



EGI-Engage

Technical design of the new Accounting Portal and implementation plan

D3.1

Date	26 August 2015
Activity	WP3
Lead Partner	CESGA
Document Status	FINAL
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. The users are then presented with homogeneous view of the gathered data and displaying it in a user-friendly way. The portal helps EGI members and external parties understand resource utilization, serving the needs of various 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.



This material by Parties of the EGI-Engage Consortium is licensed under a [Creative Commons Attribution 4.0 International License](https://creativecommons.org/licenses/by/4.0/).

The EGI-Engage project is co-funded by the European Union (EU) Horizon 2020 program under Grant number 654142 <http://go.egi.eu/eng>

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

	<i>Name</i>	<i>Partner/Activity</i>	<i>Date</i>
From:	Iván Díaz Álvarez	CESGA (JRA1.3)	
Moderated by:	Małgorzata Krakowian	EGI.eu/WP1	
Reviewed by	Gergely Sipos Alexandre Bonvin	EGI.eu/WP6 WeNMR/MoBrain	
Approved by:	AMB and PMB		31/08/2015

DOCUMENT LOG

<i>Issue</i>	<i>Date</i>	<i>Comment</i>	<i>Author/Partner</i>
v.1	07/08/15	Initial Version	Iván Díaz/CESGA
v.2	11/8/15	After Internal Review	Iván Díaz/CESGA
v.3	16/8/15	After External Review	Iván Díaz/CESGA
FINAL	26/08/15	Final version	Iván Díaz/CESGA

TERMINOLOGY

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

Contents

1. Introduction	4
2. Architecture of the Accounting Portal.....	7
2.1 Backend.....	9
2.1.1 SSM and Messaging.....	9
2.1.2 Metadata Gathering	9
2.2 Model.....	11
2.2.1 ORM	12
2.3 Views.....	13
2.3.1 User restricted Views	15
2.3.2 View mock-ups	16
3. Processes	19
3.1 Requirement gathering process	19
3.2 Release Process.....	19
3.2.1 Plan Release	20
3.2.2 Build Release	20
3.2.3 Distribution	20
3.2.4 Test Release	21
3.2.5 Documentation	21
3.2.6 Notification	22
3.2.8 Review Release.....	22
3.3 Validation Process.....	22
4. Roadmap.....	24
5. Technologies	28
5.1 Python / Django	28
5.3 Bootstrap Library	29

1. Introduction

The EGI Accounting Portal¹ 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, the EGI Community (resource users, site and VO admins, NGI staff and others) knows how the resources are used, when, where, by whom. The portal can also help these parties to understand the efficiency of resource consumption.

The accounting data is gathered from individual sites using the APEL middleware², which parses grid and cloud job data from local job logs and send CPU, user, data and multicore job records using the SSM messaging system³ 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 EGI Community and is 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 restricted to authorized 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. This data is updated incrementally from the APEL side, but not dropped automatically after a period.

The portal also publish the WLCG accounting reports as a special use case, 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, and as result, it contains a number of legacy elements, particularly on the interface. For this reason and, also, for the availability of new web tools that could offer a better experience, the portal users have expressed their need for an improved look and 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 have been identified that requires new approaches on accounting data visualisation. The portal should provide users with additional views and different kind of query tools to properly cover them. Usage of different types of resources (e.g. grid and cloud) will be merged when possible to present a global view to the end-users.

¹ <http://accounting.egi.eu/>

² <https://github.com/apel/apel>

³ <https://wiki.egi.eu/wiki/APEL/SSM>

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, to implement a common authentication mode across organizations.

Besides this, the portal also needs to be improved taking into account the requirements collected by several stakeholders including the EGI-Engage competence centres representing large Research Infrastructure communities from the ESFRI⁴ roadmap. In addition, historical accounting information needs to be preserved and provided including information for Resource Centres and Operations Centres during the entire lifetime, including the post-decommissioning phase.

