

D5.10 - FIRST 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	30th June 2016
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, Paweł Wolniewicz
Internal Reviewers	Andreas Geiges, Paweł Wolniewicz
Keywords	Portal, Tools, CoE Services
Total number of pages:	24

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

	Name	Partner	Date
From	F. Javier Nieto	ATOS	13.06.2016
Intermediate Draft	F. Javier Nieto, Michael Gienger, Paweł Wolniewicz	ATOS, USTUTT, PSNC	24.06.2016
First Version	F. Javier Nieto, Burak Karaboğa	ATOS	27.06.2016
Reviewed by	Andreas Geiges, Paweł Wolniewicz, Marion Dreyer	GCF, PSNC, DIA	28.06.2016
Updated Version	Burak Karaboğa, F. Javier Nieto	ATOS	30.06.2016
Approved by	ECM Board	ALL	30.06.2016

1. Abstract

Together with the first release of the CoeGSS Portal, it is necessary to provide some additional information which may be necessary for understanding the implementation performed and for facilitating new deployments. This document aims at providing important information about the components developed and deployed, as well as about the way they have been configured. The document also provides additional information about the infrastructure services provided together with the first version of the Portal, such as monitoring.

Table of Contents

1. Abstract	3
Table of Contents	3
2. Introduction	5
3. Implemented Portal Architecture	6
4. Frontend	9
5. LDAP Authentication	11
6. Data Manager	13
7. Training Manager	15
8. Community Manager	17
9. Infrastructure Services	20

10. Summary..... 22

References 23

List of tables 24

List of figures..... 24

List of Abbreviations..... 24

2. Introduction

According to the design described in [1], the CoeGSS team has been developing the CoeGSS Portal, as an important tool for facilitating HPC adoption in the GSS environment. Since the mentioned document pointed out some priorities, the first release has been focused on implementing those functionalities which will facilitate the development of CoeGSS pilots during the first stage of the project, providing tools for data management, training and community management, together with an initial frontend and an authorization tool which facilitate navigation and users management.

This document complements the CoeGSS Portal implementation, describing 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, in 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. Finally, Section 10 summarizes the document and provides some conclusions.

3. Implemented Portal Architecture

3.1 Implemented Features

For the first release of the CoeGSS Portal, according to [1] and the requirements coming from [2], we had committed to provide the following features for the first release (as described in the Portal roadmap):

- Data management, so it will be easier for pilots and other users to publish, find and manage data in general;
- Training, so it will be possible to publish training courses about HPC and GSS;
- Community management, so we will enable the possibility to provide support to users and create discussion spaces.

The current implementation of the Portal provides these features in the following way:

- *Data management*: A CKAN [3] instance is part of the Portal, together with several extensions which facilitate the download and upload of data, comments, harvesting, etc...;
- *Training*: A Moodle [4] instance is integrated in the Portal, with just a few initial extensions, but available for publishing courses;
- *Community management*: Askbot [5] is the tool provided as part of the Portal, in order to have a discussion space based on questions and community answers.

Besides these main features, also the required authentication features are available, providing a unique mechanism for managing users in all the tools.

3.2 Implemented High Level Architecture

The current architecture of the CoeGSS Portal remains the same as the one presented in [1], since all the components identified for providing each feature have been included in the current implementation.

As expected, LDAP is the solution provided for the authentication mechanism, so it contains all the information related to the Portal users' credentials. All the components support authentication with LDAP in different ways (native, by plugins or by extensions), so it has been totally integrated in the architecture.

Moreover, users' management is complemented with the Frontend, which facilitates operations such as login or users registration. It also provides a unique access point for all the tools integrated in the Portal, in such a way users do not need to look for concrete links when accessing the features. This Frontend provides the adequate menus for navigating through the features, customized according to the CoeGSS styles. Therefore, the Frontend is considered the main integrator for the different functionalities provided by the CoeGSS

Portal. For more details about the interactions among the different components with the Frontend, please, refer to the interaction diagrams described in the design document [1].

