CoeGSS

Centre of excellence

D5.9 - Initial Portal Design

Grant Agreement	676547
Project Acronym	CoeGSS
Project Title	Centre of Excellence for Global Systems Science
Topic	EINFRA-5-2015
Project website	http://www.coegss-project.eu
Start Date of project	October 1st, 2015
Duration	36 months
Due date	31st March 2016
Dissemination level	Public
Nature	Report
Version	1.2
Work Package	WP5
Leading Partner	ATOS (F. Javier Nieto)
Authors	F. Javier Nieto, Burak Karaboga, Michael Gienger, Andrea Rivetti
Internal Reviewers	Marcin Plociennik, Andrea Rivetti
Keywords	Portal, Tools, CoE Services
Total number of pages:	40

The CoeGSS ("Centre of Excellence for Global Systems Science") project is funded by the European Union. For more information on the project please see the website http:// http://coegss-project.eu/

The information contained in this document represents the views of the CoeGSS as of the date they are published. The CoeGSS does not guarantee that any information contained herein is error-free, or up to date.

THE CoeGSS MAKES NO WARRANTIES, EXPRESS, IMPLIED, OR STATUTORY, BY PUBLISHING THIS DOCUMENT.

Version History

	Name	Partner	Date
From	**F. Javier Nieto**	**ATOS**	**11.02.2016**
First Version	**F. Javier Nieto, Burak Karaboga**	**ATOS**	**17.03.2016**
Second Version	**F. Javier Nieto, Burak Karaboga, Michael Gienger, Andrea Rivetti**	**ATOS, USTUTT, TOPIX**	**26.03.2016**
Reviewed by	**Andrea Rivetti, Marcin Plociennik**	**TOPIX, PSNC**	**30.03.2016**
Updated Version	**F. Javier Nieto, Michael Gienger, Burak Karaboga**	**ATOS, USTUTT**	**31.03.2016**
Approved by	**ECM Board**		**31.03.2016**

1. Abstract

The CoeGSS Portal is a key element for supporting the operations of the Centre of Excellence (CoE). It is expected to give access to several services and features, in line with the services to be provided at the CoE level. Therefore, this document analyses the requirements coming from the pilots and the services to be offered, in order to identify the features that the Portal will implement. In line with these features, this document describes the main architecture of the Portal and those functional blocks to be implemented for the first release. Finally, the document describes the deployment plans, so the Portal will become available for the CoeGSS consortium.

Table of Contents

2. Introduction

CoeGSS Portal is the front-end where the CoeGSS services will be delivered to the target parties and bring the HPC and GSS communities together through these service implementations. The aim of this document is to propose a set of initial features for the CoeGSS Portal, according to the inputs provided by WP4 (through the deliverable D4.1 [1]), by the Executive Centre Management (through the CoeGSS Services Profile) and by other project participants (mainly from WP5).

The document initially details the main portal features and the services focusing on the aspects of Dissemination and Community Building, Training, HPC Resource Access, Repositories and finally Software and Data tool requirements. High level architecture for the portal, the portal components, actors and portal-user interaction definitions are also explained in detail.

Following the initial development plan, an initial implementation design is provided which elaborates each individual component of the portal. Finally the document describes the portal deployment plan by depicting the general hosting concept of the project, foreseen implementations and the hardware required for hosting the CoeGSS services.

3. Portal Features and Services

3.1 Introduction

This section aims at providing a view of the requirements the CoeGSS Portal addresses, as well as the main features that we propose the Portal to offer to the consortium members and to other stakeholders in the future. Moreover, in order to show its usefulness, we have analysed how the Portal will support the services to be provided by the Centre of Excellence (under definition in WP2).

3.2 Requirements for the Portal

The Centre of Excellence (CoE) will provide some services and know-how to third parties, as well as to the participants of the Centre itself and, therefore, the CoeGSS Portal is expected to act as an enabler for providing/accessing those services and all those data gathered which can benefit end users. Although the Services Portfolio is still under definition by WP2, there are already some services identified which should be supported by the CoeGSS Portal:

- **Expertise Consulting**, seen as providing our expertise in GSS and HPC to our stakeholders;
- **Solution Consulting**, taking our stakeholders' problems and providing a solution to them;
- **Training**, so the stakeholders may gain some knowledge about the strengths of HPC and GSS combined;
- **Support**, so any stakeholder aiming at using our solutions will be able to interact with us for getting support when putting in place its solutions;
- **Knowledge building**, extracting lessons learnt from our experience and creating shareable knowledge from it;
- **Co-Design** SW / SW and HW / SW, meaning to facilitate the code optimization for solutions oriented to generating synthetic populations and simulations;
- **Repositories**, providing a centralized place where software and solutions can be found and accessed (both open and commercial);
- **Community Services**, providing our stakeholders with the tools they may need for their systems.

Moreover, although the work related to exploitation and sustainability of the CoE has just started, we have to take into account future needs related to the services provided from the CoE and other aspects that will influence the way to provide them (i.e. the existence of different business models for the various assets of the CoE).

On the other hand, it is clear that end users will have some needs related to Global Systems Science, stated by the pilots (as their representatives) in [1]. They need the CoE to provide some new tools for them (i.e. tools for generating populations, for simulations management, for visualization, for data analysis, for managing agent-based systems, etc.), so they will be able to generate new applications. But they also have more needs that need to be covered, such as training, access to execution resources, data and code collection tools, mechanisms for facilitating the testing, new datasets and possibilities to interact with other parties (from the GSS and the HPC domain).

While some of these requirements are not in the scope of the CoeGSS Portal but in the WP3 one (such as in the case of software for populations generation or Agent-Based Models management), it is feasible for the portal support the access to these kind of tools. On the other hand, there are other requests which are directly related with the features to be provided by the portal, as a tool they can use when implementing the pilots. This is the case of the data collection tools, training and the means to facilitate interaction with other stakeholders.

The following subsections provide a vision about the main features the portal should provide, according to the available inputs.

3.3 Dissemination and Community

End users (from WP4) have recognized the need to enable a close interaction between people of GSS and HPC communities, since CoeGSS aims at bringing together those domains for the first time. Even if it is a very innovative concept, it is a hard task because GSS applications work in a certain way and HPC requires applications to behave in a certain way in order to optimize their execution. Therefore, it is necessary that HPC community understands the new requirements coming from the GSS community, while the GSS community must understand the possibilities that HPC facilitates to them in terms of amount of resources available and efficiency.

This feature of the portal is in line with the CoE services of “Access to Expertise” and “Improving Excellence”, as they are related to knowledge transfer and stakeholders interaction. To be more concrete, the Portal will promote the role of CoeGSS as ‘ Knowledge / conference hub’.

3.3.1 Dissemination Means

We need to promote networking activities (conferences, workshops…) where all the stakeholders may take part and meet each other. Although the task of organizing events belongs to WP6, the Portal should be the place where any end user would be able to find any information regarding future and past events.

In that sense, even if the current website of the project is the main communication tool, at some point we assume that the Portal and the website should become the same asset for the CoE (fusing their features).

The future Portal could provide the means to access information about previous events, but it could also serve as the main platform for managing new events, providing features such as the publication of the agenda for each session, the registration of attendees, the management of payments (if any), the publication of the presentations done and other material, etc..

Of course, the Portal will provide access to any article, review, whitepaper or any other material which will be created by CoeGSS, in order to facilitate knowledge transfer.

Also, although the Portal will not articulate the collaboration with other European, national and local programmes and pan-European HPC-GSS activities, it will be the place where all the information related to those collaborations will be published.

3.3.2 Community Building

We should facilitate common spaces in the portal so people coming from the GSS and HPC environments will be able to agree on common terms and understand each other. Moreover, such community should support new end users when setting up their applications and environments, thanks to the experience of other members of the community.

Lessons learnt, potential issues, discussions about new features, success cases and announcements about new releases are some of the topics that the CoeGSS community should address. We can say they will represent the 'community services' that CoeGSS wants to provide to its stakeholders.

For doing so, it would be necessary to put in place several mechanisms, such as a wiki, a forum and any other tool which could facilitate interactions among members of a community (in a similar way as it happens in StackOverflow[1]). Such tools could be integrated with social networks, such as LinkedIn[2], so it would be possible to know more about the experts and they could increase their contact networks.

Functionalities provided in this section need to be made available in coordination with the CoeGSS website in order to avoid overlapping.

The website provides features in support of Community Building along three lines:

- internal, among project partners
- inclusive, among HPC and GSS insiders and experts
- public, aimed at engaging third parties and potential users interested in the services provided by the Centre of Excellence

[1] http://stackoverflow.com/

[2] https://www.linkedin.com/

The tools covering the overall strategy for CoeGSS are intended to engage potential partners and can be summarized as follows.

Offline community building:

- events, courses, conferences, etc

Online community building:

- for internal project partners
    - mailing lists (provided on HLRS infrastructure, already in use)
    - wiki space (provided on TOP-IX infrastructure, already in use)
- for the wider HPC and GSS communities and third parties interested in the services offered by CoeGSS
    - news section (provided on CoeGSS website, replicated on portal via RSS feed if needed)
    - newsletter (in progress, gateway on CoeGSS website)
    - knowledge base
        - project deliverables, articles and papers (provided on CeoGSS website)
        - technical documents, how-to's and FAQ about the services provided by CoeGSS (to be implemented on the portal)
    - feedback channels
        - questionnaires and forms about the project activities (to be provided on CoeGSS website)
        - questionnaires and forms about the portal services (to be provided on the portal)
        - ticketing system for CoeGSS services (to be provided on the portal)
        - user discussion forum about help on CoeGSS services (to be provided on the portal)

3.4 Training

It is necessary to provide training material for both HPC and GSS domains since, as mentioned before, this is a rather new combination of technologies which requires some guidance for new end users aiming at adopting the technologies provided by the CoE.