The new version of the portal should also make it ready for the new types of data 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.

In brief, the work will consist of:

- A complete rewrite of all the code from PHP to Python, improving maintainability, functionality and code separation.
- A reimplementation of all the views with new code, and new simplified user interfaces with AJAX functionality, so that information is refreshed on the go and advanced functionality is exposed only when requested by the user.
- New views for new types of accounting, fusing old views into merged views and new reports that will be organized differently.
- Re-implementing the model supporting ORM⁵ and new types of accounting.
- A new graphic system that will allow interaction to expose further details or zoom certain parts and also support geographically-backed graphs.
- A complete client code rewrite to support new client technologies like HTML5, Dojo and Bootstrap, responsive layouts that change depending of the device and new templates and look.

This document describes the design of the new Accounting Portal, an activity included in the workplan of the task JRA1.3. In Section 2, we detail the architecture of the new portal, including the modules that will be implemented, and considerations that affect the high-level design.

In Section 3, different processes for requirement gathering, building releases and validation are presented.

⁴ https://ec.europa.eu/research/infrastructures/index_en.cfm?pg=esfri

⁵ https://en.wikipedia.org/wiki/Object-relational_mapping

In Section 4, the document shows the defined roadmap to implement these changes and the requirements that should be addressed in each phase.

Finally, the low-level design and technological consideration are depicted in Section 5. A review of the technologies that will be used in the portal development are also included.

2. Architecture of the Accounting Portal

The Accounting Portal is a web application based on 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 details 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 system⁶ 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 etc.) and metadata (topology, geographical data, site status, nodes, VO users and admins, site admins etc.) and a series of parametrized queries,
- A set of views that expose the data to the user. These views contain a form to set the parameters and metric of the report, a number of tables showing the data parametrized by two selectable dimensions and filtered by several parameters, a line graph showing the table data, and pie charts showing the percentage distribution on each dimension. It is planned that this part of the portal will evolve 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.