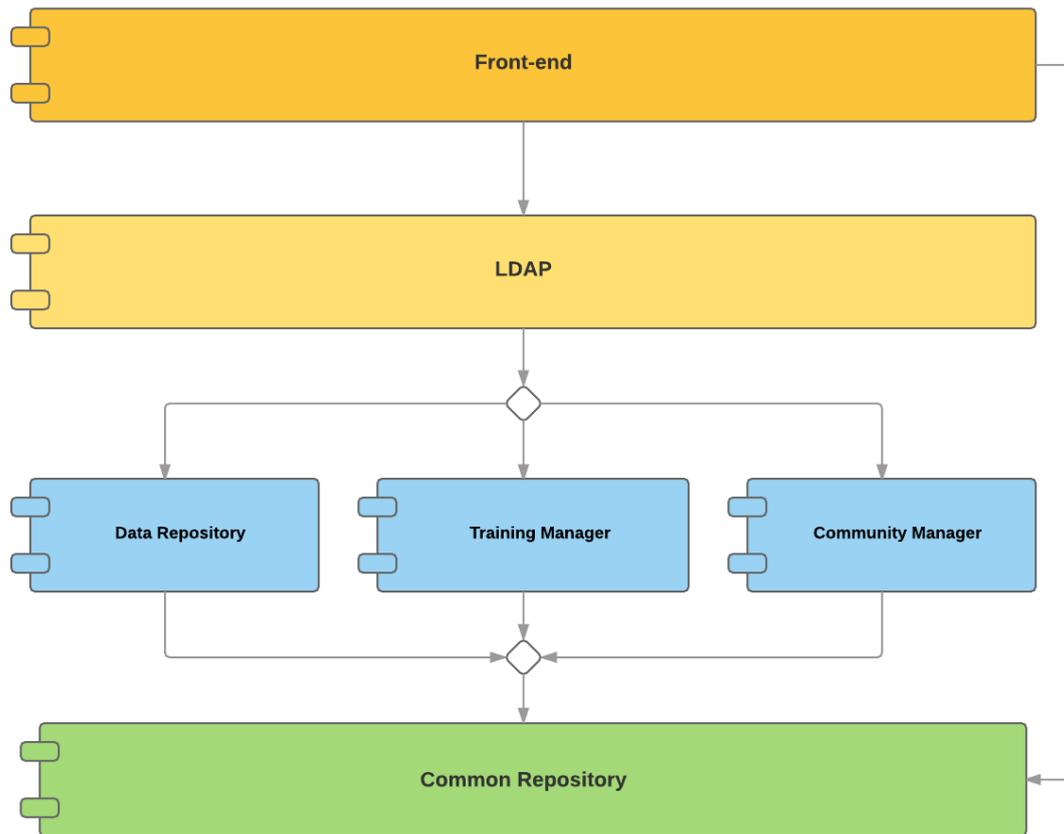


Figure 1. Current high level architecture of the CoeGSS Portal

The three main components remain as in the design document: Data Repository, Training Manager and Community Manager. As mentioned before, each one represents one of the main features included in the first release (data management, training and users’ community management). They do not interact among them, but only with the LDAP component (for authentication) and with the Frontend (which redirects users to the corresponding component depending on the feature they want to use).

Finally, the Common Repository is not a usual component currently, but some storage space enabled for storing those datasets uploaded with the Data Repository component.

All the tools and code developed has been uploaded to the internal CoeGSS Git repository (hosted at HLRS¹ under the R1 master branch) so they are available to the CoeGSS consortium. This repository is private and but, as the Portal evolves, its code will be made public.

¹ <https://projects.hlrs.de>

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

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

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. [1]. In order to implement this web application, we needed a framework which would help us build a small but complex system quickly. For this purpose we chose the Django Python based web framework which follows the model-view-controller architectural pattern and has a big online community.

The current version of the Frontend component requires the following to operate: Python 2.7, Django 1.9.6, Postgresql 9.4. We also used the following Python packages: virtualenv, django-registration-redux, python-ldap, django-auth-ldap.

The current version of the web application is run using the `python manage.py runserver` command on an Ubuntu 14.04 Linux distribution. The next iteration will make use of the version 2.4 of the Apache web server for deployment and running of the Frontend component.

