

**EGI-Engage**

Technical design of the new Accounting Portal and implementation plan

D3.1

|  |  |
| --- | --- |
| **Date** | 10 August 2015 |
| **Activity** | WP3 |
| **Lead Partner** | CESGA |
| **Document Status** | DRAFT |
| **Document Link** | https://documents.egi.eu/document/2545 |

Abstract

The EGI Accounting Portal is an operational tool that processes, summarizes and displays the Accounting Repository data, acting as a common interface to the different accounting record providers and presenting a homogeneous view of the data gathered and a user-friendly access to understanding resource utilization covering the needs of many types of actors.

This document details the technical design of a new version of the Accounting Portal that will enhance its interface, look and feel and introduce new features in order to satisfy emerging user requirements.

**COPYRIGHT NOTICE**



This work by Parties of the EGI-Engage Consortium is licensed under a Creative Commons Attribution 4.0 International License (http://creativecommons.org/licenses/by/4.0/). The EGI-Engage project is co-funded by the European Union Horizon 2020 programme under grant number 654142.

**DELIVERY SLIP**

|  |  |  |  |
| --- | --- | --- | --- |
|  | ***Name*** | ***Partner/Activity*** | ***Date*** |
| **From:** | Iván Díaz Álvarez | CESGA (JRA1.3) |  |
| **Moderated by:** | Małgorzata Krakowian | EGI.eu/WP1 |  |
| **Reviewed by** | Gergely Sipos  Alexandre Bonvin | EGI.eu/WP6  WeNMR |  |
| **Approved by:** |  |  |  |

**DOCUMENT LOG**

|  |  |  |  |
| --- | --- | --- | --- |
| ***Issue*** | ***Date*** | ***Comment*** | ***Author/Partner*** |
| **v.1** | 07/08/15 | Initial Version | Iván Díaz/CESGA |
| **...** |  |  |  |
| **...** |  |  |  |
| **v.n** |  |  |  |

**TERMINOLOGY**

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

**Contents**

[1 Example of heading 1 4](#_Toc418000825)

[1.1 Heading 2 4](#_Toc418000826)

[1.1.1 Heading 3 4](#_Toc418000827)

[2 Another example of heading 1 5](#_Toc418000828)

[2.1 Heading 2 5](#_Toc418000829)

[2.1.1 Heading 3 5](#_Toc418000830)

[3 Now for the lists and etc. 7](#_Toc418000831)

[3.1 Bullet lists 7](#_Toc418000832)

[3.2 Numbered lists 7](#_Toc418000833)

[4 Figures and captions 8](#_Toc418000834)

[4.1 Pictures 8](#_Toc418000835)

[4.2 Tables 8](#_Toc418000836)

[5 References 10](#_Toc418000837)

[Appendix I. Appendix example 11](#_Toc418000838)

# Introduction

The EGI Accounting Portal [R1] is an operational tool that processes, summarizes and displays the Accounting Repository data, gathered from all sites of the EGI infrastructure. Through the Accounting Portal all actors involved, including resource users, site and VO admins, NGI staff and others know how the resources are used, when, where, by whom and if this consumption is efficient.

The accounting data is gathered from individual sites using the APEL middleware, which parses grid a nd cloud job data from local job logs and send CPU, user, data and multicore job records using the SSM messaging system [R16] to a central accounting repository. This repository controls the correct publication of these logs and processes them internally. As a result of this process, a condensed version of these records is sent to the Accounting Portal using SSM. This data is combined with metadata from several sources to offer a comprehensive view to all involved actors and published in a graphical and understandable form.

To cover this array of needs, the portal offers a great number of customizable views and graphs of the data, some public, and some containing sensible data and restricted to privileged users. The portal needs to gather metadata from several sources to contextualize the data in a cohesive, geographical and functional way. The added value of this contextualization of the accounting data has caused the portal to become a data source for other operational tools and external data consumers.

The portal also publish the WLCG accounting reports, which are used globally to ascertain if resource pledges and MoUs negotiated between big WLCG VOs and sites from all the world are honoured.

The portal was in continuous development since 2004, so it contains a number of legacy elements, particularly on the interface. Indeed, the most recent updates focused on internal code. For this reason and also for the availability of new web tools that could offer a better experience, the portal users have expressed their desire for an improved look & feel through a more modern interface.

Furthermore, new emerging needs due to the introduction of new resource types (e.g. cloud, datasets, GPGPU, etc.) on the EGI infrastructure has been identified that requires new approaches on accounting data visualisation. The portal should provide users with additional views and different kind of queries to properly cover them. Usage of different types of resources (e.g. grid and cloud) will be merged in unique views when possible to present a global scenario to the end-users.

Another strategic requirement is to integrate the portal with the new AAI solutions for the EGI infrastructure that will be designed by the task JRA1.1, which is working on the new EGI AAI model taking into account the requirements collected by several stakeholders including the EGI-Engage competence centres.

The new version of the portal should also make it ready for the data deluge foreseen for the next years acting as component of an e-infrastructure able to manage big data. In addition, the new portal will be easier to debug, maintain and expand, paving the way for future improvements and changes that are unforeseen for now.

