

D5.11 - SECOND PORTAL RELEASE

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	May 31 st , 2017
Dissemination level	Public
Nature	Report
Version	1.1
Work Package	WP5
Leading Partner	ATOS (F. Javier Nieto)
Authors	F. Javier Nieto, Burak Karaboğa, Michael Gienger, Yossandra Sandoval, Michal Palka, Marcin Lawenda, Paweł Wolniewicz
Internal Reviewers	Andreas Geiges, Paweł Wolniewicz
Keywords	Portal, Tools, CoE Services
Total number of pages:	30

Copyright (c) 2016 Members of the CoeGSS Project.



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/](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

Version	Name	Partner	Date
From	F. Javier Nieto	ATOS	04.05.2017
Initial Template	Burak Karaboga	ATOS	18.05.2017
First Draft	Burak Karaboga, Marcin Lawenda, Michael Gienger, Michal Palka, Pawel Wolniewicz, Yosandra Sandoval	ATOS, CHALMERS, HLRS, PSNC	23.05.2017
First Version	Burak Karaboga	ATOS	27.05.2017
Reviewed version	Burak Karaboga	ATOS	31.05.2017
Approved by	ECM Board	ALL	31.05.2017

Abstract

The second release of the CoeGSS Portal introduces several changes to the existing system and adds some new features and components. Following the document structure of the previous version, this document provides detailed information about the current state of the portal; its components, their deployment and configuration details. For the sake of coherence and readability, this document aims to highlight the new features and changes to the previous version of the CoeGSS portal while keeping a summary of the text from the preceding deliverable and referring to it where necessary.

Table of Contents

- 1. Introduction 5
- 2. Adapting the Portal Roadmap 6
- 3. Implemented Portal Architecture 9
- 4. Frontend 11
- 5. Authentication and Authorization..... 17
- 6. Data Manager 20
- 7. Training Manager..... 23
- 8. Community Manager 24
- 9. Infrastructure Services 25
- 10. Future Work..... 27
- 11. Summary..... 28

References 29

List of tables 30

List of figures 30

List of Abbreviations..... 30

1. Introduction

Following the design details described in [1] and the new roadmap plan, the CoeGSS team has been working on the new features and enhancements to the existing components of the CoeGSS Portal. The first action performed was to analyse whether the existing roadmap was appropriate to the context of the project, proposing a new one as the business models were available [3]. The work towards the second release has been focused on the implementation of the new features such as the Yellow pages and the Matchmaking tools, enhancements to CKAN functionality and the authentication and authorization mechanism of the CoeGSS portal to provide a single sign-on (SSO) mechanism. This release also introduces visual changes to these existing services to improve the visual integrity of the portal.

This document describes the CoeGSS Portal implementation, detailing several aspects about such implementation with the objective of facilitating the understanding about the implementation done and of acting as a guideline for those who may want to deploy the Portal components, in order to have another instance of the Portal or to set up another independent Portal.

In order to do so, Section 2 describes the current roadmap and the reasons for such changes. In the section 3, the document describes the implemented features and remembers the followed architecture, while Sections 4 to 8 provide information about the components, their implementation, their configuration and their testing. Section 9 focuses on those additional services provided from the infrastructure perspective (such as monitoring), aiming at supporting the Portal availability. On section 10, the document focuses on the future work and the plans for the next release of the Portal. Finally, Section 11 summarizes the document and provides some conclusions.

2. Adapting the Portal Roadmap

2.1 Analysis of the Sustainability Model

The CoeGSS Portal has always been seen as a tool for supporting the whole project, especially considering its role with respect to exploitation and sustainability. Taking a look at the new document for defining the sustainability strategy (D2.1 – Stakeholder Context and Initial Sustainability Model [3]), we have identified the need to re-think the roadmap of the Portal, so its features will be released in such a way that they will support the different business models as soon as possible.

The sustainability model [3] identifies that some horizontal services are provided through the Portal, although it was not possible to do it with the first release of the Portal at all:

- **Community building** → We were missing, mainly, content (i.e. news, discussions...);
- **Yellow pages** → This feature was planned at the end of the project;
- **Trainings** → We were missing content, but the tool was already there;
- **Code and data repositories** → We were missing the code repositories;
- **Software and data solutions** → The adapted Pandora was available, but still more software was missing (under development);
- **Co-design** → Although the knowledge base is a good start, it was necessary to think about more ways to support it;
- **GSS matchmaking** → This feature was not considered in the initial plans as such, although a kind of similar approach was discussed related to the consultancy tools.

For some of the services expected to provide from the Centre of Excellence, this is the way the Portal could support them:

- **Infrastructure provision** → HPC as a Service (under development);
- **Software and data provision** → Data was available and software was under development;
- **Simulation provision** → Adapted Pandora was there, but not managed through the Portal, and by the release of the second version, more software would be available;
- **Information provision** → There were a few missing pieces for supporting this service;
- **Consultation provision** → The Portal was missing some important pieces (i.e. matchmaking and yellow pages).

2.2 Mapping with the Proposed Business Models

Going through the list of 17 proposed business models, we did an analysis in order to understand how those business models mapped with the Portal features and which pieces were missing for each case. The main purpose of this exercise was to identify those business

models that could be fully supported by the Portal just after the first release, with the objective of identifying new priorities for future developments.

#	Name	Missing Pieces
1	ADVERTISING MODEL	Blogs, Yellow pages, link to publications (website), Training content
2	THE OPEN SOURCE MODEL	Consultancy tools, Software repository, GSS Software, Marketplace
3	THE FREEMIUM /PREMIUM MODEL	Applications, Simulation software, Workflow implementation
4	ALL INCLUSIVE PACKAGE	Applications, Simulation software, Workflow implementation, Information
5	SUBSCRIPTION TO PORTAL	HPCaaS, Simulation software
6	SIMULATION-AS-A-SERVICE	HPCaaS, Simulation software, Workflow implementation
7	THE FRANCHISE	Most of services are missing
8	THE INCORPORATION	HPCaaS, Software, Marketplace
9	THE PROSUMER	Community full development and many other services
10	THE PURE CONSULTANCY	Consultancy tools, Software repository, GSS software
11	THE BROKERAGE	Yellow pages, Full community support, Matchmaking
12	HPC COMPUTING RESELLER	HPC-aaS
13	THE GSS DATASETS	Complete data management
14	THE VALUE ADDED RESELLER	GSS software, Marketplace
15	SELF SERVICE PORTAL "DO IT YOURSELF"	GSS software, Marketplace
16	ONBOARDING HPC	GSS software, Marketplace
17	SELF SERVICE "MARKETPLACE" FOR ISVs	GSS software, Marketplace

Table 1. Business Models and Portal Features

According to the table, it was possible to understand that some business models could not be supported for the moment, mainly because they depended on the provision of some GSS software, which was not available yet, and it would not be available just after the second release of the Portal (as there were some developments on-going). That meant that the Portal developments should be focused on other aspects that may enable some of the listed business models.