4.2 Component Configuration

The Frontend component's main configuration point is the `settings.py` file which can be found under `coegss_portal/coegss_portal/settings.py` relative path. Through this file, the path for the static and template files can be modified, authentication backends, 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',
```

```
    'registration'
]

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

AUTH_LDAP_SERVER_URI = "ldap://localhost"
AUTH_LDAP_BIND_DN = 'cn=burak,ou=users,dc=ldap,dc=coegss,dc=com'
AUTH_LDAP_BIND_PASSWORD = "<password>"
AUTH_LDAP_USER_SEARCH = LDAPSearch("ou=users,dc=ldap,dc=coegss,dc=com",
    ldap.SCOPE_SUBTREE, "(uid=%(user)s)")

DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.sqlite3',
        'NAME': os.path.join(BASE_DIR, 'db.sqlite3'),
    }
}
```

The port on which the Frontend is served can be configured during the launch of the frontend application through the *runserver* command. In order to launch the Frontend component on port 8000, the command below can be used.

```
python manage.py runserver 0.0.0.0:8000
```

4.3 Component Deployment

The current version of the Frontend component is running in a virtual machine running Ubuntu 14.04 which is hosted at the High Performance Computing Center Stuttgart and can be accessed via the IP address 172.18.10.43 and port 8000.

4.4 Component Testing

The current version of the component has been mostly tested manually and only a small portion of the Frontend is being covered by the automated tests. Django has a built in automatic testing mechanism which will be used to test our component extensively in the next iteration.

The next version of the Frontend component will include tests for ensuring the integrity of the data held in our models and validity of our views.

5. LDAP Authentication

Within this section of the deliverable, the Lightweight Directory Access Protocol (LDAP) is described. This protocol 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

LDAP is a widely used and well-adopted protocol, which is used to manage users and services in a directory-based environment. In order to guarantee the single-sign-on for all involved components for the CoeGSS Portal, this protocol has been chosen. Furthermore, due to its adoption, regular system updates and upgrades are provided so that secure and private operation can be ensured for the full project runtime and beyond.

For the CoeGSS environment, the *slapd* package that is natively distributed with the Ubuntu 14.04 Linux distribution has been installed. In addition to the LDAP server *slapd*, the *ldap-utils* as well as the *phpldapadmin* packages and its dependencies are available in order to manage the system via the command line as well as the graphical user interface (GUI).

5.2 Component Configuration

The installed LDAP relied on the standard configuration files, which are located in the folder */etc/ldap/slapd.d*. Only some minor adaptations have been made in order to set the base CoeGSS tree including an appropriate password.

In order to configure the tree, several actions are required to support all requirements of the involved components. In particular, a generic configuration has been focused that ensures the presence of all attributes and allows the extension of those. The general structure of the LDAP system focuses one single organizational unit called *People*, which consists of individual users, components groups such as *ckan* or *moodle* and management groups to administrate those components. Taking this approach enables controlling the access per user and service and in addition, provides the mandatory clear structure to adjust the settings, if required. Finally, a subset of relations between the object classes and attributes for the user as well as the management accounts are shown in Table 1 below. Colours denote the properties defined for each class in the LDAP structure created for CoeGSS.

dc=coegss,dc=hirs,dc=de		
objectClass: top		
objectClass: posixAccount	cn	Common name
	uid	User identifier
	uidNumber	User identifier number
	gidNumber	Group number

	homeDirectory	Linux home directory
	userPassword	User password
	loginShell	Type of Linux login shell
	description	Additional descriptions
objectClass: inetOrgPerson	homePhone	Home phone number
	homePostalAddress	Home postal address
	mobile	Mobile phone number
	initials	User initials
	mail	User email
	preferredLanguage	Preferred user language
objectClass: ldapPublicKey	sshPublicKey	Secure Shell public key

Table 1: LDAP user scheme

Besides the shown attributes in Table 1, there are different other attributes that are considered to be used. For instance, country, location or city are available via the standard LDAP implementation and may be added whenever necessary.

5.3 Component Deployment

The first LDAP system prototype is running in a virtual machine hosted at the High Performance Computing Center Stuttgart. For providing the service, two central processing unit (CPU) cores and four Gigabyte of main memory are available. The system can be accessed via the private Internet Protocol (IP) address 172.18.10.41 and its corresponding port 389.

As already mentioned, the LDAP component has been deployed by making use of the standard package system of Ubuntu. However, in addition to the standard installation means, the setup has been extended to provide the ability to store Secure Shell (SSH) public keys. For this purpose, the according Secure Shell scheme has been created and transferred into the proprietary LDAP Data Interchange Format (LDIF). Finally, it was added to the CoeGSS implementation and validated by a manual test.

5.4 Component Testing

In order to test the component, a twofold approach has been chosen. On the one hand side, manual testing by integrating the LDAP service in CKAN and Moodle has been performed. But on the other hand, an automated way to test the LDAP server has been focused as well. In conjunction with the Nagios monitoring system [6], which is described below, a small script has been created that periodically accesses the LDAP server to report about its general availability.

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. The system will provide these basic data management functionalities for the portal and pilot users, together with other added value features, such as facilitating harvesting from external sources, identifying links between datasets, publishing opinions about the datasets and uploading datasets to the CoeGSS infrastructure. From this working infrastructure, specific software and solutions in the GSS context will be developed together with the users.

6.1 Component Description

In order to provide the functionality of registering datasets, facilitating search of datasets and providing access to these datasets, CKAN has been chosen as the core application. CKAN is used as well as a FIWARE Generic Enabler (GE) which provides all the functionality listed above and also provides the means to extend its functionality through CKAN Extensions.

The current version of the component uses the 2.5.2 version of CKAN with these extensions: Disqus, Datastore, Harvest, DCAT, Datapusher, Filestore and LDAP. The next iteration will also include the Google Analytics extension. These extensions provide the following functionalities to the component apart from the default CKAN functionality:

- Commenting on datasets
- Automated storing of structured datasets on an SQL database
- DCAT format support for datasets
- Automated dataset harvesting of other CKAN instances.
- LDAP authentication for CKAN

The system will provide the basic data management functionalities for the portal and pilot users. From this working infrastructure, specific software and solutions in the GSS context will be developed together with the users. This extended CKAN installation also requires PostgreSQL 9.4 to be installed apart from the other requirements of CKAN itself.

6.2 Component Configuration

CKAN is configured through its `<settings>.ini` file which 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 disqus, one must remove the first entry on this list.