This document describes the design of the new Accounting Portal, an activity included in the workplan of the task JRA1.3.

The document is organised as follows. First, we will detail the architecture of the new portal, including the modules that will be implemented, and considerations that affect the high level design.

We then will present the different processes for requirement gathering, doing releases and validation.

The low-level design and technological consideration will be depicted in Section 3. A review of the technologies that will be used in the portal development will be also included.

Finally, we will review the roadmap to implement these changes and the requirements that should be addressed in each phase.

# Architecture of the Accounting Portal



Figure - Portal Architecture

The Accounting Portal is a web application based on Linux, Apache, and MySQL, which has as its primary function to provide users with customized accounting reports, containing tables and graphs, as web pages. It also offers RESTful web services to allow external entities to gather accounting data.

This chapter will detail the basic architecture of the Portal, which consists on:

A backend, which aggregates both data and metadata in a MySQL database, using the APEL SSM messaging system [R16] to interact with the Accounting Repository and several scripts which periodically gather the data and metadata described below.

* A Model represented by database schemas both external and internal which define database tables for several types of accounting (grid, cloud, storage, multicore, user statistics, ...) and metadata (topology, geographical data, site status, nodes, VO users and admins, site admins, ...) and a series of parametrized queries,
* A set of views which expose the data to the user. These views contain a form to set the parameters and metric of the report, a number of bi-dimensional tables showing the data parametrized by two selectable dimensions and filtered by several parameters, a line graph showing the table data, and some pie charts showing the percentage distribution on each dimension. This will be an evolution of the old view format, but now with interactive graphs, responsive in real time, reactive and only exposing advanced controls on user demand.

A graphical representation of these components is depicted on Fig. 1.

## 2.1 Backend

The Accounting Portal backend is a loose collection of messaging systems and scripts that gather accounting data and metadata from several external sources like GOCDB, the Operations Portal or WLCG for the portal consumption.

The accounting data are sent by each site to the central APEL accounting repository and processed and make into resumes using internal processes by APEL to make complex queries in the Portal practical, since using the raw data would make queries take up several orders of magnitude more. Metadata is a category of data, which complement that raw data and allows the portal to organize, categorize and impart new meaning to it.

### 2.1.1 SSM and Messaging

The Accounting Portal has to refresh its database periodically with data from the Accounting Repository to assure their freshness. The system used us SSM [R16], a queue messaging system based on ActiveMQ, which is also used for the communications between sites and the Accounting Repository. Since the repository uses it internally for all communications, it is also needed to gather the accounting data from them. The SSM system is composed by:

* A SSM loader for each accounting source (multicore, cloud, storage, etc..). This daemon waits for messages arriving on a queue and authenticates it with a DN and certificate. If the message is deemed valid, it is saved to a spool directory for further processing.
* A DB loader, this daemon monitors the spool directory and if there are messages these are introduced in the DB in order. This introduction at present does not delete the previous data in the tables, it only overwrites it, so manual intervention is needed for stale data.

### 2.1.2 Metadata Gathering

Metadata is a category of data which complement that raw accounting data and allows the portal to organize, categorize and impart new meaning to it. This metadata includes:

* **Geographical Metadata**: Which sites correspond to each country or NGI (which can comprise several countries). Generally, this follows current borders, but there are important exceptions. This is gathered from GOCDB using the GOG XML-based API.
* **Topological Metadata**: Sites are divided in trees, there are Country and NGI trees that correspond to geographical indications, but there are also trees for Tier1 and Tier2 sites, OSG sites and uncategorised sites. Inside Tier2 sites, their federation is also important and can trigger special code in some cases. Gathered from several sources, including OSG and WLCG databases.
* **Role Metadata**: Records about the members and managers of each VO, and the site admins. This metadata controls the access to restricted views. Gathered from GOCDB and individual VOMS servers constructing a list of individual VOMSes and querying them with the VOMS API.
* **Country affiliation data**: Each user record contains a user identifier with has his user name, institution and sometimes country. There are scripts in the backend with map each user with a country based on the institution which issues their certificate. This in turn is used in anonymized statistics on how much computation data from a country is used by country and also the distribution per country of resources used by any given country.
* **VO Data**: Although VO data is included in the normal accounting, there must be a master list of VOs to make their selection in the UI possible, filter incorrect VO names, mediate access to VO managers, and arrange accounting by VO discipline (such as “High Energy Physics”, “Biomedicine”, “Earth Sciences”, etc...). Gathered from the operations portal using its XML based APIs.
* **Site status metadata**: Sites must be filtered to exclude those not in production due to being closed or being test sites. There must be also metadata to aggregate the accounting history of sites with name changes. There are requirements to extend this functionality to NGIs. Gathered from GOCDB using its XML tables and internal tables compiled as part of EGI PROC 15 [R17] .
* **Pledge metadata**: The WLCG reports have to estimate if MoUs and other pledges between VOs and Tier sites are honoured, so the validity date and pledged hours are needed. Gathered from WLCG using their REBUS service.
* **Other metadata:** There are still other metadata we have not covered, like local privileges, SpecInt calculations, publication status, VO activities and some more. Some of these metadata is calculated internally using other types of metadata and published for other operational tools, like VO activity data, Site UserDN publishing

## 2.2 Model

The model in the portal is for the most part chosen to interchange data with the Accounting Repository and other operational tools, so for the most part is decided by external parties. The queries are parametrized to avoid SQL injections (SQL attack vectors based on malicious code on SQL input parameters), and will be implemented with ORM.