This was the case, for example, of the Advertising Model, the Brokerage, the HPC computing reseller and the GSS datasets. In some cases, on-going Portal developments would facilitate the provision of the required services (i.e. data management), but in other cases, it was necessary a modification in the Portal roadmap, so we would be able to focus developments on more useful features (i.e. improvements to some of the existing tools and developing new tools such as matchmaking, Yellow Pages and consultancy tools).

2.3 Adapted Roadmap

Taking into account the considerations mentioned in the previous subsections and the development status after the first release, the roadmap for the Portal development was updated in such a way it would fit better with the needs of the project.

In the second release of the Portal (M20), the planned functionalities were:

- Update of the CKAN extensions in the data management tools, according to new needs or the availability of other features
- Link CKAN extensions to the WP3 component for data movement
- Complete tools for community management
- Provide a tool for yellow pages and matchmaking that will be able to support consultancy services
- Basic interaction mechanisms with HPC centres
- Initial set of consultancy support tools (publication of services, ticketing tools, etc.)

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

- Complete and enhance interaction mechanisms with HPC centres as much as possible
- Marketplace for publishing tools, systems and datasets
- Code repository, testing support tools and completion of Jenkins integration
- Implementation of the process which will support the workflow for SIS
- Complete consultancy tools as needed

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

- Solve pending bugs
- Populate the marketplace
- Extend the training and community material
- Complete any pending development

3. Implemented Portal Architecture

3.1 Implemented Features

For the second release of the CoeGSS Portal, according to [1] and the new roadmap plan, we have focused on the following new features and enhancements:

- Implementation of two new CKAN extensions adding new functionality to the Data Manager component:
 - Dataset Relationship Manager:
 - Data Movement
- Implementation of two new sub-components for the Frontend both of which aim to support the user in utilizing CoeGSS:
 - Yellow pages
 - Matchmaking Tool
- Deployment and integration of a single sign-on (SSO) infrastructure to eliminate the need to sign in to each CoeGSS component separately.
- Visual improvements for the Training, Community and Data Manager components to be in-line with the CoeGSS visual style.
- Definition and Implementation of the HPC-aaS initial features.

The status of these features and enhancements are detailed for each component respectively in sections 4 to 8.

3.2 Implemented High Level Architecture

The current architecture of the CoeGSS Portal (see: Figure 1) does not have major changes over the previous one [4], keeping the major components and their relationships as they were. The only change that has been introduced to the architecture is related with the authorization and authentication.

The first release of the CoeGSS Portal was relying only on LDAP for authentication and authorization for the Frontend and the other services provided by the portal. With the second release, we introduce a SSO mechanism provided by a Shibboleth 2.0 deployment, which relies on LDAP for authentication. In the current architecture, the authentication process of the users against the LDAP backend is moved from the services layer to Shibboleth and all the services are now communicating with Shibboleth to challenge user sessions.

The motivation for deploying a single sign-on mechanism and its details are documented thoroughly in section 5.

All the tools and code developed have been uploaded to the internal CoeGSS Git repository (hosted at HLRS under the R1 master branch) so they are available to the CoeGSS consortium. This repository is private but, as the Portal evolves, its code will be made public.

For further details about the implementation of any of the mentioned components, please refer to the next sections.

