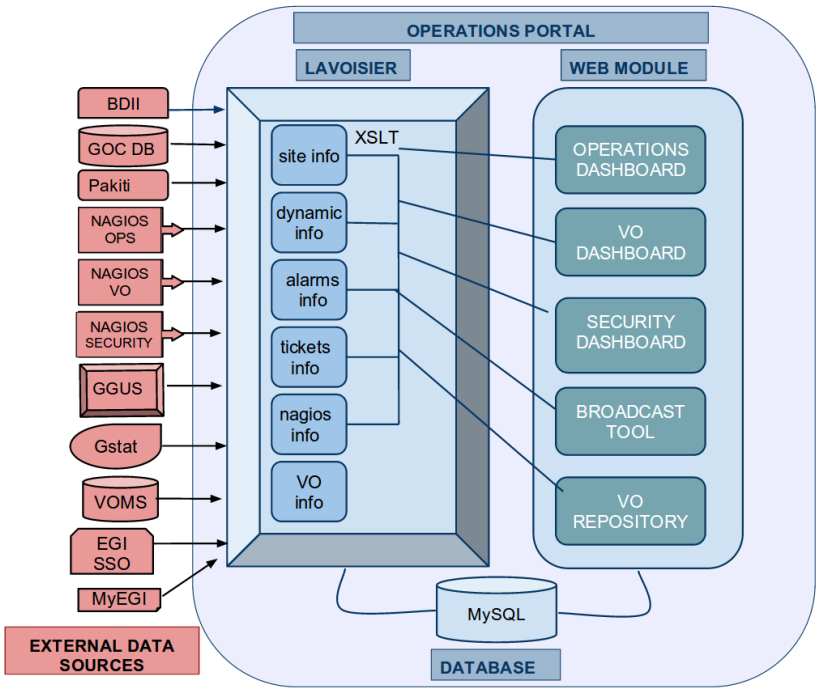

Figure 1 - Operations Portal Architecture

The different components are integrated in a high available mode:

- the Mysql Database is integrated in a cluster service;
- the web module is also integrated in a cluster service;
- the configuration of Lavoisier is stored in a subversion repository and the service is easily deployable on the fly in case of failure;
- Different instances of the database, web module and Lavoisier are deployed.

The Operations Portal is proposing a set of different tools that have evolved during the EGI project following the needs of the different users of the evolution of the procedures and the operations.

Currently the main features are:

1. The detection and the follow-up of incidents on the different sites of the EGI infrastructure trough synoptics views; The different interfaces summarize the different information related to the sites especially the different monitoring statuses and allows to open trouble tickets. Four different dashboards have been developed:
   - the historical one used for the daily operations on the different NGI;
   - a central dashboard to detect potential problems in the NGI operations;
   - one dedicated to detect and follow security incidents and vulnerabilities;
   - one oriented for the VO Operations.
2. The Operations portal is the official repository for the static information related to Virtual Organisations with information such as:
   - The VO contact points (VO managers, VO User mailing list, VO representatives, etc.);
   - The VO global information (enrolment URL, status, discipline, etc.);
   - The Acceptable User Policy of the VO;
   - The VO's Core Services;
   - The VOMS information (Groups and roles, certificate details, etc.);
   - Any other specified requirements (CPU needed, RAM needed, etc.).

   Different interfaces allow user to declare a new VO into the EGI infrastructure. The different filled information are checked and the Operation team is able through a specific module to manage the new registrations, the updates.
3. The broadcast tool is one of the most used feature: with this tool every authenticated user are able to contact several categories of stakeholders impacted by a problem, an announcement, or a specific release. The aim is to share with the different actors of the EGI community some useful information. In addition to sending out information we provide also:
   - an archiving service;
   - a search engine to retrieve the different entries in the archive system related to the search criteria (author , subject …);
   - the possibility to add customized contacts;
   - the possibility to use templates.
4. A visualisation (charts) and notification (emails or rss) system related to the downtimes impacting the services, the sites, the NGIs or the VO;

5. A reporting and computing system giving the availabilities and reliabilities of the TOP-BDII services, of the sites and of the services of a VO;
6. A user tracking tool which allows an end user to contact another user without having their email address, through his DN certificate;
7. Metrics and charts :
   - about the distribution of the users per VO, per discipline;
   - about the distribution of users;
   - about Operational activities.