```
ckan.plugins = disqus stats text_view image_view recline_view
```

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
```

6.3 Component Deployment

The current version of the Data Manager component is running in a virtual machine running Ubuntu 14.04 which is hosted at the High Performance Computing Center Stuttgart and can be accessed via the IP address 172.18.10.44 and port 5000. The harvester functionality of the Data manager can be accessed through this URL: [http:// 172.18.10.44:5000/harvest](http://172.18.10.44:5000/harvest)

6.4 Component Testing

The current version of the Data Manager component has been tested only manually. Although the default CKAN installation comes with some automated tests covering the core functionality, these tests can be extended to cover the functionality related with extensions as well.

Among the different features testing, it is interesting to highlight the harvesting of the datasets provided by the Amsterdam portal² (which is another CKAN instance), the import of datasets from the OpenData Euskadi portal³ (in RDF format) and the publication of datasets provided in other websites, such as the World Bank⁴.

² <http://data.amsterdam.nl/>

³ <http://opendata.euskadi.eus/>

⁴ <http://data.worldbank.org/>

7. Training Manager

This section describes the Training Manager component which provides the training features for both experts and consumers.

7.1 Component Description

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 this set of required tools and functionality, was our choice as the core of the Training Manager with its big open-source community and its extendibility.

This component has a very important role in the portal since bringing HPC and GSS domains together is a rather new combination of technologies, this requires providing guidance for new end users aiming at adopting the technologies provided by the CoE. [1] For example, the end users 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 [1]. The training manager aims to answer these needs by hosting introductory or advanced courses, trainings and exercises for both HPC and GSS domains. As part of the near future plans, CoeGSS will use this component in order to provide some training courses such as one for describing the process followed by GSS applications and some introductory course about HPC basics and how it can be used for solving GSS related problems (i.e. HPDA and other tools that can run in HPC facilities).

Current version of the Training Manager is built upon Moodle 3.0 with the extensions; Questionnaire, Certificates and Configurable Reports. The extensions does not require any further installations apart from the default Moodle requirements.

Moodle can work with PostgreSQL, MySQL, MariaDB, Microsoft SQL Server and Oracle Database. Currently, our Moodle instance is using an installation of MySQL version 5.5.31

7.2 Component Configuration

The Training Manager (Moodle and its extensions) can be configured through Moodle's admin user interface which can be accessed with the admin user after the default Moodle installation.