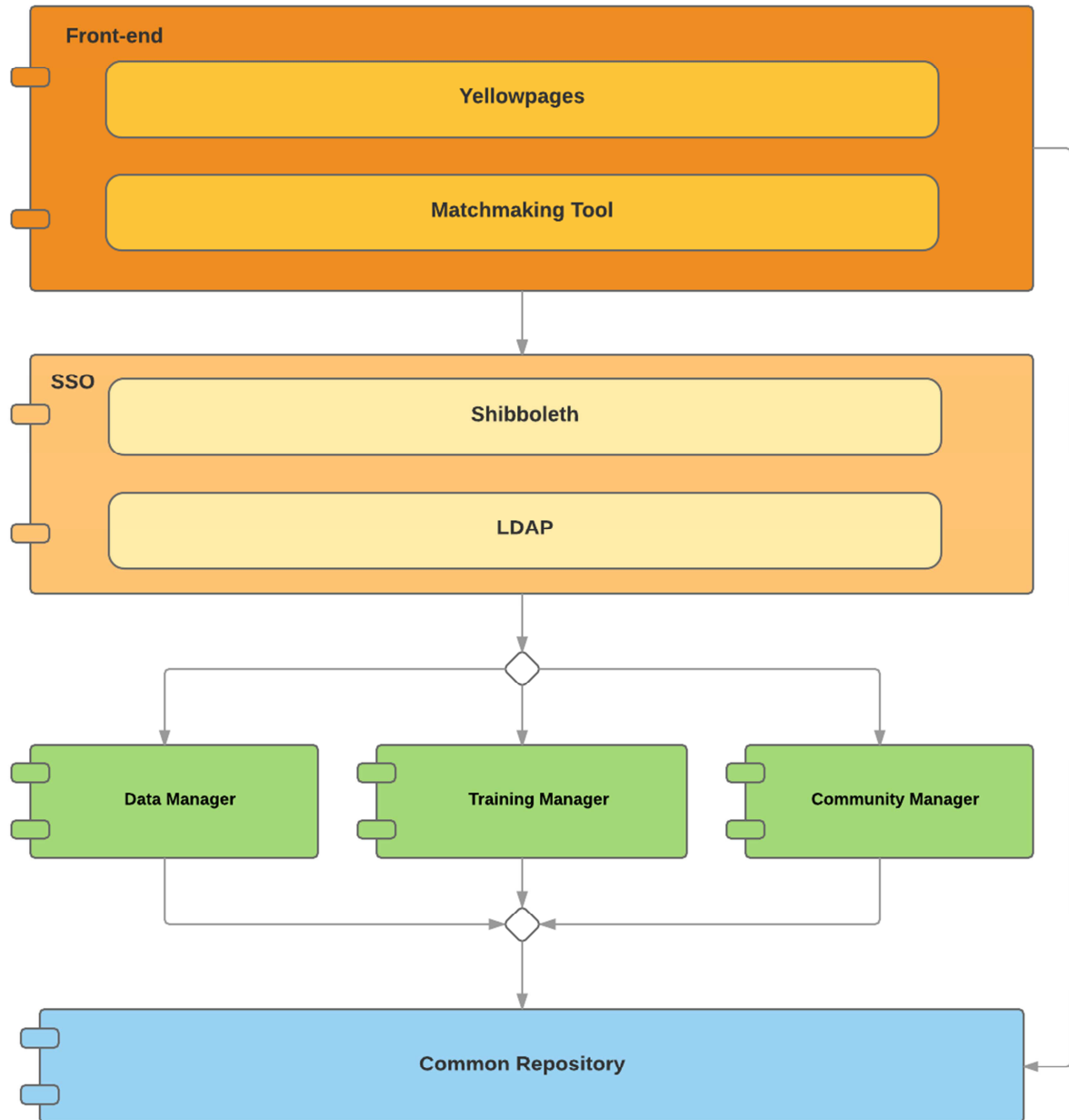


Figure 1. CoeGSS high-level architecture

4. Frontend

This section describes the Frontend component, which provides a single point of access for all other components that are implemented in the context of CoeGSS.

4.1 Component Description

Besides the login and registration functionality, the Frontend component of the first release of the CoeGSS Portal [4] was only providing access to the other CoeGSS components. The second release of the Portal introduces some changes to the configuration and deployment and two new tools, which can be considered as sub-components of the Frontend.

4.2 Sub-Components

According the roadmap defined in [2], two new CoeGSS tools have been implemented, both of which can be accessed from the Services menu on the main navigation bar of the Frontend: The Matchmaking Tool and the Yellow Pages.

4.2.1 Yellow Pages

The *Yellow Pages* is a sub-component of the Frontend, implemented within the context of CoeGSS using the Django framework and the Python programming language. This component provides a catalogue of organisations that have a relationship with CoeGSS, which can be browsed and searched through by any user of the Portal. This catalogue will be extended with experts and organization employees in the next iteration of the CoeGSS Portal.

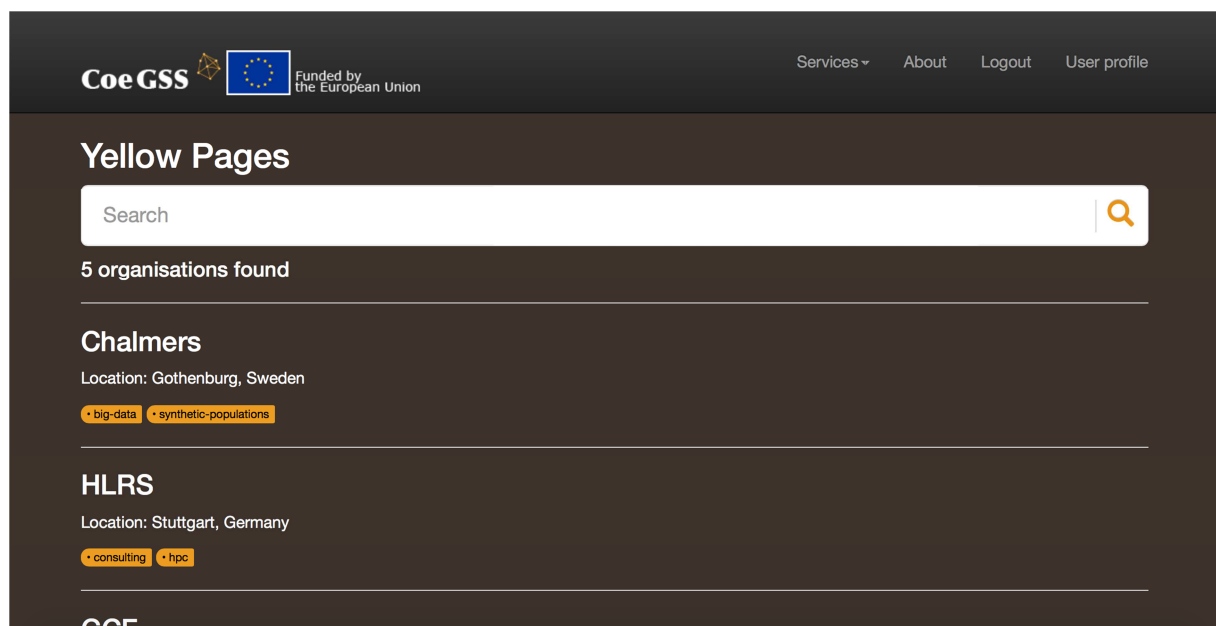


Figure 2. Search function of the Yellowpages component

The *search* function allows the user to search for organisations matching the search criteria, or simply browse through all organisations, as shown in Figure 2.

Once the desired organisation is found, the user can view its profile and detailed information, as shown in Figure 3.