In complement to this description here are some indicators of the activity of the portal :

Month	N. of different visitors	Visits	Pages	Hits
Jan 2013	2 268	8 167	52 327	307 834

Figure 2 - Operations Portal indicators

- Around 10 releases per year;
- 15 different data providers;
- 2450 tickets opened from the dashboards since September 2012;
- 140 Tickets opened against the Operations Portal in 2012.

1.2 Support

In Table 1 the effort needed for support activities in 1 year.

SUPPORT	
Central Operations Portal Support – daily maintenance • Regular Dashboard • COD dashboard • Security Dashboard • VO Operations Dashboard • Broadcast • Downtime Notification • VO Information Module • VO management Module • A/R Module • User tracking • Metrics / Charts	3 PM
Central Post-Release support : 10 releases/year	1 PM

Third parties disruption / evolution support from : • GOCDB • GGUS • My EGI • AMQ: VO / Security / Ops Nagios Boxes • Gstat • VOMS server • Pakiti • Bdii • EGI SSO	1.5 PM
TOTAL SUPPORT	**5.5 PM / YEAR**

Table 1 - Effort for support

1.3 Requirements gathering, technical road mapping, communication

In Table 2 the effort details.

Requirements gathering, technical roadmapping, communication	
Evaluation within the projects for requirements	1 PM
Milestones, activity reports , presentations	2 PM
Coordination / Internal Meetings	2 PM
TOTAL	**5 PM**

Table 2 - Effort for requirements gathering, technical roadmapping, communication

1.4 DevOps

1.4.1 Capital expenditure, licenses

In Table 3 HW and SW costs for 1 year.

Capital expenditure, licenses	
Web and Database Servers	120 Euros / year
Lavoisier : 2 Machines (1 physical server + 1 VMware)	Electricity : 700 Euros / year Cost of the machines : 4400 / 5 years = 880 / year
Licences : JetBrain PhpStorm + Fusion Chart	100 + 200 = 300 Euros / year
	2000 Euros / year

Table 3 - HW and SW costs for 1 year

1.4.2 Code refactoring, proactive maintenance

In Table 4 effort for code refactoring and proactive maintenance.

Code refactoring, proactive maintenance	
Ergonomics, backend and automation of procedures	1 PM
Technology upgrades : php / symfony / boostrap versions	2 PM
Code refactoring	1.5 PM
Pro active maintenance	0.5 PM
Bug Fixes • Regular Dashboard • COD dashboard • Security Dashboard • VO Operations Dashboard • Broadcast • Downtime Notification • VO Information Module • VO management Module • User tracking • Metrics / Charts • Lavoisier Web Service • MySQL Database	2 PM
	7 PM

Table 4 - Effort for code refactoring and proactive maintenance

1.4.3 Manpower for running the system, testing, high availability

In Table 5 manpower for running the system, testing, high availability.

Manpower for running the system, testing, high availability	
High availability effort Integration – MySQL Cluster – Web Cluster, – configuration files stored in SVN – Lavoisier duplications – Hardware install, maintenance and upgrade – Tests instances	1.5 PM
Efforts to run the different systems : – prod / pre-prod / dev Web Instances – 5 lavoisier instances – Mysql Databases : prod and test	1.5 PM
	3 PM

Table 5 - Manpower for running the system, testing, high availability

2 GGUS

2.1 Technical Description of the system

The GGUS system is divided into three environments: development, test and production environment. Every environment includes three layers:

- Presentation - web frontend to provide the entry point for the graphical user interface;
- Logic - AR Server which executes the workflow rules and performs the main tasks. AR Server is providing the communication interface between external systems and is accompanied by the email-engine to provide the additional mail-based interface into the helpdesk system;
- Backend - ticket database under Oracle DBMS and user database under MySql DBMS (will be moved to Oracle soon).

Environment / Component	Development	Test	Production
Web frontend 1	dev-1.ggus.eu	train-1.ggus.eu	prod-5.ggus.eu
Web frontend 2	dev-2.ggus.eu	train-2.ggus.eu	prod-3.ggus.eu
User database(MySql)	mysql-mgm,ggus.eu	mysql-mgm,ggus.eu	mail.ggus.eu
Logic 1	dev-ars-1.ggus.eu	train-ars-1.ggus.eu	prod-ars1.ggus.eu
Logic 2	dev-ars-2.ggus.eu	train-ars-2.ggus.eu	prod-ars2.ggus.eu
Data 1 (Oracle)	Ora1	Ora1	Ora1
Data 2 (Oracle)	Ora2	Ora2	Ora2

Table 6 - GGUS Environments

Each layer includes two servers. In case of presentation and logic one acts as an active server and the other as stand-by which can be activated in case of incident. In case of backend – the Oracle Real Application Cluster is maintained with two active server stacks.