Since there are a large number of possible queries, and the accounting data has many reads but is only written on updates from the repository, the portal can be very aggressive with database indexes, and there will be periodic optimizations of these queries.

The queries have a common structure derived from the views. The new views will allow to separate these on basic and advanced parameters and visualize them geographically:

* **Metric**: It is the number we want to use for the accounting, it varies from view to view (e.g. Number of VMs on cloud), but we usually have:
  + Number of jobs: The number of jobs run, without regard for the CPU or time used.
  + CPU time: The time used by CPU core in hours while executing jobs.
  + Normalised CPU Time: The time used by CPU core multiplied by a corrective factor depending on a benchmark run on the machines. This benchmark can be SI2K (SpecInt2000), or HEPSPEC06.
  + Elapsed Time: The wall time, or real time spent in executing jobs, this should be greater than the CPU time since it also includes I/O and SO time.
  + Normalised Elapsed Time: Wall time normalised in the same way that the CPU time.
  + Efficiency: Wall time divided by CPU time. This indicated the percentage of time used doing calculation instead of doing I/O or servicing other tasks. This is important for pledges and VO admins.
  + Monetary Cost: A estimation of the equivalent monetary cost of the accounted work, this is only orientative.
* **Time period**: All queries are limited to a time period expressed in months, and which can go from January 2004 to the present.
* **Dimensions:** All data shown in the portal is parametrized by two dimensions (the “rows” and “columns” of the tables), these include, but are not limited to:
  + Date: The month of the accounting data
  + Region: The NGI or federation in which it was accounted
  + Country: The country that the data was accounted for.
  + VO: The VO that the jobs were run as.
  + Site: The site the data was accounted for
  + Number of processors: The number of cores used by the job.
* **VO Group**: The VOs that will appear in the accounting:
  + LHC: The VOs directly associated with the Large Hadron Collider in Geneva, comprises “alice”, “atlas”, “cms” and “lhcb”.
  + TOP10: The top 10 VOs in the selected range in raw CPU consumption.
  + ALL: All available VOs
  + Custom: It shows all VOs available in the range so the user can select which to display.
* dteam VO: It excludes the “dteam” and “ops” VOs, these are used for admin and test purposes and are not production ones.
* **Local Jobs**: Some sites can account jobs that have been processed locally on site bypassing the Grid middleware, the options are “Grid Jobs only”, “Grid and local jobs” and “Local Jobs only”.

There are customized reports and views which use other inputs, but in general those are the usual inputs of the common queries.

### 2.2.1 ORM

ORM or “Object/Relational Mapping”[R15] is a programming facility to map relational data onto virtual objects, so that the mismatch between the web application language and SQL is abstracted. The main advantage of this would be to simplify the code that would not need to contain explicit SQL statements and parameter matching and make it more robust in the case of schema migrations. In this case, the extremely dynamic nature of many queries would benefit for a move to ORM which would avoid unnecessary code.

Other benefit of ORM would be to abstract SQL away, so that in the future a NoSQL solution working directly with non-scalar values can be used, but that is not currently in the scope of the rewrite as the dependences with other tools and performance don't justify this yet.

For this rewrite of the Portal we will opt for using ORM for most of the queries, but in some cases with more specialized queries we will use a SQL pass-through (of course, with proper parametrization) to interact with the database.