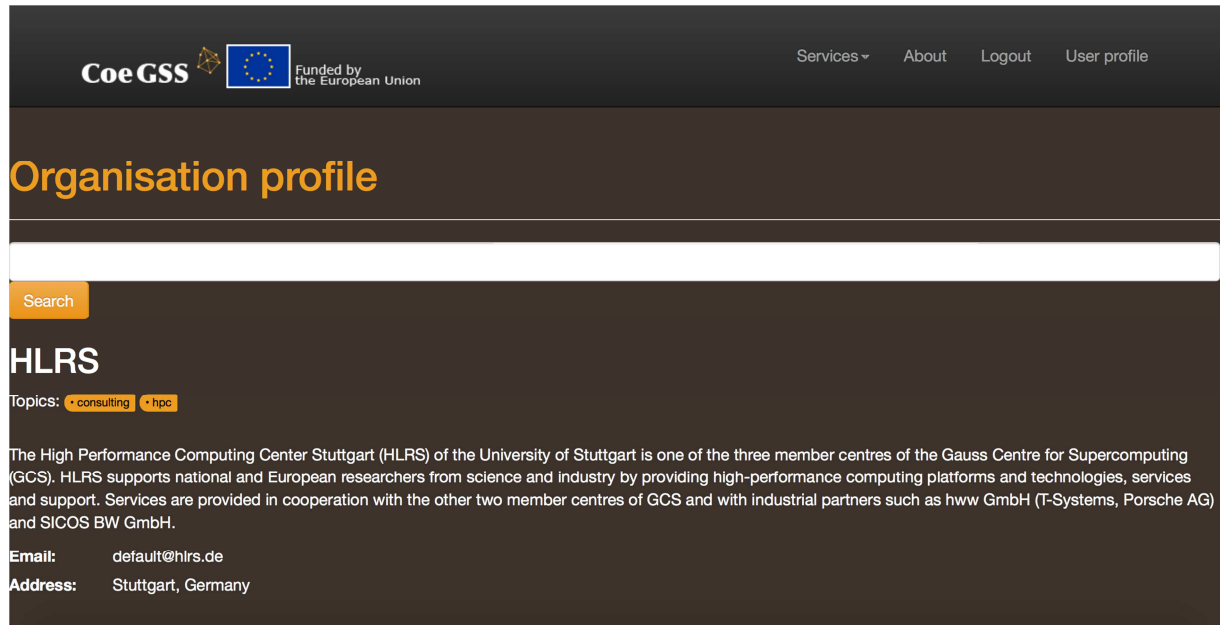


Figure 3. Organisation profile in Yellow pages

Basic data, such as contact data and description are available on the profile pages. In addition, every organisation is described by a number of *topics*, which indicate the areas in which the organisation is active, such as *HPC* or *data-analytics*. Topics provide a convenient way of categorising organisations and can be used to browse through organisations belonging to a certain topic.

The *Yellow Pages* component interacts with the database, as shown in Figure 4 in the next section, by reading profile information from the database, and allows certain users to edit profiles of their own organisations.

4.2.2 Matchmaking Tool

The Matchmaking Tool is a web based graphical user interface that allows searching for particular requirements related with the CoeGSS project. The tool was implemented using the Django framework and the Python language after the first Portal release. The Matchmaking tool consists of several web pages where the user can search and visualise information about different topics.

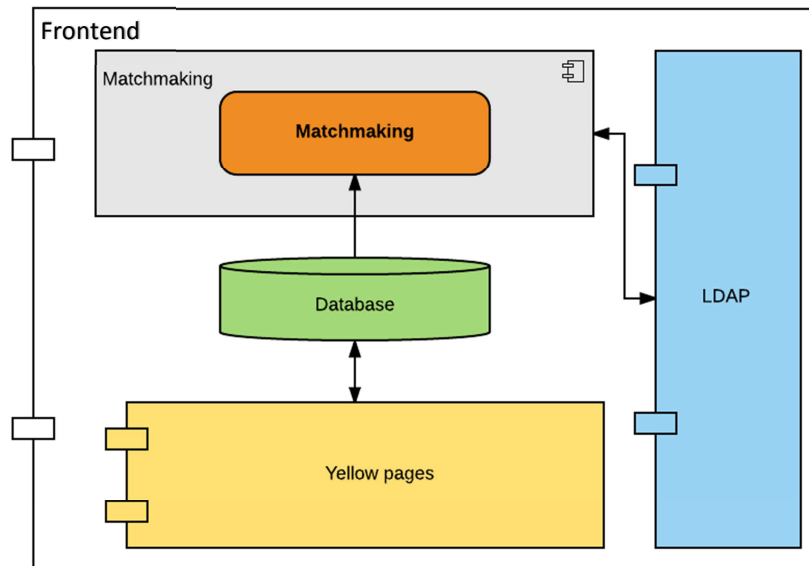


Figure 4. Detailed design for the matchmaking tool

The Matchmaking tool is a sub-component that will use the data stored by the **Yellow Pages** in order to provide the best possible contact point when the user requests information through the Portal. As mentioned before, the **Database** will be populated by the Yellow Pages, which contains records about Organizations, Users and Resources, grouped by Categories. The component is available only for registered users, therefore it uses the **LDAP** credentials to give access to this section. In Figure 4, the detailed design for Matchmaking tool is presented.

After entering username and password via the login page the user can access the Matchmaking menu option on the Services menu as presented in Figure 5. When the user chooses the Matchmaking menu option, the web interface of the tool is displayed, see Figure 6. There, he/she will select the criteria in which he/she is interested on and then press the button *Match* to request the information. Subsequently, a page will be presented with the matching of the request and the contact data suitable for the user. This result could consist of a single or a set of partners that match the request.