GGUS is being monitored by ICINGA and integrated into on-call duty service. In case of a service incident the on call engineer (OCE) will fix the problem according to instructions described in on-call duty service wiki. In case the OCE does not succeed to fix the problem, the GGUS expert can be called.

2.2 Support

GGUS receives continuously tickets against the GGUS support unit itself and contact emails via the contact form at the website or via email to support@ggus.eu.

The majority of tickets deal with bugs and incidents regarding the GGUS system whereas the contact mails were mostly related to user accounts and general questions. The amount of work to follow up of these two communication ways takes around a 2 PM/Y.

2.3 Requirements gathering, technical roadmapping, communication

Requirements for GGUS coming from EGI are recorded in three different queues of the EGI RT:

- Dashboard GGUS-Requirements;
- GGUS AB;
- InSpire-JRA1.

The requirements in these queues are being discussed in weekly meetings with the GGUS developers and once per month in the GGUS Advisory Board. The items are being prioritized and assigned into releases. During the last years GGUS used to have one release per month, the usual release date is the last Wednesday per month.

Major tasks in the near future will be the development and integration of alarm processes for the EGI central operations tools, alternative authentication methods to secure access GGUS without X.509 certificates and the completion of the high availability solution for the GGUS architecture.

2.4 DEVOPS

2.4.1 Capital expenditure, licenses

The cost of hardware and licenses per year are:

- Amortisation costs for the servers: 3000 €;
- License costs for ARS (BMC Remedy Action Request System): 13500 €;
- Power/Electricity: 1500€;

2.4.2 Code refactoring, proactive maintenance

The effort needed for bug fixing, proactive maintenance, improvement of the system, code refactoring is 4.5 PM/Y.

2.4.3 Manpower for running the system, testing, high availability etc.

The manpower needed to maintain the tool divided into subtasks:

1. Running the system: 2 PM/Y
2. Testing: 1 PM/Y
3. System administration: 4 PM/Y

The GGUS team foresees a 50% reduction of the work-load after EGI-InSPIRE assuming that GGUS has reached a fully developed and stable status at the end of the project. The effort values inserted in the spreadsheet are been defined taking in consideration this work-load reduction.

3 GOCDB

3.1 Technical description of the system

For the GOCDB service, this includes three virtual machines (VMs) hosted in the STFC production VM infrastructure (gocdb-base, gocdb-test and godb-failover that is currently being installed at Daresbury Labs). Each VM hosts: Enterprise RedHat OS, Apache2 webserver, PHP5, nagios/ganglia monitoring components for our in-house production monitoring. Gocdb-base is UPS enabled.

The GOCB databases are hosted by the STFC DB-Services group on the production Oracle cluster. This includes nightly DB back-ups to the STFC tape storage facility and UPS support. The spreadsheet quotes a zero figure because the Oracle license and DB production infrastructure is currently **absorbed** by STFC who will host the Oracle cluster and service regardless - this cost is not currently passed on to EGI. While GOCDB has an Oracle dependency, this would add an extra ~£2000/yr to the total (this is a very tentative figure, the DB services group host production databases for a wide variety of projects/users and this figure is to be regarded as a loose estimate at best). However, since v5 supports different open source DBs, we can potentially remove the Oracle license costs in near future.

3.2 Support

Total GGUS support currently requires approximately 1 PMs/yr. We expect this to remain largely static for PY4. This includes v5 release and cover for post v5 new developments (Glue2 XML, multiple endpoints, scoping extensions, extensibility mechanism). Post PY4, we envisage total support may potentially drop to approx. 0.5 PMs/yr (assuming stabilization of new features). We also recommend the support is continued to be split between at least two staff members in order to provide continuous operational support and cover, e.g. considering holidays/sick-leave.

3.3 Requirements gathering, technical roadmapping, communication

Total is 1PM/yr to cover requirements gathering and technical road-mapping for PY4. This does not include PMs for actual development and implementation (4~5PM/yr). We expect ~3 releases up the end of PY4 to put v5 and post v5 new features into production.