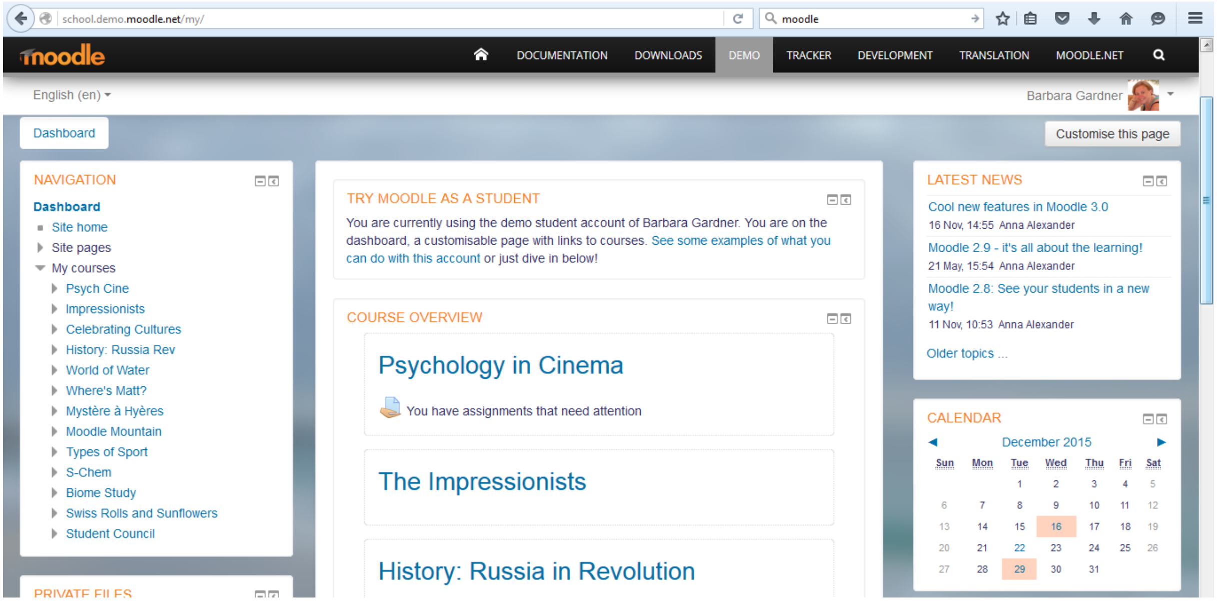
Features under this topic are very important, since they have been identified as elements with impact in the areas of ‘Access to expertise’, ‘Improving Excellence’ and ‘User Support’.

According to the requirements coming from the end users, training should offer not only guides and other material (i.e. webinars), but also practical works, so students may experiment with what they learn. They have highlighted the need to provide introductory and technical training in the HPC field, so they can learn how to prepare their applications to run in a HPC environment.

Feedback will be very important, in order to improve the published material. Therefore, the Portal should include mechanisms which will facilitate this task from the end users perspective.

One of the envisaged solutions is to integrate in the Portal one of the existing education platforms. There are many open source platforms out there, which may offer similar features, although some of them are more known and used than others.

According to our expertise, it seems that Moodle[3] could be an interesting option. It is generic, so it can cover most (if not all) the features we may need for the CoE, it allows for some customization and, moreover, it has a strong community behind. In any case, we will perform a deeper analysis about the tool to be used during the following months.



NAVIGATION

Dashboard
- Site home
- Site pages
- My courses
  - Psych Cine
  - Impressionists
  - Celebrating Cultures
  - History: Russia Rev
  - World of Water
  - Where's Matt?
  - Mystère à Hyères
  - Moodle Mountain
  - Types of Sport
  - S-Chem
  - Biome Study
  - Swiss Rolls and Sunflowers
  - Student Council

PRIVATE FILES

TRY MOODLE AS A STUDENT

You are currently using the demo student account of Barbara Gardner. You are on the dashboard, a customisable page with links to courses. See some examples of what you can do with this account or just dive in below!

COURSE OVERVIEW

Psychology in Cinema

You have assignments that need attention

The Impressionists

History: Russia in Revolution

LATEST NEWS

Cool new features in Moodle 3.0
16 Nov, 14:55 Anna Alexander

Moodle 2.9 - it's all about the learning!
21 May, 15:54 Anna Alexander

Moodle 2.8: See your students in a new way!
11 Nov, 10:53 Anna Alexander

Older topics ...

CALENDAR

December 2015

Sun	Mon	Tue	Wed	Thu	Fri	Sat
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		

Figure 1. Screenshot of a Moodle demo showing the student dashboard

We have to take into account the existing training services of the different partners which are part of the CoE. For instance, according to the D5.1 deliverable [2], both HLRS and PSNC offer training for those who want to use their HPC infrastructure. This implies that we need to look for the way to integrate these training offers in our Portal as well (i.e. announcing these training sessions and facilitating access to the information about them).

3.5 Access to HPC Resources

One of the key aspects of CoeGSS is the availability of HPC resources that would be used by GSS applications in order to run in an optimized way. The CoeGSS Portal should, somehow, articulate the provision of these resources whenever possible, facilitating the deployment of the GSS applications in the HPC infrastructures. This is what we relate to the idea of 'Service operation / hosting / on-boarding', one of the services in the portfolio to be offered from the CoE to its stakeholders.

Therefore, the way to support access to HPC resources can be twofold:

[3] https://moodle.org/

- Support the submission of formal resources requests
- Facilitate deployment in HPC infrastructures.

In the first case, since each HPC Centre (and PRACE) have their own way to request access to the HPC resources, the Portal will support this process by allowing the download of the required forms (such as for HLRS) or even linking the appropriate URL for accessing the HPC centre forms (such as in the case of PSNC).

On the other hand, the Portal will aim at supporting the easy deployment of CoeGSS tools in certain HPC infrastructures. Since the resources available for the project will be, mainly, provided by HLRS and PSNC, we will investigate the way to do this, in order to ease the interaction with these centres.

Another feature will be related to accounting information. Since the HPC centres can provide some information about this aspect (in principle, per groups, instead of per user), such information will be integrated in the Portal, as a way to facilitate accurate information about spent and remaining resources.

In line with this last feature, we consider it would be interesting also to explore the idea to ‘sell’ or facilitate the reservation of computation time in the Portal through the marketplace. Since it will be a difficult task, it is an option that will be analysed.

3.6 Repositories

In some cases, it is convenient to have available a place where developers and users can leave their code, so they can maintain, configure and test it in an easier way. In those cases in which the CoE is offering a consultancy service around some end user’s application, this tool is especially useful, so people from the CoE may manipulate directly the code, together with other involved stakeholders. Moreover, it is very important to facilitate code installation.

In the same way, it may happen that some experiments will need to use repeatedly the same basic datasets. Therefore, it would be very useful to have access to a common data repository which can be used by anyone.

Moreover, according to the main principles of services in the ‘Application Services’ area, repositories are necessary for some of the services to be offered from CoEGSS to its stakeholders. To be more concrete, there are several elements which are mentioned as potential elements in the services portfolio (‘unique and complete data base for all kinds of simulations’, ‘code repository’ and ‘solution repository’).

Therefore, the Portal will facilitate both software and data repositories, as a way to facilitate these features mentioned before.

3.6.1 Code Repository

A code repository is very useful due to several reasons (facilitate code management, deployment and testing, enable smooth interaction in consultancy services, etc…). Basically, this set of tools should be able to manage code releases and modifications traceability, to provide easy integration with typical development tools to automate testing and packaging, and to facilitate deployment and installation tasks.

Mechanisms such as SVN and Git have proof to be very effective and useful, so we will analyse which is the best solution to be integrated with the Portal, in order to provide features related to code management and traceability.

Moreover, another way to support development tasks is related to continuous integration and testing. Tools, such as Jenkins, are very useful, since they facilitate code compilation, integration, testing and even deployment with a few clicks.

Finally, in order to make even easier deployment/installation tasks, we will analyse the usage of tools such as Chef, Puppet and Ansible, since they provide very powerful features in this sense, just with some scripts.

These features and tools will be in line with the idea of ‘Compute performance for synthetic population code owners’, among others, since it will be possible to facilitate evaluation of tools by their developers.

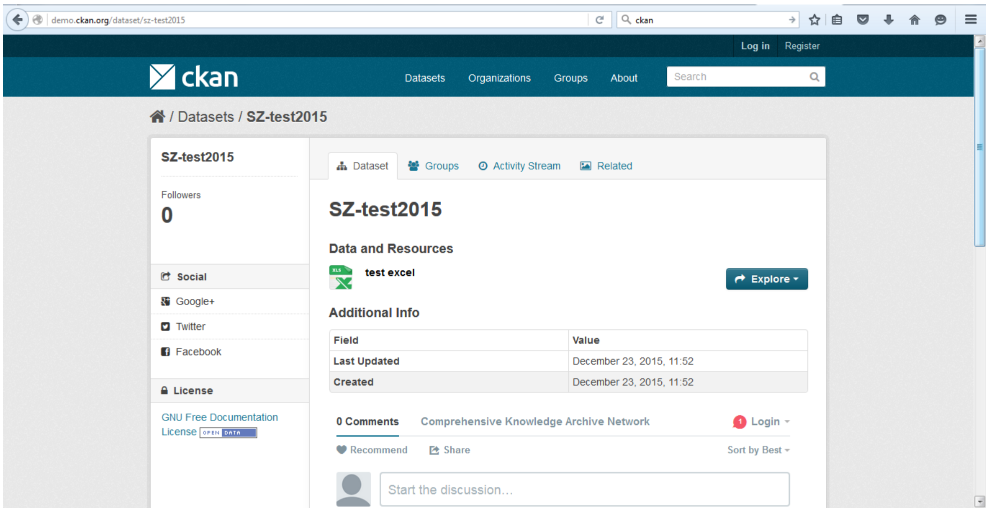
3.6.2 Data Repository

Common data repository will be used in order to share it between several experiments or even, at least, to have some base datasets which can be used for testing purposes. According to the users’ requirements [1], a data repository would be a really interesting tool (as they were requesting tools for data collection) and, therefore, several discussions have been carried out with respect to the kind of features that would be useful. They have stated the need for organizing the datasets, but also another useful features, such as identifying gaps in the datasets and a mechanism for pointing out which datasets could be complementary. These last features could be developed by WP3 and integrated later in the Portal, but the CoeGSS Portal must provide some basic tool first.

Data may have many different forms (PDF, XML, CSV, RDF, other customized formats, SPARQL endpoints, etc…). Therefore, we need a tool which will be able to manage this heterogeneity. Files can be simply left in a public folder available for users, but it is possible to provide a more complete tool for managing datasets. A tool such as CKAN[4] would be very useful in this case. It is a platform for managing datasets which allows publishing, accessing, and searching thanks to a metadata model based on DCAT [2], which includes information about the author, last updates, endpoints, licensing scheme, etc… There are also plenty of

[4] http://ckan.org/

CKAN extensions available which add features to the tool such as, allowing users to add comments about the datasets, providing statistics about the access to a dataset, etc…



/ Datasets / SZ-test2015

SZ-test2015

Followers
0

License
GNU Free Documentation License OPEN DATA

SZ-test2015

Data and Resources

test excel

Additional Info

Field	Value
Last Updated	December 23, 2015, 11:52
Created	December 23, 2015, 11:52

0 Comments Comprehensive Knowledge Archive Network

Figure 2. Screenshot of a CKAN instance

This platform can be used for exposing datasets and managing them, while other solutions should be given for storing the datasets themselves and for facilitating the upload to the concrete HPC Centre where data will be used. Moreover, all the available datasets should be published in a marketplace, in such a way it will be possible to find them and enable several business models around them.

3.7 Software and Data

GSS developers and users have defined a list of potential software tools they would need in D4.1. These are, mainly:

- Tools for data collection
- Tools for data adaptation/migration (especially, from social media)
- Software for generating populations
- ABM software for modelling the evolution of the system
- Tools for sensitivity analysis and calibration
- Visualization tools

It is also mentioned that a user interface should be provided for accessing these tools. Therefore, the Portal should provide all these software tools and data in such a way it will be easy to publish, find and access them. Moreover, we must take into account that these tools and data may be related to certain business models, so the Portal should provide a way to take this into account.