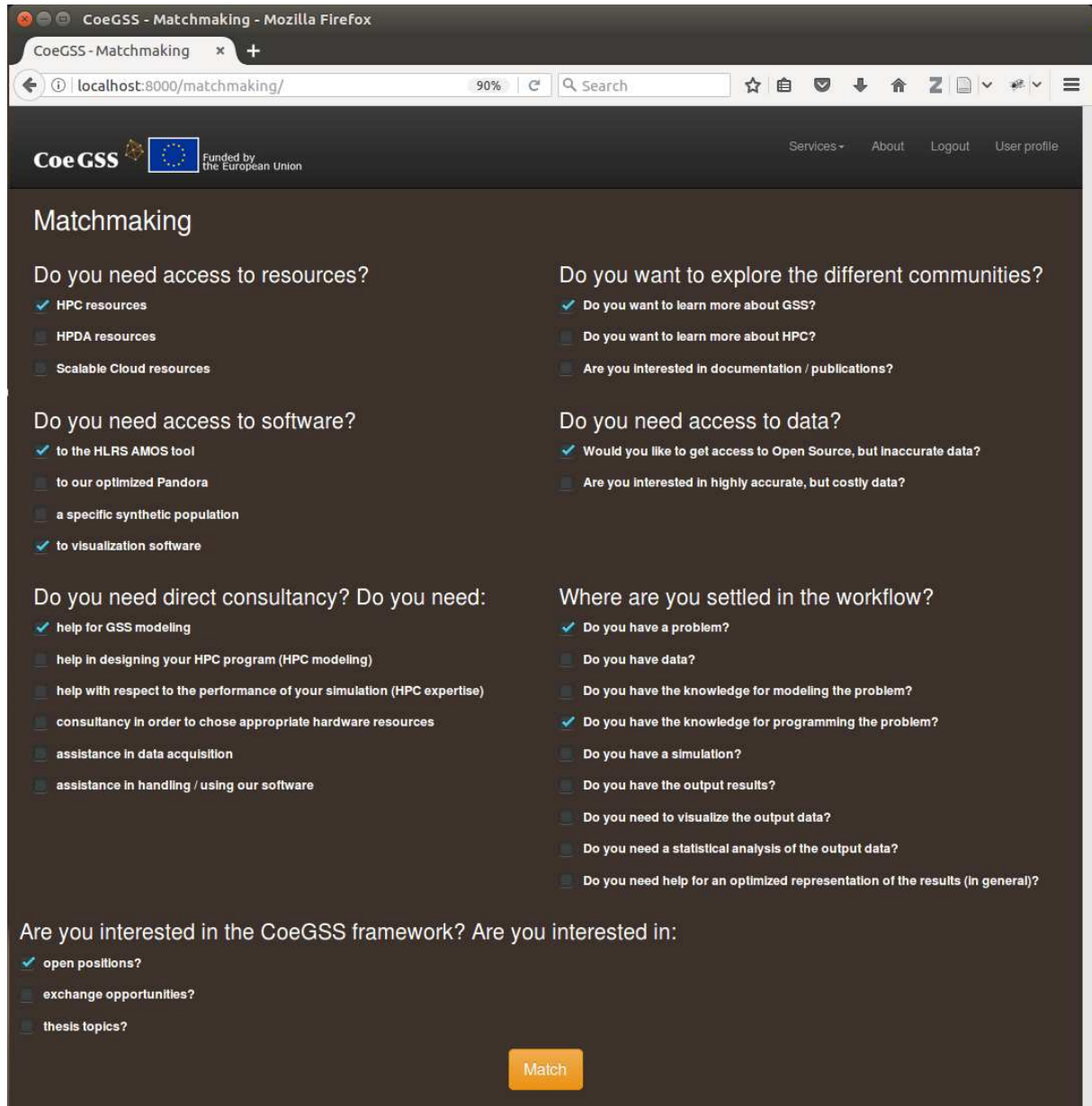


Figure 5. Matchmaking web interface

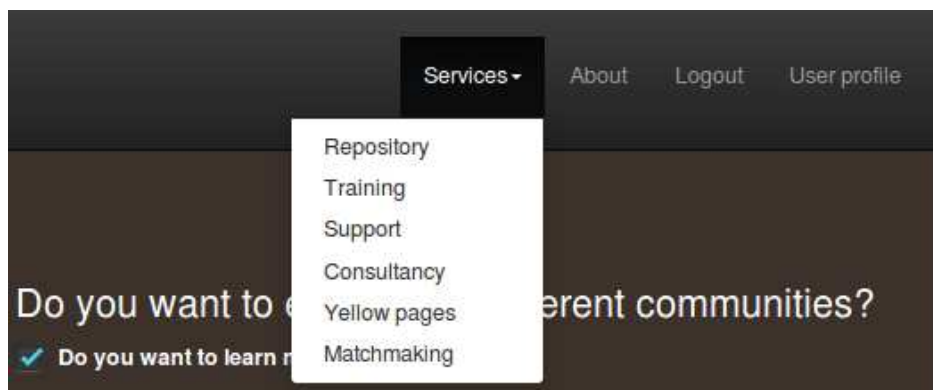


Figure 6. Matchmaking tool menu option

4.3 Component Configuration

The Frontend component's main configuration point is the *settings.py* file, which can be found under the relative path *coegss_portal/coegss_portal/settings.py*. Through this file, the path for the static and template files can be modified, authentication backends as well as LDAP configuration can be managed, Database connection can be configured and other Django apps can be added to the web application.

The current settings for the configuration points listed above are listed as follows:

```
BASE_DIR = os.path.dirname(os.path.dirname(os.path.abspath(__file__)))
```

```
TEMPLATE_PATH = os.path.join(BASE_DIR, 'templates')
```

```
STATIC_PATH = os.path.join(BASE_DIR, 'static')
```

```
INSTALLED_APPS = [  
    'django.contrib.admin',  
    'django.contrib.auth',  
    'django.contrib.contenttypes',  
    'django.contrib.sessions',  
    'django.contrib.messages',  
    'django.contrib.staticfiles',  
    'frontend',  
    'matchmaking',  
    'tags_input',  
]
```

```
AUTHENTICATION_BACKENDS = [  
    'django_auth_ldap.backend.LDAPBackend',  
    'django.contrib.auth.backends.ModelBackend',  
]
```

```
AUTH_LDAP_SERVER_URI = "ldap://172.18.10.41"
```

```
AUTH_LDAP_BIND_DN = "cn=admin,dc=coegss,dc=hlrs,dc=de"
```

```
AUTH_LDAP_BIND_PASSWORD = "<password>"
```

```
AUTH_LDAP_USER_SEARCH = LDAPSearch("ou=People,dc=coegss,dc=hlrs,dc=de",  
ldap.SCOPE_SUBTREE, "(uid=%(user)s)")
```

```
DATABASES = {  
    'default': {  
        'ENGINE': 'django.db.backends.postgresql_psycpg2',  
        'NAME': 'portal_db',  
        'USER': 'portal',  
        'PASSWORD': '<password>',  
        'HOST': 'localhost',  
        'PORT': '',  
    }  
}
```

4.4 Component Deployment

The current version of the Frontend component is deployed in a virtual machine running Ubuntu 14.04 and is hosted at the High Performance Computing Center Stuttgart. The service can be accessed via [http:// portal.coegss.hlr.de](http://portal.coegss.hlr.de)

4.5 Component Testing

As the previous version [4], the current version of the component has been mostly tested manually and the plans for creating automated testing mechanisms for the component has been postponed to the next iteration.

5. Authentication and Authorization

Within this section of the deliverable, the CoeGSS authentication mechanism, which is now composed of Shibboleth and LDAP is described. This SSO mechanism represents a key component of the entire CoeGSS Portal architecture since it is used for user management including authentication and authorization.

5.1 Component Description

In the previous release, this component was composed of an LDAP deployment only, which was serving as an identity provider for all the services of CoeGSS. Although the users' authentication information was centralized and the users could login to the frontend and the services using the same credentials, the login process had to be repeated for each individual service which was cumbersome and made it harder to maintain the user sessions, if not impossible.

The second release of the CoeGSS Portal aims at eliminating this problem by introducing a single sign-on mechanism provided by a Shibboleth 2.0 deployment that relies on the existing LDAP deployment for authentication. Shibboleth consists of two sub-components:

- Identity Provider (IdP), responsible for accepting authentication requests and authenticating the users against LDAP
- Service Provider (SP), responsible for intercepting access to a protected resource and issuing a SAML authentication request to the selected IdP.