Another configuration point for the Training Manager is the virtualhost configuration file. This is a standard Apache configuration file where the server name, alias and the port can be configured. [5]

7.3 Component Deployment

The current version of the Training Manager component is running in a virtual machine running Ubuntu 14.04 which is hosted at the High Performance Computing Center Stuttgart and can be accessed via the IP address 172.18.10.42 and port 80.

7.4 Component Testing

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

8. Community Manager

8.1 Component Description

Within this section of the deliverable, the CoeGSS Q&A portal is described. Q&A Software is online software that attempts to answer questions asked by users (Q&A stands for "question and answer"). Q&A software is frequently integrated by large communities to allow users in similar fields to discuss questions and provide answers to common and specialist questions.

The purpose of this component is to enable a common space where users may ask questions about CoeGSS, the usage of HPC (i.e. applications configuration, coding, access, etc.) and other questions related to how GSS may exploit HPC technologies. In order to support users, other users will be able to answer these questions and the CoeGSS partners will also support users with their knowledge and experience.

CoeGSS Q&A system is based on Askbot which is an open source question and answer oriented Internet forum started on July 2009. Askbot is a free and open source question and answer forum web application. It is similar to Stack Overflow and Yahoo Answers. The Askbot platform is written in Python and Django. Large open source software projects like Fedora, LibreOffice and OpenStack are using it as their main Q&A website.

Askbot is easy to configure, customize, and use its rich set of features. Askbot provides efficient question and answer knowledge management. It enables focused on-topic discussions. Best answers (according to voting) are shown first. Comments under questions and answers allow free follow-up discussions. Askbot offers advanced full text search, SEO and support for over 10 languages.

8.2 Component Configuration

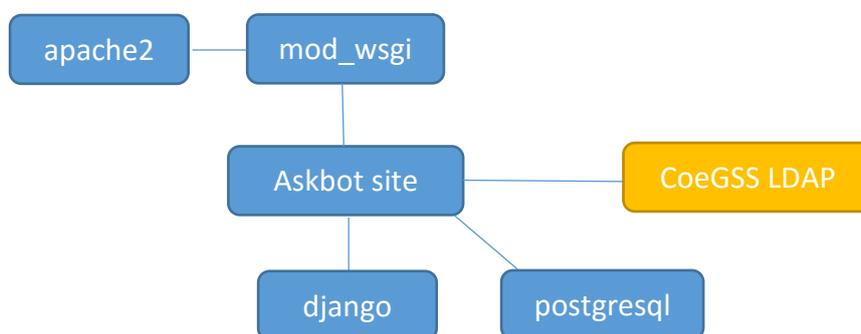


Figure 2. Subcomponents involved in the Community Manager

Apache / mod_wsgi

Serve the incoming http request using the mod_wsgi Python WSGI adapter, through a virtual host and SSL virtual host. The site vhost also contains url aliases to serve static content of the theme and all uploaded image files, including the site logo.

Apache virtual host on port 80 serves the Askbot site resources, only /admin (administration) and /account (logging) are redirected to SSL virtual host on port 443. SSL certificate is self-signed.

Askbot site/django

The Askbot django application is the website which allows asking and answering questions to the community. Askbot is installed and configured according to documentation at askbot.org and, therefore, it does not require any especial modification in its configuration parameters (besides the adaptation of URLs).

Askbot can be managed by using the admin user. We have configured an initial login, complementing the users' management done by LDAP.

Postgresql

A postgresql database hosts the content and dynamic site configuration. Postgres database 'coegss_qa' was created owned by Postgres's user 'coegss'.

8.3 Component Deployment

The first CoeGSS Q&A system prototype is running in a virtual machine hosted at the High Performance Computing Center Stuttgart. For providing the service, two central processing unit (CPU) cores and one Gigabyte of main memory are available. The system can be accessed via the public Internet Protocol (IP) address 141.58.0.66 on standard http and https ports 80 and 443. Second IP address is 172.18.10.45 and is used to access LDAP service via private network.

The Q&A system has been deployed by making use of the standard package system of Ubuntu and standard Python Management System. Some required packages needed to be installed before Askbot: python2.7, python-dev, postgresql, postgresql-server-dev-all, ldap_utils, libtiff4-dev, libjpeg8-dev, zlib1g-dev, libfreetype6-dev, liblcms2-dev, libwebp-dev, tcl8.5-dev tk8.5-dev ,python-tk, libsasl2-dev, libldap2-dev, libssl-dev, apache2, apache2.2-common apache2-mpm-prefork, apache2-utils, libexpat1, ssl-cert, libapache2-mod-wsgi. Also some required python modules needed to be installed: kombu, billiard, pycogp2, python-ldap.