All these elements are in line with the ‘Application services’ mentioned as one of the main principle of services for the Centre of Excellence, especially in the area of off-the-shelf services.

Bearing in mind these requirements, the best solution is to provide a marketplace, where providers could publish their tools and data and, moreover, even complete systems already built with some of those tools (or with tools not published and already deployed in other infrastructure).

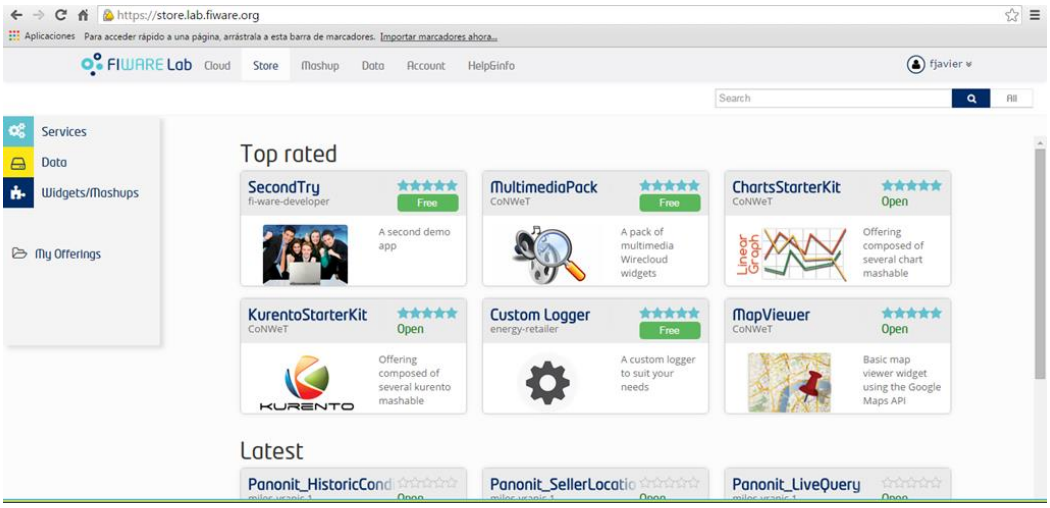
End users will be able to search for the tools their need and will be able to access information about those tools. In the case the business model is based on some pricing scheme, the Portal will allow users to buy and pay the software tools and data they are interested in (providers could also publish them for free for users).

There are some tools from the FIWARE[5] environment that could be used for these purposes: the WStore and the Registry. These tools provide a marketplace where items can be published and bought, managing payments whenever necessary. Since these tools support USDL [4], it is possible to specify, for each item, the associated business models to be used when users want to purchase an item.

In such an environment, items can be for free as well, and it facilitates the provision of additional information of each item, so end users will have a better idea about the tools available. It allows defining several categories, so it will be possible to have a category for datasets and other categories according to the kind of software tools that will be considered in CoeGSS.

It is also interesting to highlight the possibility to include additional information such as users' scores for each item, so other users will have access to opinions which will guide them through the selection of the tools they may need for their projects.

Additionally, it would be possible to synchronize our store with a global marketplace provided by FIWARE, reaching a wide group of potential stakeholders.



https://store.lab.fiware.org

Aplicaciones Para acceder rápido a una página, arrástrala a esta barra de marcadores. Importar marcadores ahora...

A second demo app

A pack of multimedia Wirecloud widgets

Offering composed of several chart mashable

Offering composed of several kurento mashable

A custom logger to suit your needs

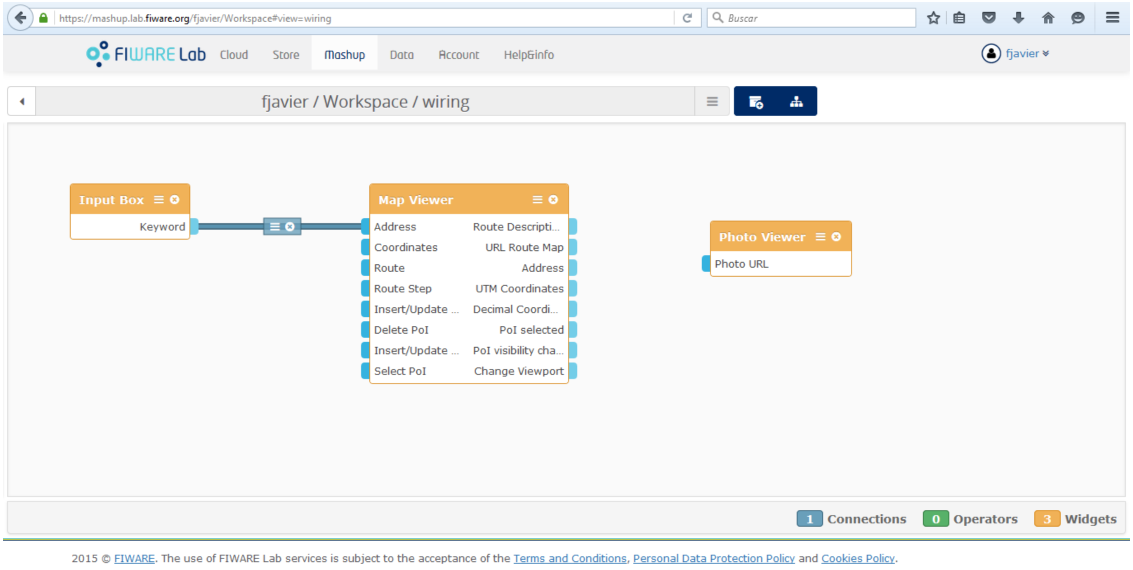
Basic map viewer widget using the Google Maps API

Figure 3. Screenshot of a WStore instance in FIWARE

[5] https://www.fiware.org/

After the access to tools and data is available, and according to section 5, the Portal might deploy the tools in the corresponding infrastructure.

Finally, it would be desirable to enable the possibility to combine different tools, as a way to create a complete system in an easy way. End users could decide to use certain dataset as the input for one of the tools for generating a synthetic population which, after that, is used as input for another tool modelling the expected climate, for instance. The possibility to facilitate a smooth integration between a chain of tools would be a really interesting functionality for stakeholders, since they could create new services easily or just do some testing with the tools they are interested in.



2015 © FIWARE. The use of FIWARE Lab services is subject to the acceptance of the Terms and Conditions, Personal Data Protection Policy and Cookies Policy.

Figure 4. Screenshot of a WireCloud instance in FIWARE

With this idea in mind, CoeGSS will explore the possibility to use solutions such as Wirecloud to facilitate this functionality. That FIWARE generic enabler allows creating mashups thanks to the usage of widgets and a very easy to use interface, where different software pieces can be linked graphically. It is already integrated with the marketplace so it could be used, although it requires tools exposed in the marketplace to provide a widget interface and a web service API.

3.8 Others

There are other features and services being offered from the Centre of Excellence which are not so easy to integrate with the Portal. This would be the case of consulting services for new GSSs, for instance. In that case, we envisage an analysis coming from WP2 in which different services and their business models will be identified. The Portal will aim at supporting those services by providing announcements, a description of the services and a way to communicate with the responsible (this could be related to a ticketing system, for instance, as a way to start the interaction).

The optimization of code is another service to be taken into account. Since there is no tool in mind for doing this, it could be provided by the typical services offered by the HPC centres belonging to the CoE. It could be also provided through some training material for the basic optimization any user could perform without strong support. It would be important to find the way to support CoE partners in publishing their already existing services in the Portal, as in the same way as it is done for those services offered by the CoE to its stakeholders.

In the initial CoEGSS portfolio, we envisage services such as 'Expertise consulting', 'Solution consulting' and 'Co-Design SW / SW and HW / SW', which must be supported by the Portal in the way mentioned above.

In line with this feature, the Portal should manage a registry of stakeholders of HPC and GSS (e.g. Centres, Code Owners, Service Providers, etc…) offering their expertise to customers, amongst other channels, through the CoeGSS portal (single entry point).

Finally, the Portal should host a career portal, where the centre members and stakeholders will be able to list vacancies, PhD positions, fellowships, etc… offered in the HPC and GSS fields, as a way to support professionals and entities. It would be interesting to explore its interaction with other social tools such as LinkedIn, where professionals have already some information available, in such a way they will not need to create and maintain a completely new profile in the CoeGSS Portal.

3.9 Mapping between CoE Services and Portal Features

The Centre of Excellence is defining the services that it will provide to its stakeholders. Although there is not still a complete list of services, WP2 has provided an initial list of services which are in line with the features of the Portal. The following table shows the mapping between the

CoE Services	Portal Features
Expertise Consulting	Stakeholders Yellow pages, Community Building, Training
Solution Consulting	Code and Data Repositories, Software and Data Solutions, Ticketing Service, HPC Resources
Training for several communities	Training, Community Building
User Support	Community Building, Ticketing Service, Training
Knowledge Building	Dissemination Means, Community Building, Training
Co-Design	Code Repository, Community Building, HPC Resources
Repositories	Data and Software Repositories
Community Services	Community Building, Software and Data Solutions, HPC Resources

Table 1: CoeGSS services and portal features mapping

4. High Level Architecture for the Portal

4.1 Introduction

In this section we have identified the main components which will belong to the first release of the Portal architecture. For doing so, we have followed a top-down approach with the following steps:

- *Identification of the functionalities:* Once we know the services to be provided by the CoE and the main features to be provided by the Portal, we have selected a set of features for the first release;
- *Identify the main components:* We have identified the main functional blocks of the Portal and the high level architecture which will articulate their interactions, in such a way that the Portal architecture will be flexible and modular;
- *Define interfaces:* Taking into account the features to be provided, the required interactions among components have been defined, identifying the external interfaces that each component must expose in order to facilitate the integration;
- *Detail the components:* The design of each component has been detailed, identifying its main internal elements and describing how the component will work internally.

As a result, the proposed architecture is designed for being built in a modular way, so it enables flexibility (in terms of substitution of components) and evolution of the Portal, so it will be possible to add more features in the following iterations and releases.

4.2 High Level Architecture

We have defined a high level architecture in line with the features we plan to implement for the first release of the CoE Portal. These features are the following:

- **Data management:** Since use cases will need several datasets for developing their systems, since the search of data sources is the first step in their process, this tool will be very valuable and will support them from the beginning of the project.
- **Training management:** HPC and GSS worlds are quite different and it is necessary that each community knows and understands the other one. Therefore, the provision of a training platform and some initial courses (especially from the HPC side) will be very useful from the beginning of the project, since it will provide end users the knowledge they need.
- **Community management:** It is expected that there will be doubts, issues, questions, etc… related to the tools to be used, how to use them, potential alternatives and other topics. A set of tools for the community management, facilitating the interaction among users from the HPC and GSS community will be a useful tool for solving those issues and progressing with the expected implementations.