3.4 DEVOPS

3.4.1 Capital expenditure, licenses

Capital expenditure figure is for 3 host installs in our production VM infrastructure by paid STFC/gridpp (gocdb-base, gocdb-test, and new gocdb-failover that is being installed at Daresbury Labs):

- 3 OS Redhat licenses: £ 105
- VMware inc Hardware, power: £ 789,2
- VMware Storage Costs: £ 500//TB/year
- SCT machine management Infrastructure (Nagios/Gangia/deployment, etc): £ 300
- Basic System Admin: £ 1200

- Database admin: £ 0[1] [2]

Total £ 2894/ Year = 3429 Euro / Year.

3.4.2 Code refactoring, proactive maintenance

3 PM / Year.

3.4.3 Manpower for running the system, testing, high availability etc.

2 PM / Year.

[1] Is zero because the Oracle license and DB production infrastructure is currently absorbed by STFC who will host the Oracle cluster and service regardless - this cost is not currently passed onto EGI. While GOCDB has an Oracle dependency, this would add an extra ~£2000/yr to the total.

[2] since v5 supports different open source DBs, we can potentially remove the Oracle dependency in near future.

4 SAM

4.1 Technical description of the system

Service Availability Monitoring (SAM) is a monitoring infrastructure framework supporting WLCG and EGI/NGI operations. SAM is a key operational tool that supports rich functionality including remote monitoring of services, visualization of the service status, dashboard interfacing, notification system and generation of availability and reliability reports. SAM is currently used by all the NGIs participating in the EGI project.

SAM is a system made up of several components, some commodity and some specifically designed and developed for SAM. These include Nagios to execute tests, Messaging to transport test results between components, databases to store both configuration information: the Aggregate Topology Provider (ATP) and the Profile Management Database (POEM), databases to store the test results produced by Nagios: the Metric Result Store (MRS). Other components such as the Availability Calculation Engine (ACE) processes the raw test results to calculation metrics such as site and service availability and reliability. A portal, MyEGI is provided to visualize both test results and availability calculations.

The execution of the SAM tests is done through a set of 33 distributed Nagios instances around the world. These are independent instances capable of receiving the test results back (thanks to the distributed EGI Messaging Infrastructure), store these results in their local MySQL databases, compute status of services, and present the results through their local MyEGI web interfaces.

SAM is also used to monitor all the EGI Operational Tools. This is done through a specific SAM instance called OPS Monitor (https://ops-monitor.cern.ch/nagios/) that also runs Nagios, MySQL and the different SAM components required to compute the status of these services and present the results in its MyEGI instance (http://ops-monitor.cern.ch/myegi/).

The aggregation of all EGI metric results is done in the SAM central system at CERN. Thanks to the EGI Messaging Infrastructure all metric results are collected and stored in an Oracle database. From there, status and availability of services and sites are computed. These results are exposed through the SAM central MyEGI web service and its programmatic interface (XML & JSON supported). On top of that, the SAM Reporting System generates monthly availability reports about sites and operational tools for use of the EGI management.

4.2 Support

The support effort expected for SAM after EGI-InSPIRE through GGUS is around 6 PM/year. This is mainly required for the operational support of the 33 distributed Nagios instances and to analyse and perform availability re-computation requests.

4.3 Requirements gathering, technical roadmapping, communication

The expected effort needed to gather new requirements from OMB and UCB of EGI, analyse technical requirements, and prioritize them, attend meetings, report to the different bodies, define the SAM yearly technical roadmap and follow up each release (development, validation and deployment to production) to guarantee the execution of the planned roadmap is of around 12 PM/year.

4.4 DEVOPS

4.4.1 Capital expenditure, licenses

Effort/resources needed for the daily running of the SAM system is estimated in around 81 EUR/day. This includes only cost of hardware but not the cost of personnel providing those services (hardware, OS, alarms, Computing Center operators, etc.) to the SAM team, as this is accounted under column F (effort for sys-administration and testing) in the cost breakdown table (see section 9).

For the cost of hardware, we consider that 18 boxes are required to provide the same level of QoS that we have today:

- 6 production instances: 1 Central (grid-monitoring), 1 OPS-Monitor, 4 VO Nagios
- 6 pre-production instances: 1 Central (grid-monitoring), 1 OPS-Monitor, 4 VO Nagios
- 6 nightly validation boxes: 2 Central, 2 NGI Nagios, 1 OPS-Monitor, and 1 VO Nagios.