## 2.3 Views

* **Main View**: The new redesigned main view of the Portal, compatible with EMI3 data, will implement a dashboard showing abstracted data:
  + A table with the following metrics for the last 24 hours, 30 days and last year:
    - Sum elapsed CPU time (normalized and absolute values)
    - Sum CPU (normalized and absolute values)
    - Number of jobs
* **Cloud View**: A new view of the sites working under the Federated cloud initiative, which uses Cloud middleware instead of the Grid one. Therefore, it has number of VMs instead of jobs, and no normalization for the moment. CPU time metrics are also in seconds, not hours and there are additional fields like network use and VM identifiers. This view will be fused with the old one that was grid data only, offering a combined view.
* **Tier1 View**: A specialised view for Tier 1 sites, it is equivalent to the main view, but with the site tree and topology of Tier1 (which is mostly flat). This view will be updated for cloud data.
* **Tier2 View**: Another specialised view for Tier 2 sites, with the topology of Tier2, which is Country – Tier 2 federation – Site. Although there are NGIs in the main view which expose federations (NGI\_UK), most don't, so this view offers a categorically different view of the topology. This will be updated for cloud data.
* **Countries View**: A view similar to the main one, but with countries instead of NGIs. Normally NGIs map to countries, but there are NGIs with more than one country (NGI\_IBERGRID), and NGIs which are a subset of a country (CERN), so this view is also different from EGI. New geographical graphs will plot data in map based diagrams.
* **OSG View**: A view for the sites in the OSG (Open Science Grid)[R9] initiative. OSG is a Grid in the United States with more than 100 sites. OSG sites are included in the WLCG report, which is a concern for the portal that surpasses mere EGI accounting. This view allows OSG staff to check the accounting that is used in the report. This will be also cloud compatible.
* **VO\_Discipline View**: A view that instead of using a tree with sites as leaves uses VOs as leaves, with scientific disciplines as branches. Of course, tables can use “Site” as a dimension. Recently expanded with a new Scientific discipline classification. This view will be updated with the new scientific disciplines classification.
* **VO\_Metrics View**: A view that shows a bird's eye view of the VOs in term of utilization. This view classifies automatically VOs into three tiers based on their CPU consumption and offers graphs and a table to see their monthly evolution. Formerly this view was based on weekly data. The new view will be month based.
* **WLCG Tier1 Report**: A report of consumption for high tier VOs , including the LHC ones, and others from biomedicine, astrophysics and HEP on Tier1 sites. It includes normalised CPU days and wall time days per month for CERN + Tier1, only Tier1 and site by site. This will also include integrated cloud based data.
* **WLCG Country Report**: A VO consumption report split by country and Tier x site, with number of jobs, normalised CPU and normalised elapsed time.
* **WLCG Tier2 Report**: A LHC specific consumption report on Tier2 sites. This report is geared to check if pledges are honoured, this means a number of hours must be spent with a certain efficiency that depends on the year.
* **InterNGI Report**: A detailed report of the relative consumption of resources between countries and NGIs. It will allow to set the time and metric used and includes:
  + A cross-over table with each NGI, rows are NGIs that use resources and columns NGIs whose resources are used. The diagonal represents the internal use of resources by each NGI. Inside each cell there is a number which represents consumption in the chosen metric, a blue percentage that represents % of consumption in the column, and a red percentage with the % of consumption in the column
  + A cross-over table like the above, but for countries.
  + A matrix of UserDN publication for NGIs(on which the accuracy of InterNGI depends). It includes the percentage of sites publishing UserDNs in the NGI, the normalised CPU time without a valid UserDN and % from the total and number of active users in the NGI
  + A matrix of UserDN publication like the above, but for Countries.
  + New map-based geographical charts.
* **InterNGI NGI Charts**: A series of reports for each NGI, After setting the metric and time range shows for each NGI in order:
  + A table with the distribution in absolute value and percentage of this NGI users per NGI.
  + A pie chart representing the first table.
  + A table with the consumption in absolute value and percentage of this NGI consumption per NGI.
  + A pie chart representing the second table.
  + New map based geographical charts.
* **InterNGI Country Charts**: The same as above, but with Countries.

### 2.3.1 User restricted Views

All of this views require authentication in the form of a valid grid certificate issued by a recognised CA, the portal has an internal metadata table with all the approved staff compiled from other operational tools.

These views are restricted since they expose UserDN records, with link the accounting with individuals with full names and affiliations. There are other views that use these tables, like the InterNGI reports, but these views anonymize sensible data.

In the restricted views, User names are obfuscated to avoid users taking a screen capture or scrape the data in an automatic way, User names are shown as “User number #”, and their real UserDN is only shown on mouse-over.

As part of the requirements these views will support ELIXIR-derived authentication.

Currently in the portal there are the following restricted views:

**VO Manager View**: A view designed for VO admins that lets them see the consumption done by users of their VO by UserDN and Role. There are versions of this view for both Cloud and Grid accounting data. VO admins can only see data from the VOs in which they have privileges.

**VO Member View**: A view in which VO Members can see their resource consumption, this view is limited only to the accounting of that particular user, it does not expose accounting data from others.

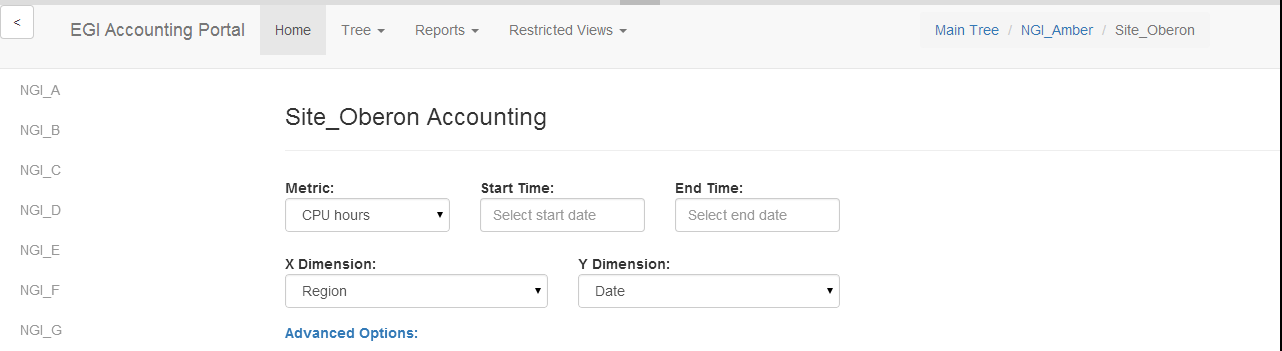
**Site Admin View**: A view similar to VO Manager View, but for site admins, it allows access to all user accounting in the site from all VOs, but does not expose accounting from outside that site.