According to those features, the following figure shows which are the main components already identified and how they are related.

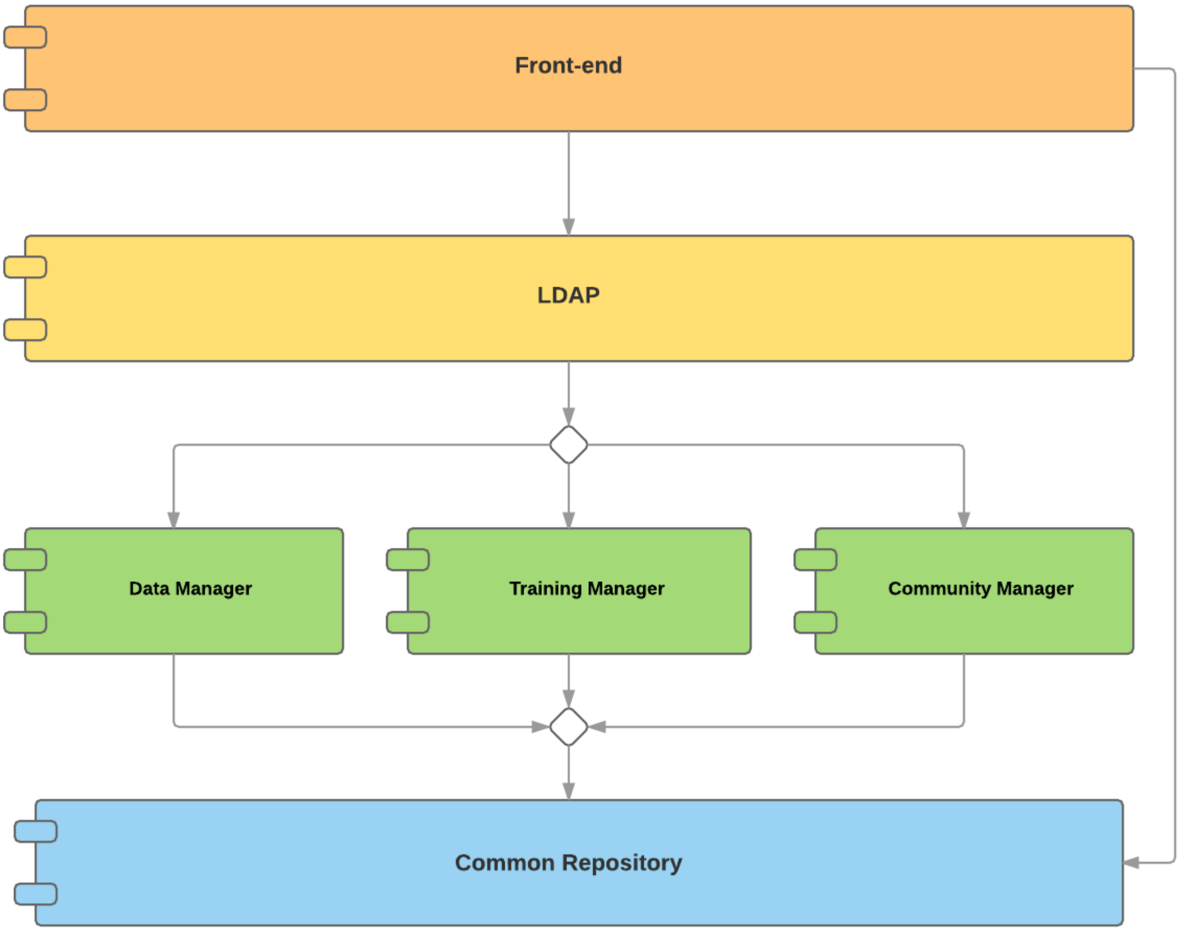


Figure 5. High level architecture of the CoeGSS Portal

The architecture has been built based on the idea that there will be a common single point of access for users to the different features (the Front-end), so it will be easier for them to navigate through the different tools they may need.

Additionally, since the Portal needs to manage users and the access to the functionalities, another component has been included (LDAP) in order to facilitate a common way to grant access to the different tools accessed through the Portal.

All these components are described and detailed in the following sections, also showing how they interact when implementing the different functionalities.

4.3 Components Description

The main components identified so far in the high level architecture of the CoEGSS Portal are the following:

- **Frontend:** This component aims at managing some interactions with the users, acting as an aggregator of the different features and facilitating access to them. It is the single point access from where any other component can be reached.
- **Data Manager:** This is the component which takes care of the management of the different datasets that the GSS applications may need. It performs management operations about data (add datasets, search for datasets, store datasets locally, etc…) as a way to facilitate access by the CoEGSS tools and by any user who may need to access data sources.
- **Training Manager:** Since it is necessary to keep a management of the different courses, webinars and other training material for the HPC and GSS communities, this component will take care of those features, enabling the possibility to manage the material and to access training courses created specifically for the CoEGSS purposes.
- **Community Manager:** This is the component which manages any kind of interaction among the different CoEGSS stakeholders. This means not only to provide forums where stakeholders may discuss, but also to provide access to a knowledge base for CoEGSS and to facilitate information about any dissemination activity that may occur in the context of CoEGSS.
- **Common Repository:** It represents a common storage solution for the Portal. It will be used for storing data that can be used by the different components and it will also store downloaded datasets.
- **LDAP:** This is the component in charge of the users' authentication operations. It can also provide some information about authenticated users, which can be used by other components. LDAP will be used as the centralized mechanism for authentication in the other tools, avoiding multiple and isolated authentication mechanisms.

4.4 Actors

These are the stakeholders we expect to interact with the CoE Portal at this stage:

- **Admin:** A user who has the total control of the tools. He has permission for creating/removing/modifying users, changing configurations in the different tools, controlling groups, etc…
- **CoE Contributor:** Somebody (partner) from the Centre of Excellence who is granted to provide content to the Portal, in terms of new datasets registration, new courses, news publications, knowledge base update, etc…
- **End User:** Stakeholders (both internal and external to the CoeGSS project) who want to use the tools we provide. They can access all the information available, the data management tools, the courses, etc… but they cannot modify the courses content or (for the moment) publish new datasets. They can participate actively in the community tools (i.e. posting in forums).

4.5 Interactions Definition

4.5.1 User Registration

The process of registering a new user is not very complex thanks to the usage of LDAP. When the Administrator wants to register a new user, he/she just needs to go to the Frontend component, fill in the form with the user information and submit the data.

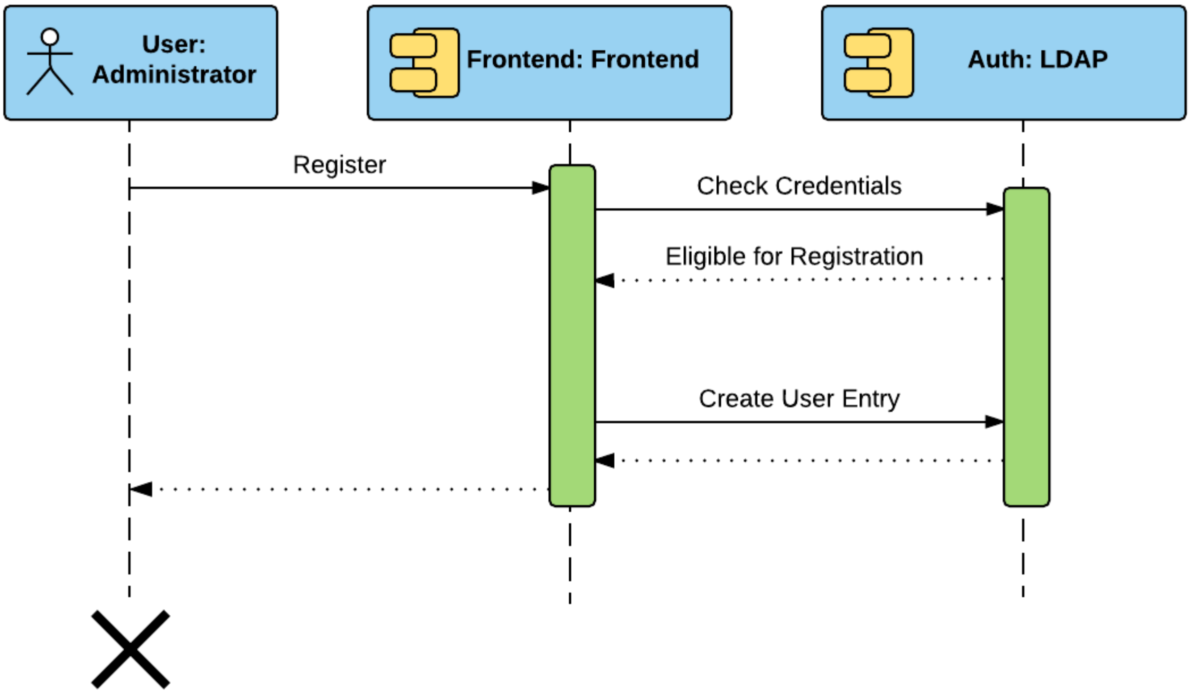


Figure 6. User registration interactions diagram

The Frontend will check the Administrator credentials in LDAP and, if accepted, it will send the gathered information to the LDAP component, creating a new user entry. From that moment on, the users will be able to access any featured provided by the Portal (according to their role), since the LDAP system manages the authentication mechanism for all the integrated tools.

4.5.2 Access Training Course

Accessing a training course is done through the frontend. The user provides their credentials and this information is sent to LDAP for authentication. Once the user is authenticated, he/she will be directed to the CoeGSS portal home page where CoeGSS tools can be accessed.