The new authentication flow based on this new Shibboleth deployment works as follows¹:

1. User accesses a restricted resource and is directed to the SP
2. SP prepares an authentication request and sends it to IdP
3. IdP authenticates the user against LDAP
4. IdP sends the authentication response to SP
5. SP validates the response, creates a session for the user and sends him/her to the resource.

Although the second release of the CoeGSS portal introduces major changes to the authentication mechanism, there have been no changes to the existing LDAP installation or configuration, Shibboleth IdP is simply configured to use the existing deployment as it is. Please refer to [4] for a detailed description of the LDAP installation.

¹ See: <https://shibboleth.net/about/basic.html> for more information.

5.2 Component Configuration

There have been no changes to the configuration of LDAP. Please refer to [4] for a detailed explanation on the configuration of the current LDAP deployment.

On the other hand, it was necessary to configure Shibboleth appropriately. Since tests are still being carried out for the Shibboleth deployment, the SSO mechanism in general, the configuration details provided below could have minor changes.

5.2.1 IdP Configuration

There are several configuration files for IdP, most of which have been left at its default values except the ones that are listed below (related to the storage and the LDAP configuration):

../shibboleth-idp/conf/global.xml

```
<bean id="MyDataSource" class="org.apache.commons.dbcp.BasicDataSource"
  p:driverClassName="com.mysql.jdbc.Driver"
  p:url="jdbc:mysql://localhost:3306/shibboleth?autoReconnect=true"
  p:username="##USER_DB##"
  p:password="##PASSWORD##"
  p:maxActive="10"
  p:maxIdle="5"
  p:maxWait="15000"
  p:testOnBorrow="true"
  p:validationQuery="select 1"
  p:validationQueryTimeout="5" />

<bean id="shibboleth.JPASStorageService" class="org.opensaml.storage.impl.JPASStorageService"
  p:cleanupInterval="%{idp.storage.cleanupInterval:PT10M}"
  c:factory-ref="shibboleth.JPASStorageService.entityManagerFactory"/>

<bean id="shibboleth.JPASStorageService.entityManagerFactory"
  class="org.springframework.orm.jpa.LocalContainerEntityManagerFactoryBean">
  <property name="packagesToScan" value="org.opensaml.storage.impl"/>
  <property name="dataSource" ref="MyDataSource"/>
  <property name="jpaVendorAdapter" ref="shibboleth.JPASStorageService.JPAVendorAdapter"/>
  <property name="jpaDialect">
    <bean class="org.springframework.orm.jpa.vendor.HibernateJpaDialect" />
  </property>
</bean>

<bean id="shibboleth.JPASStorageService.JPAVendorAdapter"
  class="org.springframework.orm.jpa.vendor.HibernateJpaVendorAdapter">
  <property name="database" value="MYSQL" />
</bean>
```

../shibboleth-idp/conf/idp.properties

```
idp.session.StorageService = shibboleth.JPASStorageService
idp.consent.StorageService = shibboleth.JPASStorageService
idp.replayCache.StorageService = shibboleth.JPASStorageService
idp.artifact.StorageService = shibboleth.JPASStorageService
```

../shibboleth-idp/conf/ldap.properties

```
idp.authn.LDAP.authenticator = bindSearchAuthenticator
idp.authn.LDAP.ldapURL = ldap://172.18.10.41:389
idp.authn.LDAP.useStartTLS = false
```

```
idp.authn.LDAP.useSSL = false
idp.authn.LDAP.baseDN = ou=People,dc=coegss,dc=hlrs,dc=de
idp.authn.LDAP.userFilter = (uid={user})
idp.authn.LDAP.bindDN = cn=admin,dc=coegss,dc=hlrs,dc=de
idp.authn.LDAP.bindDNcredential = ###LDAP ADMIN PASSWORD###
```

5.2.2 SP Configuration

The Service Provider component has also several configuration files but the only file worth mentioning that is crucial to the component's operation is the *shibboleth2.xml* file, for which the modification details for CoeGSS can be found below (basically, to configure the IdP at HLRS):

../shibboleth/shibboleth2.xml

```
<ApplicationDefaults entityID="https://sp.coegss.hlrs.de/shibboleth"
    REMOTE_USER="eppn persistent-id targeted-id">

<Sessions lifetime="28800" timeout="3600" checkAddress="false"
    relayState="ss:mem" handlerSSL="true" cookieProps="https">

<MetadataProvider type="XML" validate="true" file="idp-metadata.xml"/>

<SSO entityID="https://idp.coegss.hlrs.de/idp/shibboleth">
    SAML2 SAML1
</SSO>
```

5.3 Component Deployment

The LDAP system, which was deployed for the first release of the CoeGSS portal, has not been changed and details about its deployment can be found in [4].

Both of the Shibboleth components are hosted by their own virtual machines at the High Performance Computing Center Stuttgart. The virtual machine hosting the IdP component is based on a standard Ubuntu 16.04 installation, which is extended to provide the ability to store Secure Shell (SSH) public keys. On the other hand, since Ubuntu is not officially supported by Shibboleth for installing the SP component, our deployment is hosted by a virtual machine with a Centos 7 installation.

Neither of the VMs are reachable from outside world and only have local IPs; 172.18.10.50 for IdP and 172.18.10.51 for SP.

5.4 Component Testing

This component has only been tested manually.

6. Data Manager

This section describes the Data Manager component, which enables the users to find, publish/share and make use of the data that already exist on the system.

6.1 Component Description

The second release of the CoeGSS Portal introduces two new third party CKAN extensions:

- `ckanext-extractor` automatically extracts text and metadata from the existing resources on CKAN and adds them to the search index so that they can be found easily using the search functionality of CKAN
- `ckanext-oaimph` extends the CKAN Harvester to parse OAI-MPH (Open Archives Initiative-Protocol for Metadata Harvesting) metadata sources and import datasets.

Besides these two third party extensions, two custom extensions; the Dataset Relationship Manager and Data Movement have been implemented and integrated into CKAN, which were developed by ATOS and PSNC respectively.

For a detailed description of this component with the rest of its extensions installed for the first release of the CoeGSS portal, please refer to [4].

6.2 Custom Component Extensions

6.2.1 Dataset Relationship Manager

Although CKAN provides several ways to tag, group and search datasets, it does not provide the user any functionality through its user interface to create relational or hierarchical links between the datasets. Being able to create such links would not only increase the accessibility to the datasets, but it would also help users make more sense of the datasets they are browsing.