### 2.3.2 View mock-ups

The views and reports described above will be implemented using the Bootstrap UI library, below we will see a series of mock-ups showing the functionality of the portal interface. These are not representative of the final appearance of the new Portal.

First, as can be seen in Fig.2, the page will have a drop down menu above with direct access to the views, reports and restricted views. There will also be a breadcrumb bar to the right to improve navigation. To the left, there will be a tree that will change depending on the view. In this case, it is a NGI tree with sites as leafs.

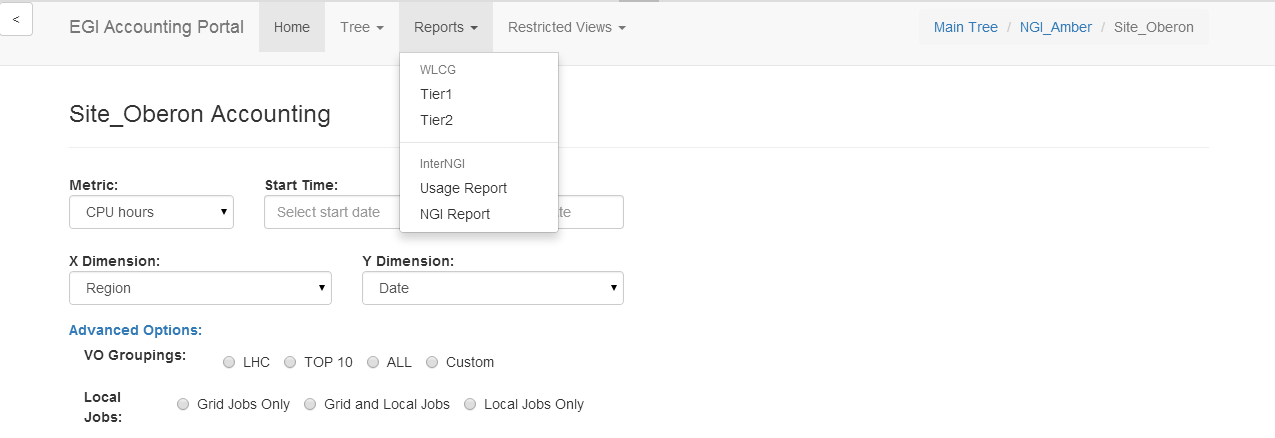
The main pane is occupied by a simplified form with just the basic options, the metric used, the start and end dates, and the dimensions. These widgets will have help popups explaining the different options to first time users.

Fig.2 – Interface mock-up

In Fig 3. we can see several functionalities at once. First, a menu for the navigation bar is shown. This menu is divided in sections and groups the reports by concern, making them more accessible to users with differing interests.

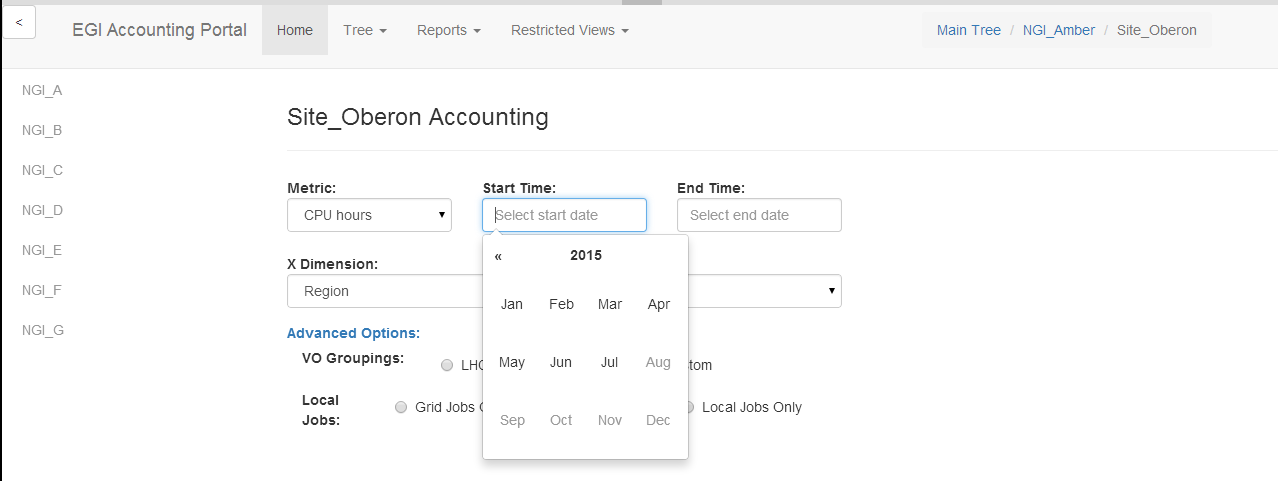
It can also be seen that the sidebar has been hidden from view, this can be toggled in real-time without reloading with the button in the upper right. This leaves more screen space for the reporting and reduces visual clutter.

Lastly, this figure also shows what happens after clicking “Advanced options”, the form now exposes options suitable for advanced users. This is done with a Javascript animation with no need to reload the page. Additional hidden sections can be added, even inside one another.