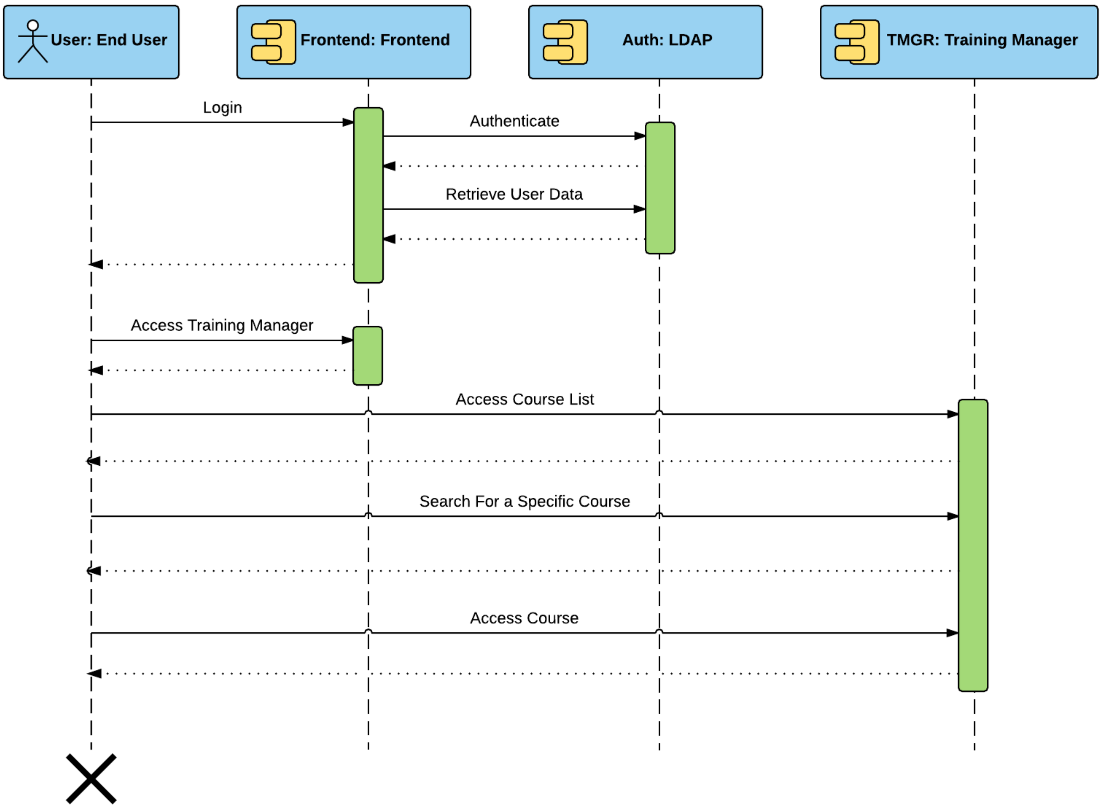


Figure 7. Access training course interactions diagram

Once the user has access to the training manager, a list of available courses can be requested. The user will be able to make a search for a specific course among the list of results and access a course of her/his preference.

4.5.3 Register a Dataset

In order to register a dataset to the system, the user has to be authenticated first by the LDAP server first by providing her/his credentials. After the authentication process the user will be able to access the CoeGSS tools through the portal home page.

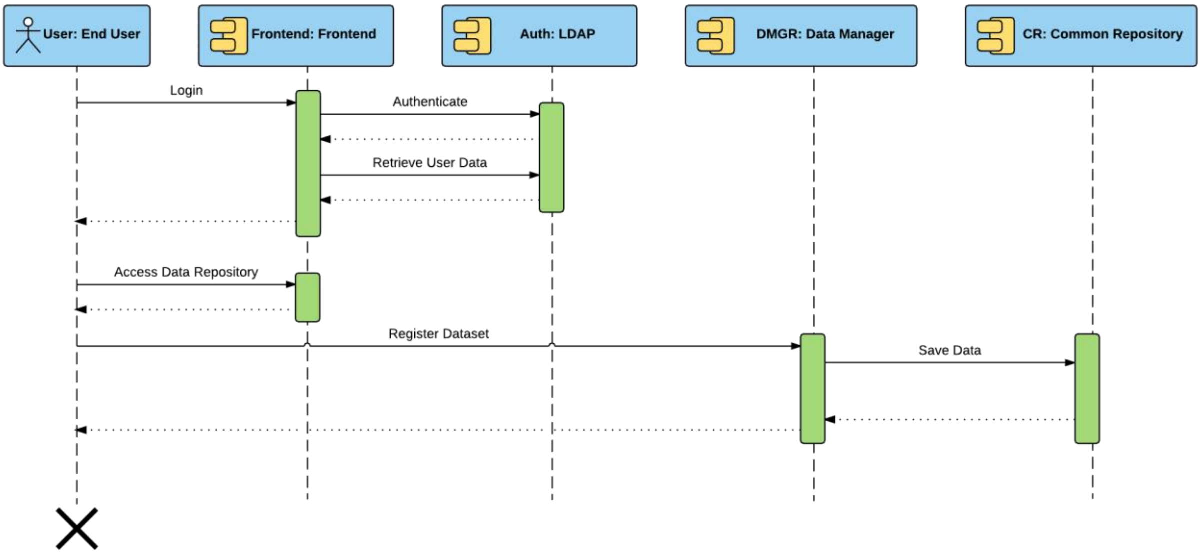


Figure 8. Register dataset interactions diagram

Once the user accesses the Data Manager, he/she can register a dataset by using the Add Dataset function, providing the required details of the dataset and uploading the data itself or providing the link to it.

4.5.4 Access a Dataset

In order to access a registered dataset, the user has to go through the authentication mechanism as in the previous section.

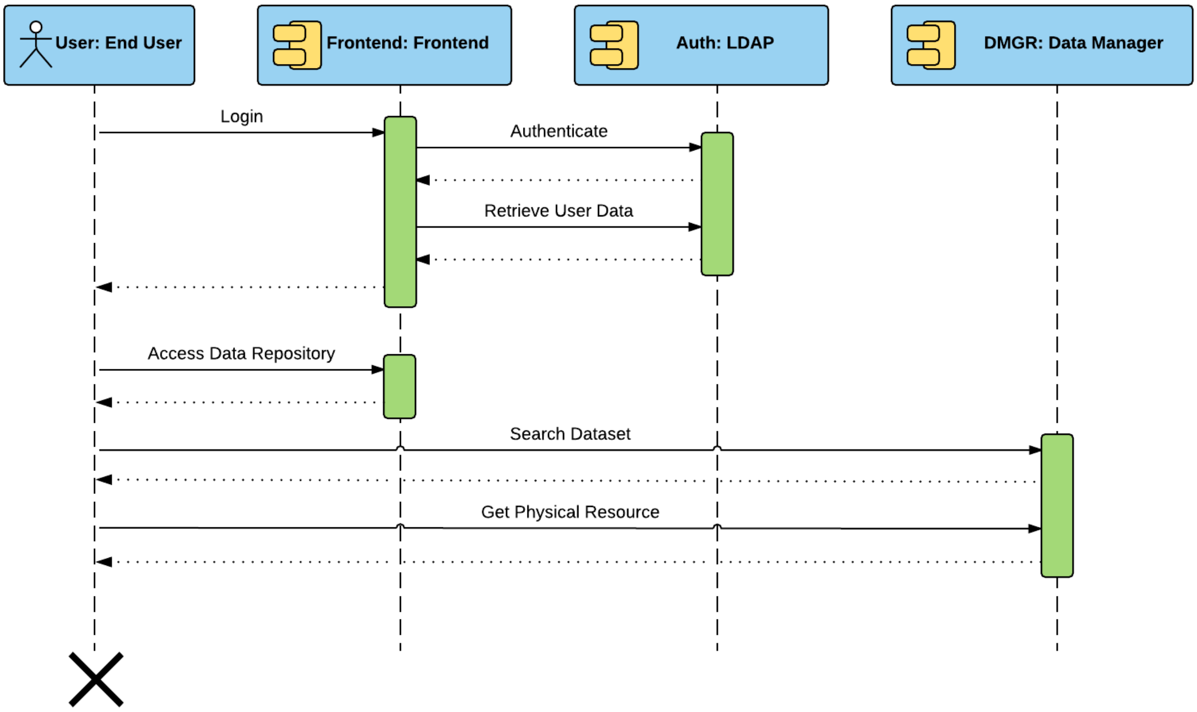


Figure 9. Access dataset interactions diagram

Once authenticated, the user will be able to search for a specific dataset or a group of datasets by using the search mechanism provided by the Data Manager. When the user finds the data he/she is looking for, the data will be available for download either through the external link provided or directly as a file from the Data Manager's local storage.

4.6 Initial Development Roadmap

Since the list of features identified for the portal is quite long, it is necessary to prioritize those ones which should be implemented and updated before others. Therefore, this subsection provides a list of the features to be included in each release, according to the working plan and to different criteria: how useful the features are, how complex their implementation may be and whether there are some dependencies which need to be fulfilled before the implementation.

In the first release (M9), the features to be implemented are the following:

- Common authorization mechanism, with an initial set of user properties
- First version of the frontend which will integrate all the tools and will facilitate users management
- Tools for data management, including the data repository, with an initial set of extensions for CKAN
- Training platform for the end users
- Initial set of tools for community management
- Cross-linking with the CoeGSS website

In the second release of the Portal (M20), the expected functionalities are:

- Update of the CKAN extensions in the data management tools, according to new needs or the availability of other features
- Addition of training material for the training platform (at least, two courses about HPC and GSS)
- Complete tools for community management
- Basic interaction mechanisms with HPC centres (retrieve information for users)
- Code repository and testing support tools (Jenkins)
- Marketplace and WStore for publishing tools, systems and datasets

In the third release of the Portal (M30), the expected functionalities are:

- Enhance interaction mechanisms with HPC centres as much as possible
- Populate the WStore with tools, datsets and those services prepared by the use cases

- Implementation of the process which will support the workflow for SIS (i.e. using WireCloud)
- Consultancy support tools (publication of services, ticketing tools, etc.)

In the fourth release of the Portal (M36), the expected functionalities are:

- Solve pending bugs
- Complete any pending development
- Experts yellow pages

5. Initial Implementation Design

5.1 Introduction

This section provides the detailed design of those components identified in the high level architecture, with the purpose of detailing the way they will be implemented. Each subsection describes the main features of each component, the internal components and the way the component works.

5.2 Frontend

This component is responsible for providing a single point of access for all other components that are implemented in the context of CoEGSS. The Frontend basically consists of several web pages, by using which the user will be able to, register and login to the CoEGSS portal, access CoEGSS components such as; the Data Manager, Training Manager, Community Manager etc.

Frontend is a simple component in terms of the number of entities it contains; **Frontend** itself which is the set of web pages the end-user will interact with and the **Database** which will be responsible for holding the basic log data and user information.

Frontend component's only direct relation with one of the other components of the system is the LDAP server. For each user registered to the system, Frontend will create an entry in the LDAP server and will use this information to authenticate and authorize the CoEGSS users.

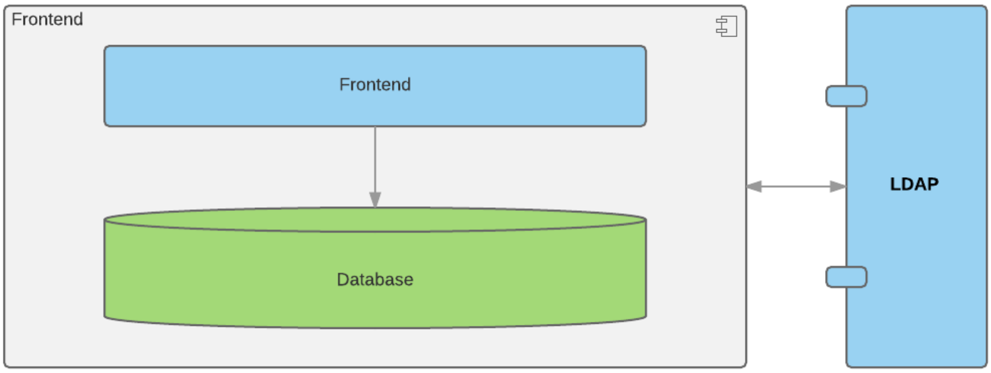


Figure 10. Detailed design for the Frontend

5.3 Data Manager

As mentioned before, this component is in charge of registering datasets, facilitating search of the datasets and providing access to the datasets. The selected implementation platform is CKAN, since it is a very mature platform, it provides many features for the data management, it provides metadata about the datasets based on standards and it allows for a lot of flexibility thanks to its extensions, which increase the available features and provide extra APIs and GUIs for accessing them.