4.4.2 Code refactoring, proactive maintenance

As documented in the cost breakdown table, the estimated effort needed (after EGI-InSPIRE) for bug fixing, proactive maintenance, improvement of the system, and code refactoring for SAM is of 30 PM/year. This effort does not include the development of new features.

4.4.3 Manpower for running the system, testing, high availability etc.

As documented in the cost breakdown table, the estimated effort needed to operate and test SAM is the following:

- Operation of SAM: 10 PM/year
- Testing of SAM: 4 PM/year

5 MESSAGE BROKERS

5.1 Technical description of the system

The EGI Production Messaging Infrastructure (PROD MSG Network) consists of 4 ActiveMQ broker endpoints located in Greece (AUTH - 1), Croatia (SRCE - 1) and CERN (2). The production broker network serves as a backend infrastructure for EGI operational tools that need to use a message broking functionality (i.e. SAM infrastructure, APEL).

In parallel to the production network a test network using a similar setup has been operational. The test network consists of 3 ActiveMQ broker instances located in Greece (2) and Croatia (1) and it is used for testing purposes mostly prior to applying major updates or configuration changes on the production network. In addition to this network two additional brokers known as "dev" and "pilot" are running at CERN. These are used daily by EGI (for instance SAM is using one for its validation).

5.2 Support

Support regarding the production message broker network is provided via GGUS (Messaging Support Unit). Considering the current development roadmap and the fact that more and more operational tools rely upon the production broker network we foresee that the amount of support effort required after EGI-InSPIRE for support activities will be 3PMs/year.

5.3 Requirements gathering, technical roadmapping, communication

Due to the distributed nature of the broker infrastructure strong collaboration among the different teams running the broker instances has been at place since the beginning of the operations. An average of 3-4 updates on a yearly basis have been performed and several communications are needed both prior to the updates (for scheduling downtimes and notification purposes) and during the updating procedures among the teams for coordination. In addition and at least twice per year coordination meetings among the teams take place in order to follow up on the requirements gathering and technical roadmapping procedures.

After the end of EGI-InSPIRE we foresee that an effort of 1.8PM/year evenly split among the teams running the production and test broker networks (AUTH, CERN and SRCE) will be required to cover these operations.

5.4 DEVOPS

5.4.1 Capital expenditure, licenses

No licensing is required at the moment for ActiveMQ software. The license of ActiveMQ release running on the production and testing broker instances is currently Apache License 2.0.

At least once every 3 years the hardware infrastructure upon which the broker endpoints reside should be replaced. The estimated costs are 17,000 Euro per year (5,000 Euros accounted for AUTH brokers, 5,000 Euros for SRCE brokers and 7,000 Euros for CERN brokers).

5.4.2 Code refactoring, proactive maintenance

Several bugs in the operation of ActiveMQ have been already identified and tackled with to a satisfactory degree. Based on experience, bug hunting/fixing in ActiveMQ is very costly. In 2012,

around 40 bugs have been submitted via CERN support contract plus a few more via apache.org's JIRA. Each bug requires resources to reproduce it, isolate it, and work on it with the support staff. In order to keep up with the proactive maintenance of the broker endpoints and the further improvement of the service an effort of 0.5PM/year for AUTH and 2.5PM/year for CERN will be required. Total: 3PM/year.

5.4.3 Manpower for running the system, testing, high availability etc.

The effort foreseen for daily administrative operations (i.e. checking the healthiness of a broker network endpoint) is 0.8PM annually per team (AUTH, CERN, SRCE). Total: 2.4PM/year.

As described in the first section, testing is performed on the test broker network. Testing is required in cases prior to applying software updates or configuration changes on the production broker network. The effort required for testing is 0.2PM annually per team (AUTH, SRCE). Total: 0.4PM/year.

6 ACCOUNTING REPOSITORY

6.1 Technical description of the system

The Accounting repositories store CPU, cloud and storage accounting data collected from sites participating in the EGI and WLCG infrastructures. The accounting information is gathered from different sensors in to central accounting databases where it is processed to generate statistical summaries that are available through the EGI/WLCG Accounting Portal. The CPU Accounting repository (APEL) has been re-developed and there are currently old and new versions of the repository in production, their data combined is sent on to the Accounting Portal. The old data will have been migrated by the end of EGI-Inspire but we expect one component of the old APEL Server system to still be in place (the APEL broker and consumer servers) as the previous version of the APEL client will still be in use at some sites and these servers will handle that data until all old versions of the APEL client have migrated to the new EMI APEL 3 client. The other types of accounting supported all use the same system components as the new APEL server. The client systems for storage and cloud accounting are not written or supported by STFC.