**Fig.3 – Expanded view**

In Fig.4 we can see the date-picker control. This is a Javascript powered control to input dates. In this case the control is configured for a granularity in months.



**Fig.4 – Date-picker detail**

# Processes

In order to make possible these changes in the Portal and to guarantee proper feedback and communications between the users, policy makers and technical providers involved some processes have been established. We will talk here about the requirement gathering, release and testing procedures.

## 3.1 Requirement gathering process

A tool like the Accounting Portal has a large variety of users with very different needs. Some are only interested on checking monthly data for their site or VO, others are power users that want programmable access to large quantities of data. Others want a birds eye view of the use of the infrastructure. This users can also be from anywhere in the world.

This means that requirement gathering is particularly difficult. With this in mind, the Accounting Portal Advisory and Testing Board [R2] (from now on, the ATB) was established.

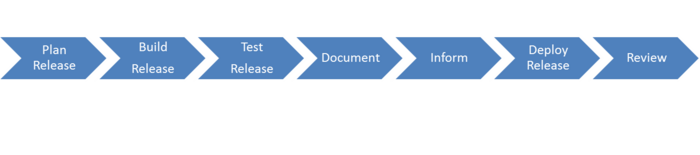
This board is composed from the Accounting Portal and Repository PTs, EGI Operations staff, VO and site admin representatives and other actors, and represents an open Special Interest Group, which welcomes people interested in the current and future evolution of the portal.

The ATB convened several times to define the future roadmap and requirements of the new Accounting Portal, and this document is composed primarily of its input. All requirements are gathered and proposed by members of the ATB, and future requirements should also be vetted and prioritized by this board.

These requirements are reported in EGI RT tickets, and can be seen in the Accounting Portal Requirements Dashboard [R3], along with their updates and resolution status. These tickets are discussed and prioritized by the ATB, which meets monthly. The current list of requirements integrated in the roadmap can be seen in the Roadmap section in this document.

## 3.2 Release Process

The Portal will follow the EGI-Engage release protocol, detailed in Figure 5 and explained below. There will be a release each 6 months starting in M14, totalling 3 planned releases in the total life of the project

Fig.5 – EGI Release Process

### 3.2.1 Plan Release

Releases for the portal are planned within 6 months intervals and detailed in Section VI. These releases will implement requirements and features according to the roadmap, and any future requirements approved by the ATB having in account the resources and time available for the implementation.

The OMB will be notified with at least a week in advance so it can notify all users and manage problems.

Also, the release should be presented in the monthly OMB meeting prior to the release to production.

In the case a emergency release is needed, it will be announced in the following meeting.

### 3.2.2 Build Release

Due to the open-source nature of the developed software, and since the Portal uses Git internally, the release will be published in GitHub [R5], which is a Public release platform based on Git.

This platform will document the changes in the code and each release will be marked by a tag, a metadata flag that can be assigned to any commit. Since the portal is an Operational Tool, it will not need to be packaged as an UMD release, as it will be only installed as the central production, development and testing instances and the instances part of the regional portals.

### 3.2.3 Distribution

Since the Portal is a Web application it should only need to be downloadable as source code through GitHub, which paired with the correct Apache configuration files, certificates and computational resources should be executable on any standard physical or virtual Apache server (the central instance runs on a virtualised environment).

Tarballs may be offered for releases for archival purposes, but the preferred download method should be using Git, since this will preserve the modification history, will be directly upgradeable (in the case of tarballs, overwriting the previous installation does not work if some files have been removed on the interim, and deleting and replacing does not respect additional local files and is risky), and local modifications by regional users of the portal can be sent back to the main portal transparently via git or diff files.

A complete guide to install the portal will be available as part of the documentation.

### 3.2.4 Test Release

Testing of the release will be the responsibility of both the development team and the ATB. In the case of main releases there should be member for the user communities in the ATB which will perform acceptance tests on the release based on the expectations and use experience they have as members of the community they represent.

The testing phase will be announced at the end of the development phase of the release. This announce will include all the PTs (as operational tools have dependences) and the ATB members which should do the testing. The announcement should contain:

* Release notes, containing changelog, installation & configuration steps to apply the update, any known issues
* Documentation links
* Detailed test plan
* All the information needed by the EGI Quality Criteria[R6] set by the SA2 activity for the software providers.
* The expected release date and the kind of testing will depend on each specific release and on its importance.

If a test fails a report will be produced and the release sent back to development to restart the cycle. Tests will include a documentation review and a documentation update if needed. The test phase can be performed internally to the development team if no other tools or services are affected.

### 3.2.5 Documentation

The Accounting Portal Development team is responsible for creation and maintenance of documentations, instructions and manuals related to the tool in collaboration with EGI Operations team.

Before each release documentation should be checked and updated as needed as part of the testing process.

Documentation will be available both for end-users and to install and maintain the portal.

### 3.2.6 Notification

As written above, the information about a next release should be communicated during a OMB meeting at least one week before release, but preferably a month before.

This communication can be in the form of a single presentation slide sent to the email address of EGI Operations[ R7] before the meeting.