Those authorized users will be able to navigate through the GUIs, doing searches and accessing specific information about the datasets. The proposed extensions are reflected in the webpage which shows the information about each dataset, so there will be menus and specific buttons for accessing the extra functionality.

Therefore, the detailed design of this component is divided in two main parts: one with the core CKAN platform and another one with the set of extensions we have selected for the CoE purposes, in line with the requirements of the end users. There is another main component which is the **CKAN Database**, which is the supporting storage for the CKAN platform and for some of the extensions.

Some external entities are included in the diagram as well, since they are important parts for the component, although they do not belong to it. The **External Resources** entity refers to those datasets which are exposed by other platforms or which are available at external storage solutions. The **Disqus** entity is an external platform which provides services for publishing comments, which can be integrated in many websites, platforms, etc. Finally, a **Disk** entity and a **Datastore Database** have been highlighted which are a part of the Common Repository, and they are necessary for storing datasets locally, in file format or in database format.

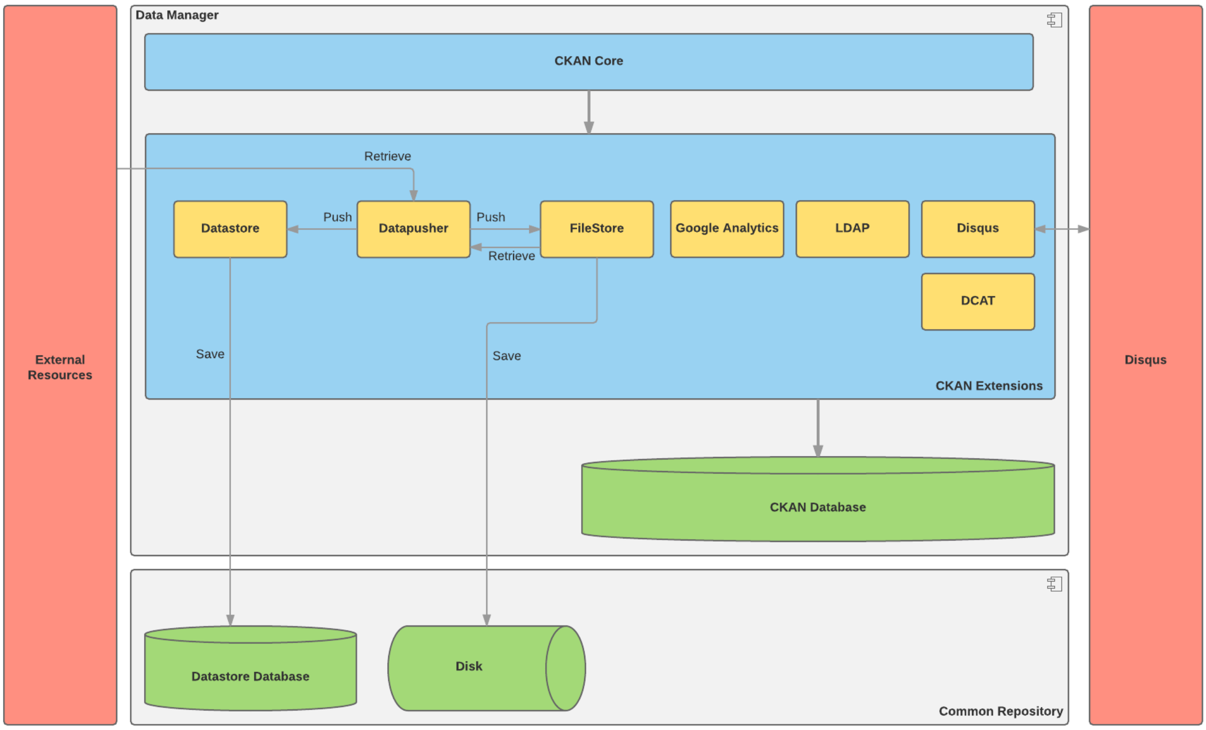


Figure 11. Detailed design for the Data Manager

The selected extensions for CKAN are the following:

- **Datapusher:** It is an interesting extension which enables the possibility to download a concrete dataset, so there will be a local copy of the data. It may use the Datastore or the FileStore extensions, depending on the format of the dataset, once the user decides to upload a dataset.
- **Datastore:** This extension allows automatically storing structured data in a database, in such a way it will be possible to send queries for navigating through the data and updating it.
- **FileStore:** This extension allows uploading files to the designed disk space. These files can be datasets or other files, such as logos.
- **Google Analytics:** This extension sends information to Google Analytics about the usage of the dataset and it also retrieves statistics from Google Analytics in order to show the information to the user.
- **LDAP:** This is a crucial extension which enables the possibility to use LDAP as the authentication mechanism for CKAN. Since we aim at using LDAP as a single authentication point, it is a key extension.
- **Disqus:** This extension allows users to publish comments about datasets, so it will be possible for others to access to the opinions of those people who used the data before, in order to determine whether the dataset could be useful.
- **DCAT:** It allows to publish in DCAT format the catalog of the current CKAN instance, and it enables as well de possibility to import the catalog of external platforms, by parsing its DCAT description, so it will be possible to include in the current CKAN instance information about the available datasets.

All these elements together conform the component, which will have a GUI and a REST API as well, since CKAN provides a rich programming interface that can be used for integration purposes.

5.4 CoeGSS Community Manager

This component will implement a section of the CoeGSS Portal aimed at supporting the community building activities by engaging potential participants.

These include potential stakeholders that are directed to the website/portal after being contacted pro-actively by CoeGSS either online or at live events (courses, conferences, etc) and users engaged through wider communication releases via the CoeGSS website, social media or partnering websites that are interested in following the project activities and participating to the community around it.

The portal needs to integrate the community building tools related to the services offered by the portal itself.

The services needed on the portal can be described in two parts with regards to their access restrictions:

Publicly accessible:

- **Knowledge base** containing a FAQ about the services provided by CoeGSS
- **Feedback channels** containing questionnaires and forms about the portal services
- **News section** which can include the RSS feed from the news section of the website which contains updates about the project activities and services and social media feeds

Registered users:

- **Knowledge base**
  - A repository of technical documents containing the technical documentation about the software tools the users need to be familiar with when accessing CoeGSS services
  - A repository of how-to documents detailing specific procedures users must follow in order to perform common or required tasks on the portal
- **Feedback channels** containing a technical discussion forum or a ticketing system as the main access point for assistance on the CoeGSS portal services.

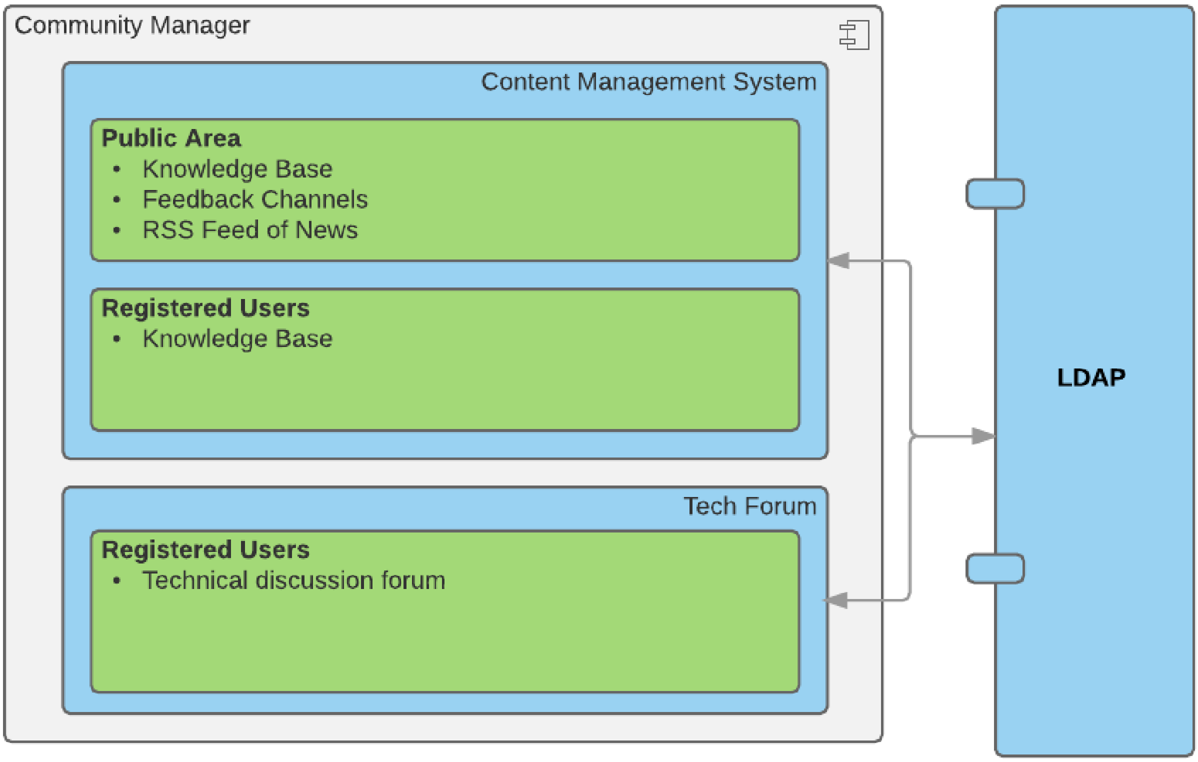


Figure 12. Detailed design for the Community Manager

A web discussion forum aimed at a technical community for finding answers to specific issues needs to provide the necessary usability features such as:

- web-based in order to make al content accessible and searchable
- tagging system to provide flexible categorization
- voting system in order to mark the most useful information

The forum needs to be monitored and moderated at least daily by technical staff and should be considered the primary source of support for the portal. As the community of users grows, a well maintained technical forum can provide a valuable source of information where most of the most common issues are already answered in details avoiding new interactions with tech support staff. It may also encourage active users to participate in discussions providing feedback and improving the overall quality of the answers produced.

Tech support people need to be allocated to review daily the issues that are open in the system and provide support for their resolution.

The content of the forum is also a valuable source of information regarding what users focus on when accessing the services of the portal and identifying the most common issues that need to be covered in the FAQ collection.