System Components:

Old APEL Server broker and consumer:

- VMs running Scientific Linux
- Java
- ActiveMQ
- MySQL

New Accounting systems:

- Mix of VMs and physical hardware running Scientific Linux
- Python
- MySQL
- Apache

All APEL servers use internal STFC backup, Nagios and ganglia monitoring services etc.

6.2 Support

Support required for the APEL client users, APEL Regional Server users, Cloud Accounting, Storage Accounting and Application Accounting. APEL Regional Server users, Cloud Accounting, Storage Accounting and Application Accounting are all new services that will be delivered during PY4. Estimate 6PM/Year.

6.3 Requirements gathering, technical roadmapping, communication

Estimate 1 PM/Year

6.4 DEVOPS

6.4.1 Capital expenditure, licenses

Daily running of the system – some servers are physical some virtual so costs include:

- Power

- VMware inc. Hardware
- VMware Storage Costs
- Physical Hardware
- Physical Power
- Physical Space
- Management Infrastructure (Nagios/Gangia/deployment etc.), backups.

6.4.2 Code refactoring, proactive maintenance

Estimate needed (after EGI-InSPIRE) for bug fixing, proactive maintenance, improvement of the system, code refactoring is 3 PM/Year. This effort does not include the development of new features.

6.4.3 Staff required for running the system, testing, high availability etc.

Staff needed to maintain the tool split into sub-activities, running the system and testing is 6 PM/Year.

7 ACCOUNTING PORTAL

7.1 Technical description of the system

The Accounting Portal aggregates and provides access to the CPU and user registers gathered by APEL on the infrastructure. The Portal is developed and maintained integrally by CESGA staff.

The production and development instances are hosted at CESGA. The portal has a frontend that renders HTML pages, and a Python backend to update the DBs. The frontend is coded on PHP 5 running under Apache, with a MySQL relational database on a SL6 distro. It uses plain PHP and pChart 2.0 for the graphs.

The databases are separated into a CPU record database, a User record database, and a topology database.

7.2 Support

The Accounting Portal has evolved from a heavy legacy base, dating from before the EGEE projects, and has been improved by several refactoring, and a change from PHP 4 to 5. In the last year of EGI, several JRA1.4 related views will be implemented, including Storage, MPI and Application accounting. There is already a new cloud view that will be showcased on the EGI CF 2013. These new views will need very serious support and evolution to match the existing legacy functionality, that is refined after many years.

The regional instances of the Portal will also add to the support load, even if few NGIs have interest, there will be the need to support these regional instances and their added functionality.

There is also a need to document better the system both for users and maintainers, a front that needs attention, but has been delayed due to very needed refactoring work.

7.3 Requirements gathering, technical roadmapping, communication

The Portal has requirements from Site admins, VO Managers, VO Members, Operation Officers, Project Administration, Fedcloud, almost all of the operational tools, and several VTs (MPI, egi-usage, scientific-classification). RT is used for requirements and GGUS for bugs and downtimes.

Personal emails are used sometimes, but there is a great load prioritizing requirements and implementing changes, so formal tools are needed. There is also the need to avoid introducing new bugs, the perhaps fragile nature of the legacy elements of the Portal means that changes need to be closely monitored.

As is, there is a lot of manpower expended on this front, and it could not possibly be reduced any further.

7.4 DEVOPS

7.4.1 Capital expenditure, licenses

We have included on the cost breakdown table the electricity and regular maintenance costs, it represents a very affordable expenditure annually. We use free software that is very well supported (PHP and MySQL), so there are no expenses on licenses.

7.4.2 Code refactoring, proactive maintenance

The codebase has many legacy components, including subtle changes and fixes made along the years and that are usually not documented. The inclusion of new functionality has forced some refactors, as the view consolidation, change to HTML+CSS instead of frames and currently, a rewrite of the core. These refactors constitute a great part of the development time, and need to be juggled with maintenance and bug fixing. The use of code versioning has been invaluable for this, and the result was that many changes were possible in a practical time.

The Portal also gathers information from many sources, so constant changes to these interfaces need to be made, we currently don't have any caching solution to avoid downtimes, and this means that fixes need to be delivered fast to avoid loss of functionality.