⁶ <https://wiki.egi.eu/wiki/APEL/SSM>

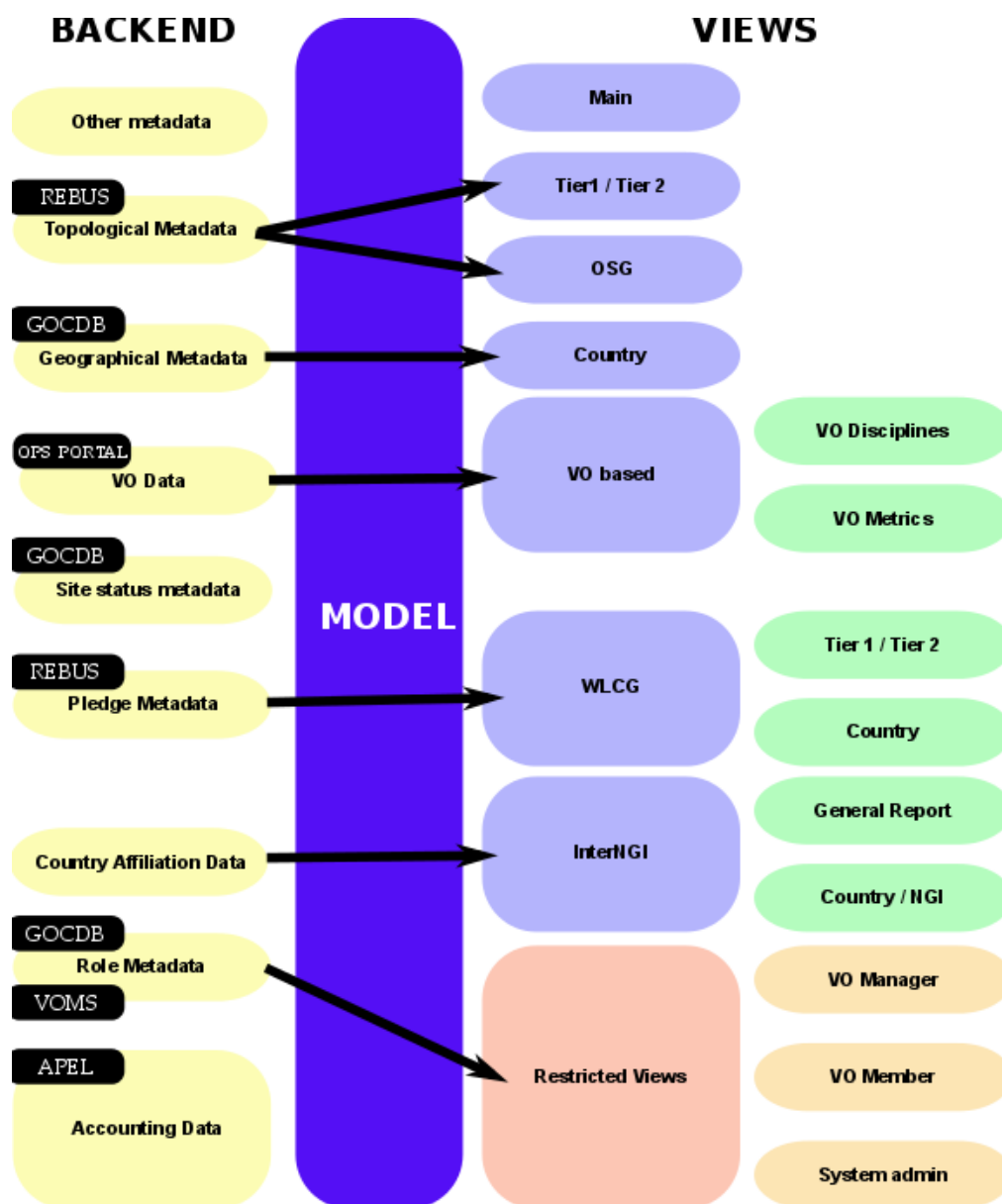


Fig.1 – Accounting Portal Architecture

2.1 Backend

The Accounting Portal backend is a loose collection of messaging systems and scripts that gather accounting data from the accounting repository and metadata from several external sources like for example, the EGI Service registry (GOCDB)⁷, the EGI Operations Portal⁸ or WLCG.

The accounting data are sent by each site to the central APEL accounting repository and processed. Resumes are made using internal processes by APEL to make complex queries in the Portal efficient. 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 that information published is up-to-date. Secure Stomp Messenger (SSM)⁹, a queue messaging system based on ActiveMQ, is used for synchronization purpose and also for the communication between sites and the Accounting Repository. 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 them with a DN and certificate. If a 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 imported in the DB in order. This import at present does not delete the previous data in the tables; it only overwrites it. Manual intervention is needed for stale data.

The accounting data is sent several times per day by APEL in chunks of 1000 registers. This obviates the need for the portal to do pull requests.

2.1.2 Metadata Gathering

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

- **Geographical Metadata:** Country and NGI affiliation of sites. Generally, this follows current borders, but there are important exceptions. This is gathered from GOCDB using its XML-based API.