With this motivation, a CKAN extension called Dataset Relationship Manager `ckanext-drel` has been implemented. The extension adds a new tab to the dataset view where the user can see and browse to the existing relationships created for that dataset. The user can switch directly to relationship editing mode from this page by clicking the Add / Remove Relations button. Another way to access the page to create relations for a dataset or remove them is using the Manage button of CKAN for a dataset then going to the Dataset Relations tab.

The user can create dependency, derivation and parent-child relationships or link datasets together. To be specific, these relationships can be created between the datasets: *depends on*, *is a dependency of*, *derived from*, *is derived by*, *links to*, *is linked from*, *is child of*, *is parent of*.

6.2.2 Data Movement

The data acquisition process is one of the first that is performed in the CoeGSS workflow. In order to do that, the CKAN harvester (implemented as CKAN plugin) was designed. It is responsible for data import from outside sources (e.g. MIDAS database) to the CKAN server. Information about the data location can be provided as a link to a remote location or the location of a file located at a local disc.

The harvester is currently capable of handling the following file types: CSV, TXT, XLS. Data included in the files are automatically parsed and converted to a tabular form used by the database. This process provides a lot of additional functionality from the CKAN service like: direct data exploration in the Portal, generation of the map with agents and their properties represented on it, extraction specific subsets of data for further processing. Remaining types of files can be also stored in the Data management system but converting procedure is not performed yet.

In order to manage data properly the standard importing form provided in the CKAN installation was adjusted to the CoeGSS needs. Improvement was based on adding meta data fields like: Scenario ID, Application ID, Source, Reliability source, License and removing those that are not needed any more.

6.3 Component Configuration

CKAN is configured through its <settings>.ini file that can be found in the following path /etc/ckan/default/development.ini by default. This file allows us to configure and manage the CKAN instance as well as the CKAN extensions (plugins).

The server IP and port can be changed by setting the values below in the [server:main] section of the settings file.

```
[server:main]
use = egg:Paste#http
host = 0.0.0.0
port = 5000
```

LDAP settings

```
ckanext.ldap.uri = ldap:// 172.18.10.41
ckanext.ldap.base_dn = ou=users,dc=ldap,dc=coegss,dc=com
ckanext.ldap.username = uid
ckanext.ldap.email = mail
ckanext.ldap.search.filter = uid={login}
ckanext.ldap.auth.dn = cn=ckan_ckan,ou=users,dc=ldap,dc=coegss,dc=com
ckanext.ldap.auth.password = ckan
```

CKAN plugins can be disabled by simply removing them from the settings entry below. For example, in order to disable for example disqus, one must remove the first entry from the list below.

```
ckan.plugins = disqus showcase drel stats text_view image_view recline_view
datastore harvest ckan_harvester extractor coegss_theme
```

It has also been necessary to define the location for storing datasets, where CKAN should have writing permission. CKAN **storage path** has been configured as:

```
ckan.storage_path = /var/lib/ckan/default
```

6.4 Component Deployment

The current version of the Data Manager component is running in a virtual machine running Ubuntu 14.04 and is hosted at the High Performance Computing Center Stuttgart. In order to access it, the following uniform resource locator can be used: <http://ckan.coegss.hls.de>. The harvester functionality of the Data manager can be accessed through this URL: <http://ckan.coegss.hls.de/harvest>.

6.5 Component Testing

The current version of the Data Manager component has been tested manually.

7. Training Manager

This section describes the Training Manager component, which provides the training features for both experts and consumers. The Training Manager will play a key role for linking the field of GSS and HPC and educating bridging experts between the two fields.

7.1 Component Description

The second release of the CoeGSS Portal only introduces small visual changes to the Training Manager component. Please refer to [4] for a detailed description of the component itself.

In this version, the look and feel of the component was fine tuned to match the visual style of CoeGSS. This fine-tuning was conducted by using the Moodle UI options in the administration setup pages.

7.2 Component Configuration

There have been no changes to the configuration of this component. Please refer to [4] for a detailed explanation on the configuration of this component.

7.3 Component Deployment

There have been no changes to the deployment details of this component except the service is now available by this fully qualified domain name: <http://moodle.coegss.hlr.de>. Please refer to [4] for deployment details of this component.

7.4 Component Testing

The current version of the Training Manager component has been tested manually. This has been done while training content was created by the Global Climate Forum (“Course on the International Conference on Synthetic Populations”) and HLRC training course were linked within Moodle.

8. Community Manager

This section describes the Community Manager component that provides community management and support for both, experts and consumers.

8.1 Component Description

The second release of the CoeGSS Portal only introduces small visual changes to the Community Manager component. Please refer to [4] for a detailed description of the component itself.

In this version, the look and feel of the component was fine tuned to match the visual style of CoeGSS. This fine-tuning was conducted by using the AskBot UI options in the administration setup pages and by modifying AskBot's CSS files.

8.2 Component Configuration

There have been no changes to the configuration of this component. Please refer to [4] for a detailed explanation on the configuration of this component.

8.3 Component Deployment

There have been no changes to the deployment details of this component except the service is now available by this fully qualified domain name: <http://support.coegss.hlr.de>. Please refer to [4] for deployment details of this component.

8.4 Component Testing

To test the component, manual testing has been conducted: login of a LDAP user, create a question, create an answer, and do full text search. Currently there is no automated monitoring, but such test can be added later to the already available Nagios monitoring system to test general availability of host and Q&A service.

9. Infrastructure Services

Within this section of D5.11, the infrastructure services of the CoeGSS Portal are highlighted. In particular, changes are focused, which have been introduced since the deployment of version 1 of the Portal.

9.1 Hosting Description

As already highlighted in D5.2 [5], D5.9 [2] and D5.10 [4], the CoeGSS Portal infrastructure is following a layered approach. The concept has been implemented by making use of virtual machines (VMs), which are easy to maintain and enable simplified adjustments, especially for the key performance indicators such as service response time or service availability.

When developing version 2 of the CoeGSS Portal, different requirements arose over time that also need to be reflected by the infrastructure environment. In particular, the single-sign-on functionality requires two more services, which have to be encapsulated in order to implement the mandatory logical separation. As a consequence, the amount of virtual machines had to change: 7 virtual machines have been required for Portal v1 while version 2, instead, requires 9 VMs.

Although the amounts of VMs changed, the layers are still following the same methodology: layer 1 focuses the direct user services, layer 2 the authentication mechanisms and finally layer 3 addresses the infrastructural needs of the CoeGSS Portal. All three layers are schematically shown in Figure 7.