7.4.3 Manpower for running the system, testing, high availability etc.

Currently the system is maintained by a minimal workforce, with sporadic management support for updating packages and virtual machine maintenance. This arrangement also maintains the Metrics Portal. As such, the manpower for the tool could not be further reduced without loss of functionality.

8 METRICS PORTAL

8.1 Technical description of the system

The Metrics Portal eases the tracking and management of the EGI project by offering a centralized interface to enter and visualize project metrics and create reports with them. It gathers information from BDII and other sources to estimate infrastructure metrics, and shows reports per Quarter, NGI or metric. The Portal is developed, maintained and hosted by CESGA

The portal has a frontend to generate HTML+CSS pages:

1. Based on Python and Django, with ORM access to a MySQL database.
2. The site form are created on-the-fly from the schema.
3. The schema is migrated to the database using South.
4. Supports SSO and certificate authentication.
5. History of modifications, wiki-like editing.
6. There is also a backend to gather data regularly for the metrics:
   - Done in Python, uses Django ORM to access DB.
   - Called daily by cron, the data is aggregated per NGI.
   - The script walks the BDII tree to aggregate data like disk usage and CPU capabilities along all the nodes.

8.2 Support

The metrics portal should improve its reporting, and eventually feature trend analysis, and cause reporting. There is a minimal core of this in the current use of the Comment field. It should also offer Geo and temporal data graphing. In the long future, metrics could be user-defined by composing primitive values, and offer retroactive trending

The Portal would need a continuing support in order to accurately determine infrastructure related metrics, any changes on the BDII schema or the adoption of a new Information System would disrupt the gathering of those. Currently there are many metrics that are only gathered by the Portal and there would be a loss of history of the infrastructure evolution.

There is an overlap in some functionalities with the Accounting Portal, notably on the consumed hours metrics and some of the topology ones, but with lesser support than current on this tool, there would be a lot of historic data that would probably be lost.

8.3 Requirements gathering, technical roadmapping, communication

The current instruments, including RT and GGUS, and emails with EGI officers to coordinate metric gathering are enough, and the changes are introduced timely. After the end of the project, the mechanisms are unclear, but this part of the development process has generated little problem and can be put on a second plane.

8.4 DEVOPS

8.4.1 Capital expenditure, licenses

We have included on the cost breakdown table the electricity and regular maintenance costs, it represents a very affordable expenditure annually. We use free software that is very well supported (Python, Django, MySQL), so there are no expenses on licenses.

8.4.2 Code refactoring, proactive maintenance

The codebase was made from scratch with Python and Django and with no legacy components, and was designed using best practices and with full knowledge of the changing requirements it would be subjected, so it is extremely flexible, for the moment it does not need significant refactoring, but this could change if the ultimate purpose of the tool changed. At any rate, this can be subsumed with the cost in running the system.

8.4.3 Manpower for running the system, testing, high availability etc.

Currently the system is maintained by a minimal workforce, with sporadic management support for updating packages and virtual machine maintenance. This arrangement also maintains the Accounting Portal. As such, the manpower for the tool could not be further reduced without loss of functionality. The data gathered with this tool is not stored elsewhere, as with the Accounting Portal, so it could not be retrieved if there was an interruption of this task.

9 COSTS BREAKDOWN

Tool	Support Effort through GGUS (Person Month/Year)	Effort to manage requirements (Person Month/Year)	Effort for the daily running of your system (KEuro/Year)	Effort for bug fixing, proactive maintenance, code refactoring (Person Month/Year)	Effort for Sys-Administration and Testing (Person Month/Year)	Total (Costs for Year)
Operations Portal	5,5	5,0	2	7,0	3	**20,5 PM + 2 KE**
GGUS	2	4,5	18	4,5	7	**18 PM + 18 KE**
GOCDB	1,0	1,0	3,429	3,0	2	**7 PM + 3,4 KE**
SAM	6	12	30	30	14	**62 PM + 30 KE**
Message Brokers	3	1,8	17	3	2,8	**10,6 PM + 17 KE**
Accounting Repository	6	1	1,15	3	6	**16 PM + 1,15 KE**
Accounting Portal	3	3	0,85	4	3	**13 PM + 0,85 KE**
Metrics Portal	0,5	1	0,85	1	0,5	**3 PM + 0,85 KE**