⁷ <http://goc.egi.eu/>

⁸ <http://operations-portal.egi.eu/>

⁹ <https://wiki.egi.eu/wiki/APEL/SSM>

- **Topological Metadata:** Sites are presented in trees, there are Country and NGI trees that correspond to geographical classifications, but there are also trees based on topological classifications like Tier1 and Tier2 sites, OSG sites and uncategorised sites. Inside Tier2 sites, the federation they belong to is also important and can trigger special code in some cases. Gathered from several sources, including OSG and WLCG databases.
- **Role Metadata:** VO members and managers, and the site admins records. This metadata controls the access to restricted views. Information is gathered from GOCDDB 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 that has his/her user name, institution and sometimes country. Scripts in the backend map each user with a country based on the institution which issues their certificate. This data is used in anonymized statistics per country on: how much resources from other countries are used by given country and the distribution of its resources used by other countries.
- **VO Data:** To make possible VO selection in the user interface, the portal stores lists of VOs. They are also used to filter incorrect VO names, provide access to VO managers, and arrange accounting by VO discipline (such as “High Energy Physics”, “Biomedicine”, “Earth Sciences”, etc.). Information is gathered from the Operations portal using its XML based APIs.
- **Site status metadata:** Sites must be filtered to exclude those that are not in production (due to being closed or being in test mode). There must be also metadata to aggregate the accounting history of sites whose name has been changed. There are requirements to extend this functionality to NGIs. Information is gathered from GOCDDB using its XML tables and internal tables compiled as part of EGI PROC 15¹⁰.
- **Pledge metadata:** The WLCG reports have to contain only those sites where MoUs or other pledges between VOs and sites are honoured, so the validity date and pledged hours are needed. Information is gathered from WLCG using the REBUS service.
- **Other metadata:** There are also other metadata like local privileges, SpecInt calculations, publication status, VO activities and more. Some of these metadata is calculated internally using other types of metadata and published for other EGI operational tools, like VO activity data and Site UserDN publishing

¹⁰ https://wiki.egi.eu/wiki/PROC15_Resource_Center_renaming

2.2 Model

The model in the portal is mostly used to interchange data with the Accounting Repository and other operational tools, so the major part is determined 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 with only written on updates from the repository, the portal can be very aggressive with database indexes. Queries will be periodically optimized.

The queries have a common structure derived from the current views. The new views will allow separating these on basic and advanced parameters and visualizing them geographically:

- **Metric:** Metric is the visualised accounting value and it varies depending on the view. The basic list of metrics is as follows:
 - 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: An estimation of the equivalent monetary cost of the accounted work, this is only indicative.
- **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 data was accounted

- Country: The country in which data was accounted
- 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 monitoring and test purposes and are not used for research.
- **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”¹¹ 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.

¹¹ <http://hibernate.org/orm/what-is-an-orm/>

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

Views in the portal differ in the type of showed accounting data, the site organization or the restricted nature of data. A restructured version of the following Views are foreseen in the new portal. New views could be added according to the user requirements that will be gathered during the project lifetime. The Cloud view is a new view of the sites that are part of EGI Federated cloud platform, which uses Cloud middleware. Therefore it requires a dedicated set of metrics like number of Virtual machines. CPU time metrics are expressed in seconds instead of 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.

- **Main Grid/Cloud View:** The new redesigned main view of the Portal, compatible with EMI3 data, will implement a dashboard showing 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

The Cloud view is a new view of the sites that are part of EGI Federated cloud. Therefore, it requires a dedicated set of metrics like number of Virtual machines. CPU time metrics are expressed in seconds instead of hours and there are additional fields like network use and VM identifiers. This view will be merged with the old one that was grid data only, offering a combined view.

- **Tier1 View:** A specialised view for WLCG Tier 1 sites, it is equivalent to the main view, but with the site tree and topology of WLCG Tier1. This view will be updated with cloud data.
- **Tier2 View:** Another specialised view for WLCG Tier 2 sites, with the topology of WLCG Tier2, which shows 3 levels: Country, Tier 2 federation, Site. Although there are NGIs in the main view which are organized as federations (such as NGI_UK), so this view is different from the main one in those cases. This will be updated with cloud data.
- **Countries View:** A view similar to the main view, but showing data per country. Usually NGIs map to countries, but there are several NGIs with more than one country (e.g. NGI_IBERGRID), and NGIs which are a subset of a country (e.g. CERN). New geographical graphs will plot data in map based diagrams.