A number of solutions released as free software are available to implement a technical discussion system as outlined here:

- Question2Answer http://www.question2answer.org/
- LamCMS http://www.lampcms.com/
- OSQA http://www.dzonesoftware.com/products/open-source-question-answer-software

5.5 Training Manager

The purpose of this component is to provide the necessary tools for the experts to create and manage training content and offer an environment for the consumers of this content where they can manage their learning process, interact with their fellows and experts.

Moodle, which provides the set of required tools and functionality described above, basically comprised of three parts; Moodle Core, Moodle Extensions and the Database. Moodle Core contains the core functionality of the tool such as; course creation and management, user registration, enrollment, grading etc. The database component is the storage solution for the tool, containing the course, user, activity and all training related data. Moodle also offers numerous plugins which help to extend the platform, add new functionality or change the look and feel of the GUI. Here are the plugins selected for this version of the deliverable:

- **Configurable Reports**: This Moodle plugin enables the content creators to produce course, user, category or timeline reports without requiring SQL knowledge. An example for a report would be the information of users and their activities in a specific course. The plugin also features filters, pagination, logic conditions and permissions, templates support and exporting reports to XLS format.

- **Certificate**: This plugin allows the creation of PDF certificates/diplomas for the students of the course which are completely customizable (borders, watermarks, seals, grade information etc.). It also implements a verification mechanism for the certifications which is useful when a supervisor or administrator wishes to verify that the printed certificate is valid for that student.
- **Questionnaire**: Allows the teachers to create a set of questions to get student feedback on a course, an activity etc. The goal of this plugin is not to create a gradable item such as a quiz to assess the students but to gather feedback data about to course to be analyzed.

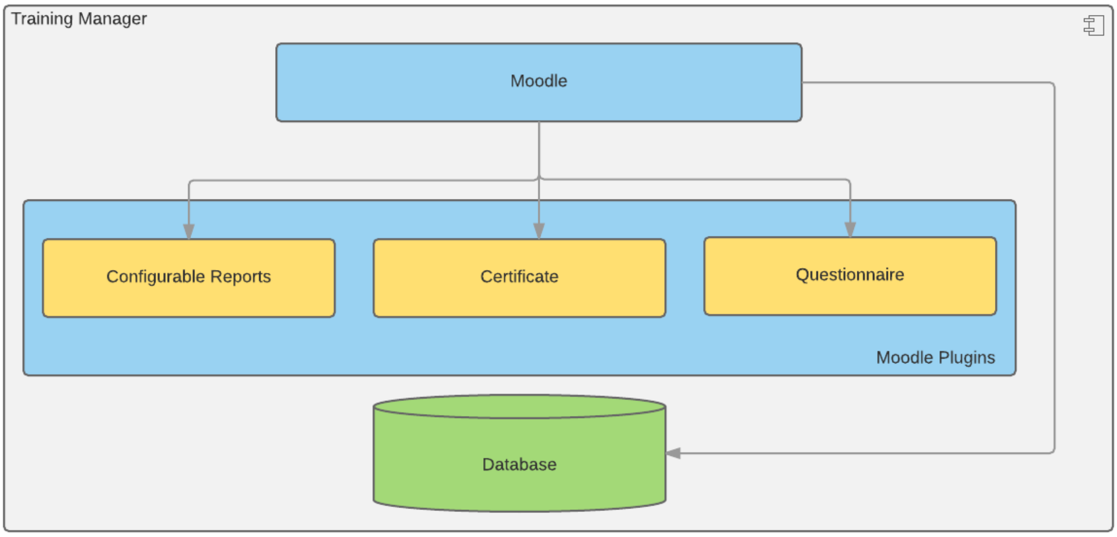


Figure 13. Detailed design for the Training Manager

5.6 Authentication Manager

As already highlighted within the sections and subsections before, the Portal relies on several, distributed components in order to control and separate the required functionality. This service-oriented approach simplifies management as well as hosting, but demands particular focus on security mechanisms, such as authentication and authorization for the various software artefacts.

In order to enable authentication mechanisms for all kinds of components, the state of the art LDAP protocol will be used to regulate access to the CoeGSS Portal services. LDAP is well adopted and most of the Linux distributions already provide common packages, so that both, installation and configuration are standardized. In addition, applications provide interfaces through native implementations or plugins to common LDAP servers, which simplify the integration significantly.

Since all services rely on independent base technology, seamless integration of the authentication and authorization mechanisms is mandatory to reduce the overall development overhead and furthermore, guarantee stable interaction between the services. Therefore, a single LDAP instance to manage the user database with its different kinds of information that is connected to all relevant CoeGSS Portal services will be established. In order to manage the users, the following information scheme is foreseen:

- **Name**
  For legal issues, the full name of the CoeGSS user will be stored within the LDAP database.
- **Nationality**
  High Performance Computing systems underlie various regulations, such as nationality of the users, amongst others. Nationality may also be important for data base access, so that this attribute needs to be implemented.
- **Email**
  In order to contact CoeGSS users, a valid email address is needed for getting access to the systems.
- **Phone**
  In addition to the email address, a valid phone number is required as well.
- **Organization**
  The organization of the user is an important attribute, especially for industrial cooperation.
- **Address**
  Users need to provide their full address details, like street, city with zip code and country.
- **PayPal account details**
  In order to develop a sustainable system, payment of services needs to be taken into account. Therefore, PayPal account details will be stored within the LDAP database.
- **Role**
  The role of the user is important to determine the access regulations. This can be handled in an automated fashion, as standard users will receive standard access and user rights. However, system administrators and moderators of services require different access rights to manage the components.
- **CoeGSS account details**
  Finally, for ease of use, CoeGSS users will be able to choose their username and password freely.

All this information will be transferred into a LDAP scheme, which will build the base for the overall hosting environment. As a consequence, all services will be able to authenticate against this centralized service.

6. Portal Deployment Plan

6.1 Introduction

Within this sub section of the document, the infrastructural challenges and initial ideas and solutions are highlighted. At first, the general hosting concept of the CoeGSS project is described. In section 6.3, the CoeGSS services and their foreseen implementations and technologies are depicted. And finally, the hardware for hosting the services is characterized and a schematic view showing the production hosting environment is presented.

6.2 Deployment concept

The overall setup for the hosting environment of the CoeGSS Portal services relies on both, a global and a local concept. Globally, two physical stages for operating and developing the services will be established, so that production operation is completely separated from the development infrastructure. This approach guarantees a maximum of performance, flexibility, operation ability and security for all kinds of users: the always stable production stage can be used by project partners or third party users whereas developers can test and improve the future packages on a completely separated stage. So even if development or integration is more complex than expected, the users on the production infrastructure won't recognize any problem.

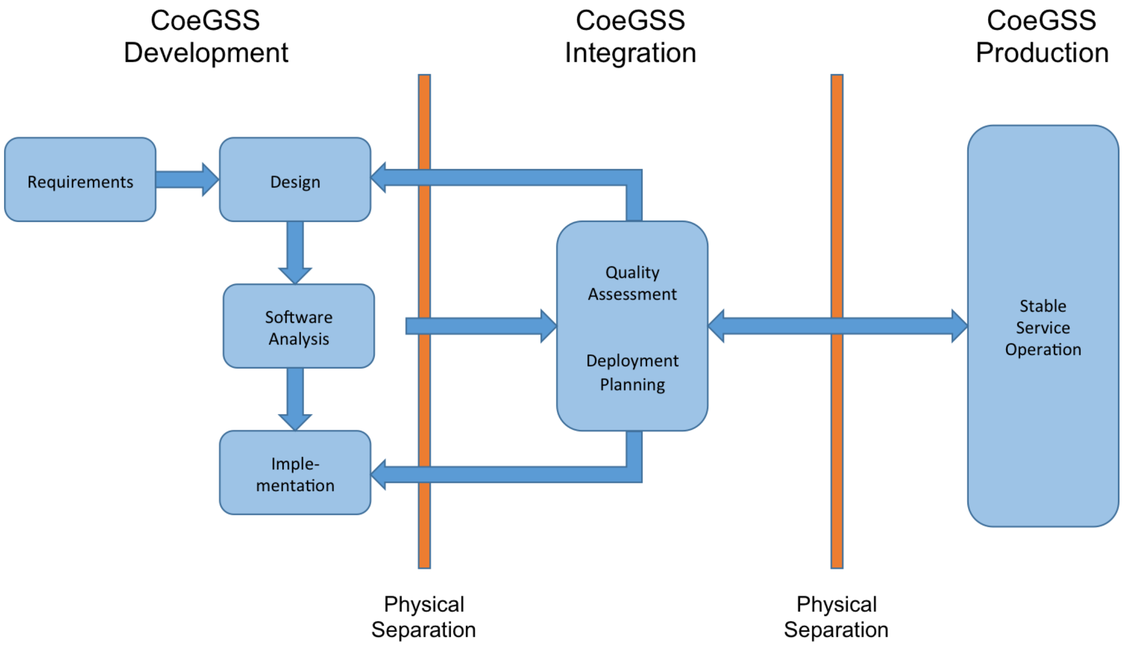


Figure 14: Staging concept of the CoeGSS project

As can be seen in Figure 14, the development chain consists of three completely separated steps: local development, the integration stage and the actual production stage. The process for deploying and deploying software is as follows:

- **Develop software on local workstations**
  Whenever software needs to be designed, implemented and tested, local developer workstations will be used for this purpose.
- **Deploy and test software on the integration stage**
  After the software components have been designed and implemented and the developer confirms functionality, the software packages will go through automated functional and integration tests on a tool like Jenkins. Once all the tests are successful, the software component will be compiled, packaged and get a new version number to be deployed on the centralized integration infrastructure. The deployment process will be handled automatically by tools like Puppet, Chef etc. Sub-releases of software components will only be limited to extended functionality and/or bug fixes. Major changes to the software containing new functionalities will be a part of the main releases.
- **Operate stable software in the production infrastructure**
  If evaluation and validation of the components pass the quality assurance process, software will be continuously integrated in the production infrastructure.

It is important to mention that integration and production stage don’t need to be duplicated completely. In particular, for integration software testing is aspired, whereas production infrastructure targets resilient operation only. However, most important for the overall process is the quality and integration feedback of the integration and production infrastructure. Only with this feedback, component functionality and interaction can be improved in order to guarantee a stable and mature production infrastructure.

In contrast to the global concept, the local approach defines the hosting environment in a fine-grained fashion. All required services are targeted, reaching from automated software management tools, up to monitoring, software testing and the actual CoeGSS Portal service operation itself.

As detailed above, the CoeGSS Portal service contains various components with different requirements such as disk size, central processing unit (CPU) performance, network capabilities or the amount of main memory. In order to satisfy all requirements in an appropriate and efficient manner, hosting of services is targeted within virtual machines (VMs). For service-oriented architectures such as the CoeGSS architecture, this concept is very beneficial since it offers well-defined administrative capabilities such as efficient operation of the services, flexibility and performance adaptation as well as enabling easy backup and migration capabilities.