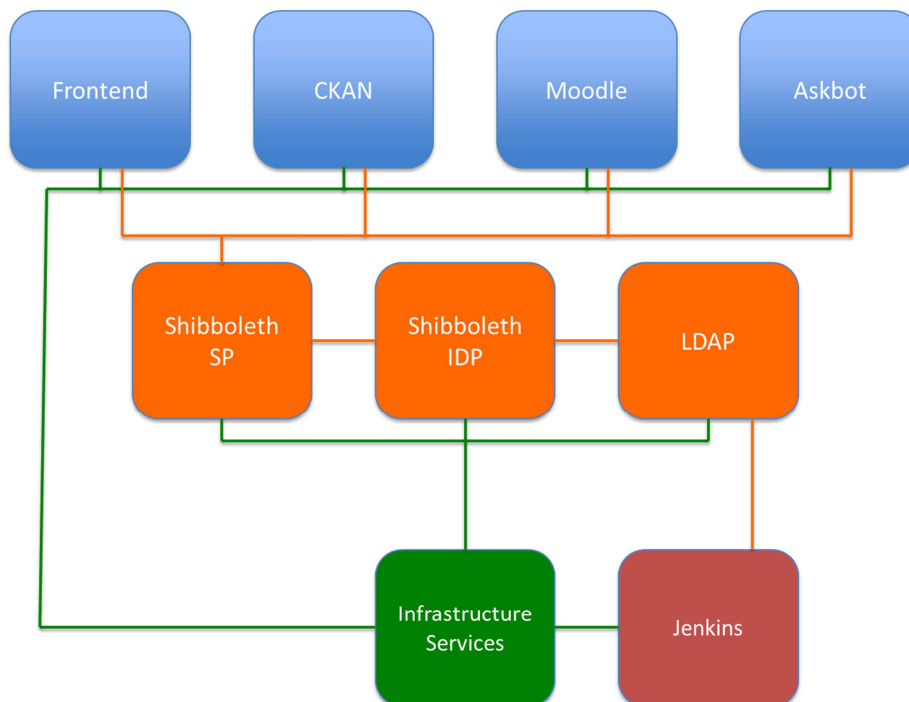


Figure 7. Layered approach of Portal v2

A detailed description about the hosting environment can be obtained in Table 2. For a better understanding, the additional components of version 2 of the Portal are coloured in transparent orange.

Name	URL	Private IP	Public IP	Protocol
Infrastructure	- / -	172.18.10.30	- / -	SSH, HTTP
Jenkins	- / -	172.18.10.41	- / -	SSH, HTTP
LDAP	- / -	172.18.10.40	- / -	SSH, LDAP
Frontend	portal.coegss.hlrs.de	172.18.10.43	141.58.0.67	SSH, HTTP
CKAN	ckan.coegss.hlrs.de	172.18.10.44	141.58.0.64	SSH, HTTP
Moodle	moodle.coegss.hlrs.de	172.18.10.42	141.58.0.65	SSH, HTTP
Askbot	support.coegss.hlrs.de	172.18.10.45	141.58.0.66	SSH, HTTPS
Shibboleth Identity Provider	- / -	172.18.10.50	- / -	SSH, HTTP, HTTPS
Shibboleth Service Provider	- / -	172.18.10.51	- / -	SSH, HTTP, HTTPS

Table 2. CoeGSS Portal VM details

9.2 Hosting Configuration

The CoeGSS Portal infrastructure still follows the concepts, which have been presented in D5.2, D5.9 and D5.10. For Portal v2, no particular changes have been implemented so that the statements of the three deliverables are still valid

10. Future Work

For the next release of the CoeGSS Portal, one of the most important goals is to introduce HPC-aaS features to provide the user a way to interact with the HPC providers. The discussions have already started to explore the interaction possibilities and define the first set of HPC-aaS features, and there is already some implementation on going for submitting jobs to SLURM and TORQUE based HPC systems.

Another important topic for the next release is to create a better role hierarchy for users of the CoeGSS Portal in order to support the users more efficiently and improve the maintainability of the portal community.

Although it is not a part of the technical implementation of the portal, content creation for the Community and Training components is also an important work item for the future, even if there is already some content published.

11. Summary

This document has described the current implementation of the CoeGSS Portal, providing information about the components deployed, their configurations and the testing performed, highlighting the changes and referring to the previous version of the deliverable where necessary. Therefore, not all functionality is described here as they were already documented in [4].

The initial plans for the roadmap were changed, in order to be more in line with the business models defined, as a way to enable supporting some of those business models as early as possible.

It is important to highlight that the current features will keep evolving in the next release, including more functionalities or even more content available for the Portal users. Since the baseline tools are open source and allow for flexibility (i.e. extensions or plugins) it will be possible to evolve their capabilities as we receive user feedback from the pilots and as we identify other potential functionalities that will be useful for them in the near future.

References

- [1] C. Consortium, "D4.1 - First Report on Pilot Requirements," 2016.
- [2] C. Consortium, „D5.9 - Initial Portal Design," 2016.
- [3] C. Consortium, „D2.1 – Stakeholder Context and Initial Sustainability Model," 2016.
- [4] C. Consortium, „D5.10 - First Portal Release," 2016.
- [5] C. Consortium, „D5.2 - First Operation Report," 2016.

List of tables

Table 1. Business Models and Portal Features 7
 Table 2. CoeGSS Portal VM details..... 26

List of figures

Figure 1. CoeGSS high-level architecture 10
 Figure 2. Search function of the Yellowpages component 11
 Figure 3. Organisation profile in Yellow pages..... 12
 Figure 4. Detailed design for the matchmaking tool 13
 Figure 5. Matchmaking web interface 14
 Figure 6. Matchmaking tool menu option 14
 Figure 7. Layered approach of Portal v2 25

List of Abbreviations

DoW	Description of Work
EC	European Commission
CoeGSS	Centre of Excellence for Global System Science
GUI	Graphical User Interface
HPC	High Performance Computing
HPC-aaS	High Performance Computing as a Service
HPDA	High Performance Data Analysis
HTTP	Hypertext Transfer Protocol
IdP	Identity Provider
LDAP	Lightweight Directory Access Protocol
LDIF	LDAP Data Interchange Format
Q&A	Questions and Answers
SEO	Search Engine Optimization
SP	Service Provider
SSO	Single Sign-On
URL	Uniform Resource Locator
VM	Virtual Machine
WP	Work Package