- **OSG View:** A view for the OSG (Open Science Grid)¹² sites. OSG is an e-infrastructure in the United States with more than 100 sites. OSG sites are included in the WLCG report, which is important for organizations external to EGI. This view allows OSG staff to check the usage of resources. This view will be also cloud compatible.
- **VO_Discipline View:** A view that provides accounting data per VO scientific disciplines defined by EGI. This view will be updated with the new scientific disciplines classification.
- **VO_Metrics View:** A view that shows high level view on 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 but the new one will migrate to monthly based.
- **WLCG Tier1 Report:** A report of consumption for high consumption WLCG tier VOs, including the LHC ones, and others from biomedicine, astrophysics and HEP on WLCG Tier1 sites. It includes normalised CPU days and wall time days per month for CERN + WLCG Tier1, only WLCG Tier1 and site by site. This will also include integrated cloud based data.
- **WLCG Country Report:** A VO consumption report split by country and WLCG Tier x site, with number of jobs, normalised CPU and normalised elapsed time.
- **WLCG Tier2 Report:** A LHC specific consumption report on WLCG 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 observation period and metric used and includes:
 - A cross-over table with each NGI where 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 mentioned previously, but for countries

¹² <http://www.opensciencegrid.org/>

- A matrix of UserDN¹³ 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 these 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 mentioned previously, but for Countries.
- **RESTFUL XML interface:** The views will have a REST¹⁴ web service interface to provide data for external users and tools. This REST service will accept as input a series of GET parameters functionally equivalent to normal form inputs and give an XML representation of the results as output. In each page there will be an automatic clickable link that will provide a hyperlink to the XML view equivalent to the page currently viewed.

2.3.1 User restricted Views

All these views require authentication in the form of a valid grid certificate issued by a CA recognised in EGI. The portal has an internal metadata table with all the approved staff (which has a valid certificate and an appropriate role) compiled from other operational tools and, eventually, for external AAI system (as those adopted by research infrastructures).

These views are restricted since they expose the 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.

¹³ User Distinguished Name

¹⁴ Representational State Transfer

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 showed as “User number #”, and their real UserDN is only shown on mouse-over.

As part of the requirements, these views will support the authentication mechanism of the research infrastructures involved in the project like ELIXIR. The requirements for each of these RIs are still under discussion. Currently in the portal, there are the following restricted views:

VO Manager View: A view designed for VO admins that allows seeing the consumption 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 own resource consumption, this view is limited only to the accounting of that particular user, and 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 re-implemented using the Bootstrap UI library to enable the new UI look and functionalities. Below a series of mock-ups are presented showing the functionality of the new 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. In the left part of the page, there will be a tree that will change depending on the view. In this case, it is a NGI tree with sites as leafs.

A simplified form with just the basic options, the metric used, the start and end dates, and the dimensions will occupy the main panel. These widgets will have help popups explaining the different options to first time users.

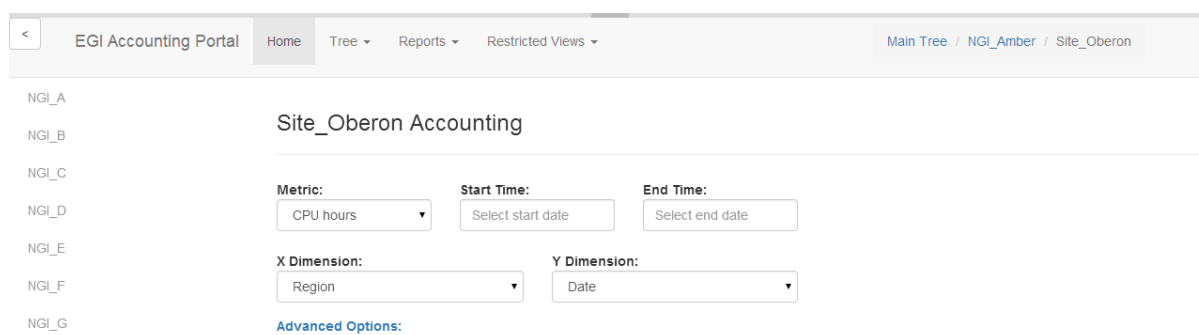


Fig.2 – Interface mock-up

In Fig 3. several functionalities can be seen 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.