Additionally, a broadcast can be done on the Operations Support Mailing List [R8].

### 3.2.7 Deploy Release

For changes of high impact and high risk, the steps required to reverse an unsuccessful change or remedy any negative effects shall be defined.

### 3.2.8 Review Release

Each release should be monitored for success or failure and the results shall be analysed internally.

## 3.3 Validation Process

The validation process will be performed by the development team and members of the ATB. There will be a instance dedicated to the testing process for their perusal, or “Testing Instance”.

Ideally, for doing the verification fully and correctly, the ATB should include the following.

* All the members of the Accounting Portal developer team.
* All the members of the Accounting Repository development team.
* Members of the EGI Operations team.
* Members from other Operational Tools teams which have dependences with the portal and have expressed their desire to be a part of the ATB.
* At least one VO admin
* At least one Site admin, preferably from a Tier-x site.
* A member from the EGI Security Team

The Portal development team should contribute the following documentation to the ATB, in addition to allow access to the testing instance:

* Release notes, containing changelog, installation & configuration steps to apply the update, any known issues
* Documentation links
* Detailed test plan
* All the information needed by the EGI Quality Criteria[R6] set by the SA2 activity for the software providers.

The testing team will decide if problems are cosmetic, minor, serious or critical. Depending on the seriousness of the problems and if they affect critical areas of the functionality (as opposed to being problems that affect code not critical to the release), will decide the course of action, from immediate patching, to defer changes to the new release or even a release rollback.

This testing will not preclude standard bug fixing and problem solving in the production release, which will not need ATB involvement, but can be notified to them, particularly when the changes are disruptive or contravene the usual behaviour.

# 4. Technologies

We will give here a brief description of the technologies that will be used for the rewrite

## 4.1 Python / Django

Python [R10] is a high-level programming language which emphasizes code that is clear, readable and compact, and which can be developed and maintained faster and which less human effort that traditional languages like Java or C++. This is supplemented by a very comprehensive set of libraries that facilitate common tasks (the “batteries included” philosophy)

Python is not limited to the imperative programming paradigm and also supports functional and object-oriented programming. Like Java, Python code includes automatic memory management is portable and very platform agnostic. A particularity of the language is that uses white-space and tabulation as block delimiters (instead of e.g. braces “{}” in C-like languages), enforcing good formatting in the code.

Its lightweight nature and library support has seen its use as a script language, both individually or as part of bigger systems, and its dynamic typing and memory management makes it very useful for web management, having the flexibility and security of Perl, but with better maintainability.

Django[R11] is an Open Source web application development framework written and designed to work with Python, which implements the MVC architectural pattern and is immensely flexible and configurable. Its goal is to ease the creation of complex database-driven sites making basic functionality very easy to use, and promoting reusability and pluggability of other components as needs change and complexity arises.

Django includes:

* An ORM (Object Relational Mapper) module which mediates between an object model and a relational database. This can be changed modularly with more complex ORM such as Alchemy.
* A Web templating engine and language that defines and serve the dynamic pages, that can be extended easily with new tags based on Python code.
* A regular-expression based URL dispatcher that acts as controller and captures automatically the arguments and promotes a human-readable URL format.
* A form serialization and validation system which can automate the translation between web forms and objects for storage in the database.
* An event signalling system.
* An extensible and modular authentication system. It can support certificate, LDAP, login and token based authentication and others.
* Security features to limit SQL injection, cross-site request forgery and cross-site scripting.

The Accounting Portal development team has experience in the use of Python and Django in the development of the Metrics Portal [R12], so it is a proven technology with hands-on experience.

## 4.2 Dojo Toolkit

The Dojo Toolkit[R13] is an Open source, modular JavaScript toolkit created to ease the rapid development of cross-platform, AJAX-based applications and web sites. It abstracts the differences between major browsers, and provides a rich set of expandable modules for dynamic charts, internationalization, Menus, sortable tables, animations, asynchronous communication, etc..

The main benefit for the Accounting Portal would be to load data in an asynchronous manner with changes in the interface and to avoid having to refresh the page to do that. The pages will be also responsive to the device and change layout fluidly and seamlessly depending on the available space.

The dynamic chart library of Dojo will allow responsive and dynamic charts that not only react to changes in the input values instantly, but will also show more information clicking in one part of them, and will be exportable to SVG for further edits. In some cases, the type of the graph will be also changeable on the fly to allow a better view of some data.

Forms will be made simpler, but without losing functionality, making advanced functionality not visible until the user selects it. In this way, new users will not be overwhelmed by the number of options, and advanced users will only need to expand the relevant parts of the form without having to reload the page.

## 4.3 Bootstrap Library

Bootstrap[R14] is a open source front-end framework based on CSS and JavaScript to create websites. It contains templates and modules for the typography, buttons, navigation, forms and interface components of web sites.

It was initially conceived by Twitter staff to promote code reuse, interface consistency, and responsive web designs that adapt fluidly to the device and available canvas space.

It is supported by the main browsers like Chrome, Firefox, Opera, Internet Explorer and Safari and is completely integrated with jQuery.