Besides the benefits above, encapsulation of services is very important: different requirements can be addressed in a resource-sparing fashion, which results in lower costs for the overall operation.

Reflecting all services of the CoeGSS project, various service requirements have to be taken into account and approached in a general and extendable architecture. For this purpose, a detailed hosting concept has been developed that contains all required services and their deployment. This information can be obtained in the following subsection.

6.3 Components Deployment

Although the first CoeGSS Portal release cannot focus the development of all kinds of services, the deployment plan needs to consider all the available requirements and capabilities. For the hosting environment, the following services and particular components have been identified:

- **Automated software management**
  In order to transfer software artefacts between the integration and production infrastructure, automated installation and configuration procedures need to be in place to guarantee software functionality and minimized downtimes of the systems.
  It is intended to use Puppet and Foreman for this purpose.
- **System monitoring**
  Stable and failsafe systems require monitoring of services, in particular their availability and their performance have to be observed.
  It is intended to use Nagios in combination with Monit to guarantee availability of services.
- **System management**
  For system access, different tools and protocols like the Dynamic Host Configuration Protocol (DHCP) or the Domain Name System (DNS) are required.
  It is intended to use the standard Linux tools and mechanisms for this purpose, various functionality is already implemented in the available infrastructure (see section 6.4).
- **Software repository**
  The manifold components of the CoeGSS hosting environment need to be accessible. For this purpose, a software repository is required.
  It is intended to use the Sonatpe Nexus repository manager to offer software packages to the infrastructure.
- **LDAP and Certificate Authority (CA)**
  For user authentication, a centralized authentication server needs to be provided. For this purpose, the state of the art LDAP protocol will be used. In addition, to secure communication, a dedicated CoeGSS certificate authority will be established.

It is intended to use the available OpenLDAP and Certificate Authority packages of a Linux distribution.

- **Software testing**
  As already detailed, quality of software and stability of services are important requirements for the CoeGSS project. Therefore, software mechanisms to build and test software packages automatically need to be developed.
  It is intended to use Jenkins in combination with Puppet and Nexus for this purpose.
- **Data repository and CKAN**
  Data sources need to be publishable, so that dedicated upload functionality for the users have to be provided.
  It is intended to use the CKAN software for this purpose in combination with standard Linux functionality.
- **FIWARE component WStore**
  The Wstore component allows users to publish services and data they want to offer to the public, so any end user may decide to buy it (or take it for free, depending on the business model) and deploy it in the FIWARE infrastructure.
- **FIWARE component Marketplace**
  The Marketplace component serves for providing a single access point for a group of WStore instances.
- **FIWARE component WireCloud**
  If end users want to combine several services and data, thanks to a solution based on widgets, WireCloud enables the possibility to put several services together easily (with drag and drop) and combine their inputs/outputs, in such a way they will work together in a dashboard. It is already integrated with the Marketplace component, so only those entities owned by the users will be available.
- **Training services**
  For hosting and managing the CoeGSS training material, interactive workshops and video presentations, a dedicated system is mandatory.
  It is intended to use Moodle for this purpose.

In total, the CoeGSS Portal is comprised of 11 individual services, which need to be translated into the virtual hosting environment. The straightforward way for deploying the services results in one service per single VM, which will be highly inefficient due to the introduced management overhead. Therefore, an approach of co-locating services on the available VMs has been chosen that is detailed in the following Table 2.

VM	CPU	Memory	Disk	Services
Management	4 Cores	8 GB	30 GB	Software management System monitoring

				System management
Security	1 Core	1 GB	10 GB	LDAP and CA
Repositories	1 Cores	2 GB	30 GB Network File System (NFS)	Software repository Data repository and CKAN
FIWARE	8 Cores	16 GB		WStore Marketplace WireCloud
Training	2 Cores	8 GB	30 GB Network File System (NFS)	Moodle
Testing	2 Cores	4 GB	30 GB	Software testing

Table 2: CoeGSS service distribution and requirements

Given the service distribution in Table 2, in total 18 CPU cores, 39 GB of main memory and 130 GB of disk space are required to host the 11 individual services in an initial configuration. The described setup is not fixed, it represents and initial attempt to distribute the services in an efficient manner. Depending on the usage, the performance and the complexity of the infrastructure, the allocation may change to offer the best possible setup for administrators and end users.

The above-mentioned service distribution will act as a baseline for providing the first setup of the portal release. For deploying the release components, the Management VM, the Security VM, the Repositories VM and the Training VMs take precedence, as those are required to operate the first portal release. In this first deployment phase of the project, only the production infrastructure will be used, since no software components are running at all. Nevertheless, after the release, the integration infrastructure will be installed on basis of the first release to support further developments.

6.4 Available Infrastructure

Within this last subsection, the available infrastructure for hosting the CoeGSS services will be described. As already detailed, the services will be hosted in virtual machines that are deployed on physical hosts. A detailed view on the physical infrastructure for hosting the production stage at HLRS can be obtained in Table 3. In general, three hosts will be used to deploy the services, however, an additional Cloud hosting backend is available on demand with more than 1.000 additional CPU cores and 2 TB of main memory. All the physical hosts operate the Xen hypervisor to enable virtual machine deployment, for managing the virtual appliances Libvirt and OpenStack are used.

Capabilities	**Service Node 1**	**Service Node 2**	**Storage Node**

CPU	16 @ 2.600 MHz	24 @ 2.600 MHz	4 @ 2.200 MHz
Memory	24 GB	64 GB	8 GB
Network connection	1 Gbit externally 10 Gbit internally	1 Gbit externally 10 Gbit internally	10 Gbit internally
Storage	1 TB 4 SATA disks	2 TB 4 SATA disks	12 TB 16 SATA disks

Table 3: Hosting infrastructure capabilities

The schematic CoeGSS production infrastructure is detailed in Figure 15. All nodes make use of an internal 10 Gbit and an external 1 Gbit connection. For hosting the public CoeGSS Portal services, a subnet of 16 Internet Protocol (IP) addresses is dedicated to the project.

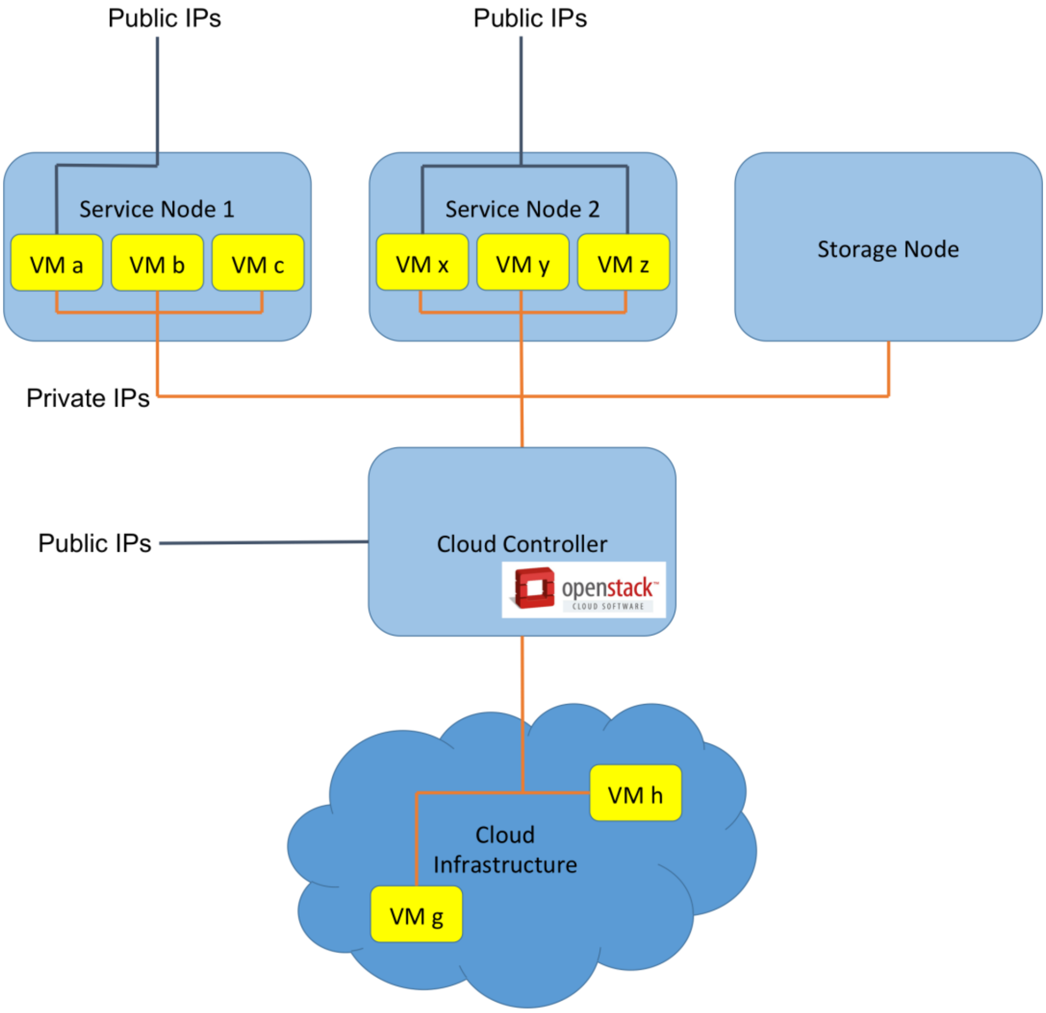


Figure 15: Schematic CoeGSS production infrastructure

7. Summary

This document described the CoeGSS portal features and services, the high level architecture for the portal, the defined roadmap for the different releases, initial implementation design and the portal deployment plan. For the first release, we decided to provide three main features, due to their potential usefulness for the pilots: Data Management, Training and Community Building.

The next release of this deliverable (to be released at M9, together with the first release of the CoeGSS Portal) will introduce new portal components in line with the proposed roadmap (the code repository, the marketplace and the mechanism for interacting with HPC centres) and their implementations while iterating the high-level architecture and implementation designs to their next versions in order to realize the features and services described in this document.

References

[1] C. Consortium, “First Report on Pilot Requirements,” 2016.

[2] C. Consortium, “Definition of the CoeGSS Operation Environment,” 2016.

[3] F. Maadi, J. Erickson and P. Archer, *Data Catalog Vocabulary (DCAT),* W3C Recommendation, 2014.

[4] K. e. a. Kadner, “Unified Service Description Language XG Final Report,” W3C Incubator Group, 2011.

[5] “Adapting Hadoop to HPC Environments,” [Online]. Available: http://www.hpcwire.com/2014/02/14/adapting-hadoop-hpc-environments/.

List of tables

List of figures

List of Abbreviations

DoW	Description of Work
EC	European Commission
EGI	European Grid Infrastructure
CoeGSS	Centre of Excellence for Global System Science
ESFRI	European Strategy Forum on Research Infrastructures5
HPC	High Performance Computing
HPDA	High Performance Data Analysis
ToR	Terms of Reference
WP	Work Package