Finally, 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.

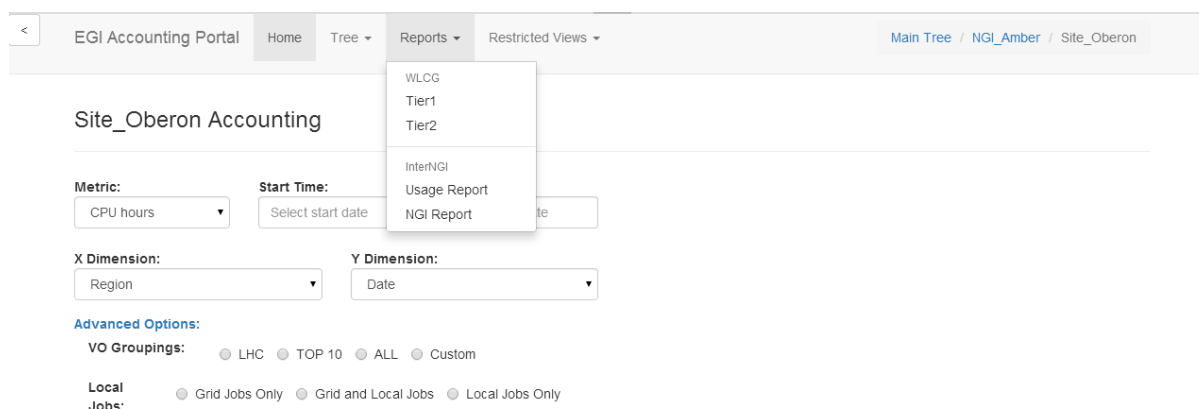


Fig.3 – Expanded view

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.

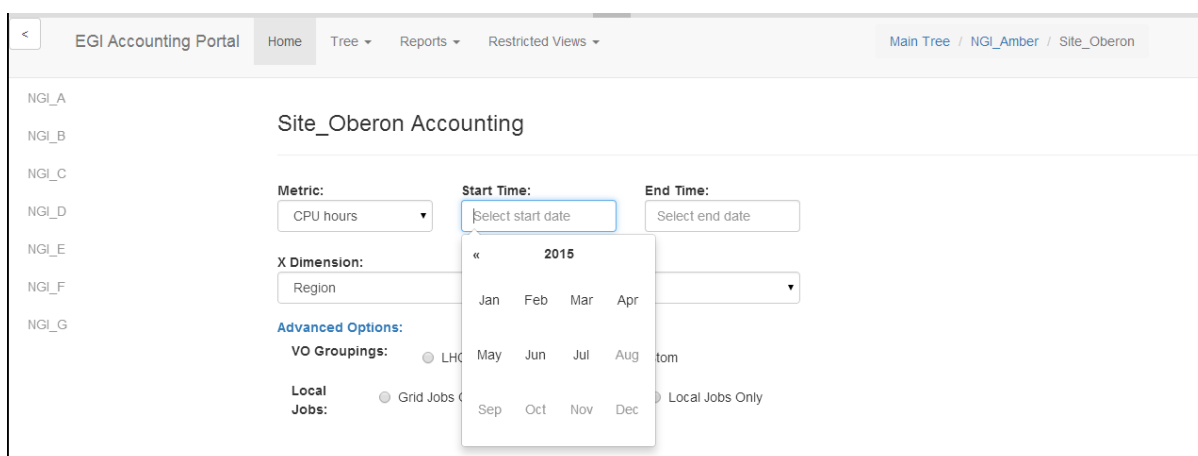


Fig.4 – Date-picker detail

3. Processes

In order to guarantee that changes align with user needs and ensure proper communication with policy makers and technical providers Accounting portal development will follow Instructions for Production teams¹⁵ which detail activities necessary for change, release and deployment management process.