# Roadmap

Below in Table 1, the current status of the roadmap as of the date of this document. Further changes and updated roadmaps can be consulted on the EGI wiki for the JRA1.3 task[R4]

Table 1 – Accounting Portal Roadmap for EGI-Engage

| Task number | Task Name | Start Date | Release Date | Dependencies |
| --- | --- | --- | --- | --- |
| 3.2.1 | Requirements collection | 03/15 | 08/15 |  |
| 3.2.2 | D3.1: Technical design of the new Accounting Portal and implementation plan | 03/15 | 08/15 | 3.2.1 |
| 3.2.3 | Modernize the accounting Portal with the adoption of technologies easier to maintain  [**First/front page (New)**](https://rt.egi.eu/rt/Ticket/Display.html?id=8823) [#8823] | 06/15 | 01/16 | 3.2.1 |
| 3.2.4 | Simplify access to some basic functionality. Avoid the use of complex forms for common statistics and get accounting information for some common queries | 07/15 | 04/16 |  |
| 3.2.5 | Support Cloud Usage Record V0.4 and accounting of long running VMs | 07/15 | 04/16 | Depends on Accounting Repository implementation |
| 3.2.6 | EGI Federated AAI Integration | 04/16 | 04/16 | Depends on outcome of AAI TF and EGI policy |
| 3.2.7 | D3.5: First release of the new Accounting Portal deployed in production  [User Documentation](https://rt.egi.eu/rt/Ticket/Display.html?id=8824) **[#8824]**  [Improve graphs visualization](https://rt.egi.eu/rt/Ticket/Display.html?id=8827) **[#8827]**  [Support for EGI-Engage metrics](https://rt.egi.eu/rt/Ticket/Display.html?id=8822)  **[#8822]** | 05/16 | 04/16 | 3.2.2, 3.2.3, 3.2.4, 3.2.5 |
| 3.2.8 | Define a complete API to get accounting data directly from the accounting portal | 07/16 | 12/16 |  |
| 3.2.9 | Integrate data in maps for a graphical distribution of the information | 07/16 | 01/16 |  |
| 3.2.10 | Integrate analytics to extract intelligence operation from the data | 08/16 | 05/17 |  |
| 3.2.11 | D3.10: Second release of the new Accounting Portal deployed in production  [**Function of basic analisys**](https://rt.egi.eu/rt/Ticket/Display.html?id=8821) [#8821] | 02/17 | 02/17 | 3.2.7, 3.2.8, 3.2.9, 3.2.10 |
| 3.2.12 | Support Data Accounting | 01/17 | 08/17 | Depends on Accounting Repository implementation |
| 3.2.13 | Support GPGPU Accounting | 01/17 | 08/17 | Depends on Accounting Repository implementation |
| 3.2.14 | Support Big Data tools | 01/17 | 08/17 | Depends on Accounting Repository implementation |
| 3.2.15 | D3.19: Final release of the new Accounting Portal deployed in production | 08/17 | 08/17 | 3.2.11, 3.2.12, 3.2.13, 3.2.14 |

# 6. References

|  |  |
| --- | --- |
| ***No*** | ***Description/Link*** |
| R1 | Production Accounting Portal  [Accounting Portal](http://accounting.egi.eu/) |
| R2 | Board to assist on requirement gathering and testing  [Accounting Portal Advisory and Testing Board](#Accounting_Portal_Advisory_and_Testing_Board) |
| R3 | RT Dashboard that groups current and future requirements for the Portal  [Accounting Portal Requirements Dashboard](https://rt.egi.eu/rt/Dashboards/5538/AccPortal-Requirements) |
| R4 | EGI Wiki page for the Accounting Task with roadmaps  [EGI-Engage:TASK JRA1.3 Accounting](https://wiki.egi.eu/wiki/TASK_JRA1.3_Accounting) |
| R5 | CitHub Collaborative Portal  [GitHub](https://github.com/) |
| R6 | EGI Quality Criteria  EGI Quality Criteria |
| R7 | EGI Operations e-mail  [EGI Operations Contact e-mail](mailto:operations@egi.eu) |
| R8 | Operation Support Mailing list  [Operations Support mailing list](mailto:operations-support@mailman.egi.eu) |
| R9 | OSG (Open Science Grid)  [Open Science Grid webpage](http://www.opensciencegrid.org/) |
| R10 | Python homepage  [Python.org](https://www.python.org/) |
| R11 | Django Project homepage  [Django Project](https://www.djangoproject.com/) |
| R12 | Metrics Portal homepage  [Metrics Portal](https://metrics.egi.eu/) |
| R13 | Dojo Toolkit homepage  [Dojo Toolkit](http://dojotoolkit.org/) |
| R14 | Bootstrap homepage  [Bootstrap Framework](http://getbootstrap.com/) |
| R15 | What is Object/Relational Mapping?  [What is Object/Relational Mapping?](http://hibernate.org/orm/what-is-an-orm/) |
| R16 | APEL SSM Messaging System  [APEL SSM Messaging System](https://wiki.egi.eu/wiki/APEL/SSM) |
| R17 | EGI PROC 15: Resource Center Renaming  [EGI PROC 15: Resource Center Renaming](https://wiki.egi.eu/wiki/PROC15_Resource_Center_renaming) |