3.1 Requirement gathering process

A tool like the Accounting Portal has users with very different needs. Some of them are interested in web-base access to small amount of data, for example checking monthly data for a site or a VO. Other users are interested in programmable access to large quantities of data. There is also a need to provide a high level view of the infrastructure use.

This means that requirement gathering is particularly difficult. With this in mind, the Accounting Portal Advisory and Testing Board¹⁶ (ATB) has been established.

This board is composed from the Accounting Portal and Repository PTs, EGI Operations staff, VO and site admin representatives and other users. Its mandate is to help developers in requirement prioritization and releasing process. The board provides forum to discuss the tools evolution that meet the expressed needs of the EGI community.

The ATB convened two times in 2015 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 Dashboard¹⁷, along with their updates and resolution status. 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 release and deployment process, detailed in Figure 5 and explained below. There will be a release each 6 months starting in M14 - 3 in total.

¹⁵ https://wiki.egi.eu/wiki/Instructions_for_Production_Tools_teams

¹⁶ https://wiki.egi.eu/wiki/Operations_Tools_Advisory_Groups#Accounting_Portal_Advisory_and_Testing_Board

¹⁷ <https://rt.egi.eu/rt/Dashboards/5538/AccPortal-Requirements>

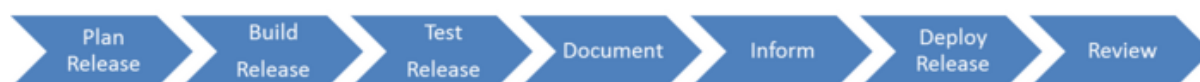


Fig.5 – EGI Release Process

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. Any future requirements approved by the ATB will take in account the resources and time available for the implementation.

At least the Operations Management Board¹⁸ will be notified with a minimum a week in advance so it can notify all users and manage problems. Other groups representing users like the UCB¹⁹ and the WP6 Competence Centres²⁰ will be also informed. Also, the release should be presented in the monthly OMB meeting prior to the release to production.

In the case an 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 GitHub²¹, 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 eventually be packaged as an UMD release, and 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).

¹⁸ <https://wiki.egi.eu/wiki/OMB>

¹⁹ https://www.egi.eu/about/policy/groups/User_Community_Board_UCB.html

²⁰ https://wiki.egi.eu/wiki/EGI-Engage:Competence_centres

²¹ <https://github.com/>

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 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 announcement will reach all the PTs (as operational tools have dependences) and the ATB members which should perform 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
- 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 an OMB meeting at least one week before release, but preferably a month before. Other groups like the UCB²², the WP6 Competence Centers²³ and all interested users will be notified. This communication can be in the form of a single presentation slide sent to the email address of EGI Operations²⁴ before the monthly OMB. Additionally, a broadcast can be done on the Operations Support Mailing List²⁵.

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 an 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
- Possibly a research community representative from new research communities.
- At least one Site admin, preferably from a WLCG Tier-x site.
- A member from the EGI Security Team

²² https://www.egi.eu/about/policy/groups/User_Community_Board_UCB.html

²³ https://wiki.egi.eu/wiki/EGI-Engage:Competence_centres

²⁴ operations@egi.eu

²⁵ operations-support@mailman.egi.eu

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

The testing team will decide if issues are low, normal or priority. Depending on the criticality of the issues the team will decide the necessary action to be taken, 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 will be communicated, particularly when the changes are disruptive or contravene the usual behaviour.

4. Roadmap

Below in Table 1, the current status of the roadmap as of the date of this document. Further version of the roadmaps could be consulted on the EGI wiki for the JRA1.3 task²⁶. We will comment each of the tasks and their implementation details.

- Task 3.2.1, “Requirements gathering” is already complete and this document (written as task 3.2.2) is part of its output.
- Task 3.2.3 consists in developing a framework of best practices and migrated Python kernel code to support the migration of the rest of the codebase. RT Requirement #8823 is explained in the link as a front page that would expose some general statistics, and relevant as a good test for Task 3.2.3.
- Task 3.2.4 will be a continuation of the work done on the mockups in Section 2.3.2 to all the forms in the Portal.
- Task 3.2.5, the support of long running VMs is dependent on the actual solution adopted by the Accounting Repository team. Recently there was a meeting delineating two possible approaches for this.
- Task 3.2.6 depends on the AAI infrastructure that is finally decided by the RIs involved in the project.
- Task 3.2.7 consists on the release of the portal following the procedure in section 3.2, along with the documentation required and the graph improvements that are discussed in Section 4.2.
- Task 3.2.8 will be an evolution of the current interface for data sharing of the portal, mentioned in the introduction, and will also produce complete documentation for it.
- Task 3.2.9 will expand the graphs for section 3.2.7 with geographical support, so that data can be correlated with a map.
- Task 3.2.10 will implement a mechanism to retrieve analytical metrics from the accounting data to support decision making.
- Task 3.2.11 consists on the second release of the portal following the procedure in section 3.2, along with support to see the variation of the data in relation to a baseline.

²⁶ https://wiki.egi.eu/wiki/TASK_JRA1.3_Accounting

- Tasks 3.2.12, 3.2.13 and 3.2.14 depend on the concrete Accounting repository implementation, which is not finalized in this moment. As the rewrite will be effectively done in that timeframe, there will be time and resources for part of the design to be realized on these dates.
- Task 3.2.15 consists on the final release of the portal following the procedure in section 3.2.

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 <ul style="list-style-type: none"> • First/front page (New)²⁷ 	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

²⁷ <https://rt.egi.eu/rt/Ticket/Display.html?id=8823>

Task number	Task Name	Start Date	Release Date	Dependencies
3.2.7	D3.5: First release of the new Accounting Portal deployed in production <ul style="list-style-type: none"> · User Documentation²⁸ · Improve graphs visualization²⁹ · Support for EGI-Engage metrics³⁰ 	04/16	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	05/16	12/16	
3.2.9	Integrate data in maps for a graphical distribution of the information	07/16	01/17	
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 <ul style="list-style-type: none"> · Function of basic analysis³¹ 	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

²⁸ <https://rt.egi.eu/rt/Ticket/Display.html?id=8824>

²⁹ <https://rt.egi.eu/rt/Ticket/Display.html?id=8827>

³⁰ <https://rt.egi.eu/rt/Ticket/Display.html?id=8822>

³¹ <https://rt.egi.eu/rt/Ticket/Display.html?id=8821>

Task number	Task Name	Start Date	Release Date	Dependencies
3.2.14	Support Big Data tools	01/17	08/17	Repository implementation 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

5. Technologies

This section will provide a brief description of the technologies that will be used for the implementation.

5.1 Python / Django

Python³² 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 than 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 it uses white-space and tabulation as block delimiters (instead of e.g. braces “{}” in C-like languages), enforcing good formatting in the code, and contributing to its readability.

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.

Django³³ 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 SQLAlchemy.
- A Web templating engine and language that defines and serves 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.

³² <https://www.python.org/>

³³ <https://www.djangoproject.com/>

- 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 Portal³⁴, so it is a proven technology with hands-on experience.5.2 Dojo Toolkit

The Dojo Toolkit³⁵ 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 simplified, 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.

5.3 Bootstrap Library

Bootstrap³⁶ is an 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.

³⁴ <https://metrics.egi.eu/>

³⁵ <http://dojotoolkit.org/>

³⁶ <http://getbootstrap.com/>

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.