The authentication bases on LDAP and local database. All other authentication modules like Google, Yahoo or OpenID providers are disabled. Askbot LDAP module bugs were corrected manually in python codes (one bug is known to Askbot developers and should be corrected in next Askbot release). After first login of LDAP user askbot asks for e-mail confirmation and creates account in local database referenced to LDAP account.

Askbot site is installed in /home/coegss/coegss_qa and symlinked to /var/www/coegss_qa.

URL to CoeGSS prototype is: <http://141.58.0.66/>

8.4 Component Testing

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

9. Infrastructure Services

This section describes briefly the hosting environment, and in particular the infrastructure services that guarantee the stable operation of the CoeGSS Portal. In general, all implementations follow the plan that has been introduced in Deliverable 5.9 (D5.9).

9.1 Hosting Description

All services of the first CoeGSS Portal prototype are hosted at the High Performance Computing Center Stuttgart when writing this deliverable. As highlighted in D5.9 [1], this infrastructure is meant to be of production quality, whereas an integration environment (more focused on development and components integration before going into production) will be replicated at the Poznan Supercomputing and Networking Center within the following months to implement release 2.

The CoeGSS infrastructure currently consists of seven virtual machines (VMs) that all fulfil different purposes. A detailed view about their individual configuration is presented in Table 2 below.

Name	CPU	Memory	Private IP	Public IP	Purpose
Infrastructure	2	4 GB	172.18.10.30	- / -	Infrastructure support
Jenkins	8	16 GB	172.18.10.40	- / -	Software testing
LDAP	2	4 GB	172.18.10.41	- / -	LDAP and Certificate Authority
Moodle	2	2 GB	172.18.10.42	141.58.0.65	Moodle services
Portal	4	16 GB	172.18.20.43	141.58.0.67	Portal services
Repository	4	16 GB	172.18.10.44	141.58.0.64	CKAN services
Support	2	1 GB	172.18.20.45	141.58.0.66	User support services

Table 2: Hosting environment specification

All virtual machines rely on a single base image of Ubuntu Linux 14.04, which was extended by a block storage to host the individual data. This approach allows maintaining and updating only one single image instead of seven individual ones.

Even if several components use some common base software such as Django or PostgreSQL, we have decided to keep the corresponding web servers and databases for each component in a separated way. Even if this solution consumes more resources, we consider that this configuration increases robustness (so a failure in a database will not affect several components but just one) and makes scalability easier (with the adequate granularity: per component). In any case, the consortium will analyse whether this solution works as expected, switching to a different configuration if necessary.

9.2 Hosting Configuration

Besides the individual service configurations, the entire infrastructure provides some basic features that are described below. In order to guarantee a stable and mature service operation, some common services such as availability and performance monitoring as well Domain Name System (DNS) or Email services need to be supported.

In order to monitor the infrastructure, the already mentioned Nagios monitoring system has been installed and configured on the infrastructure VM to check all the seven CoeGSS virtual machines. So far, *ping* services have been implemented as well as individual web service checks, which implement connection monitoring based on the Hypertext Transfer Protocol (HTTP). Finally, a LDAP check has been implemented to regularly access the system to check its authorization capabilities.

With regards to mail notifications, the entire infrastructure is supported by the redundant High Performance Computing Center Stuttgart mail relays that are configured to allow connections from the CoeGSS infrastructure. The CoeGSS services require access to mail functionality for sending dedicated user emails such as for instance, registration confirmations. Furthermore, email support is also mandatory in order to get information in case of system failures by the monitoring infrastructure. Based on this setup, not only duplicated work can be avoided, but also CoeGSS benefits from the evolved email infrastructure that caches emails even in case of emergency.

For stable Internet connections based on the DNS protocol, a replicated DNS server has been established in the CoeGSS infrastructure. This procedure enables very low latencies within the infrastructure and in addition, guarantees the availability of DNS servers for the whole environment in general.

9.3 Hosting Deployment

So far, all components have been deployed completely manually. As described in D5.9, an automated mechanism is foreseen that will be developed during the following weeks and months. As soon as this is ready, all components will be re-installed in the integration infrastructure in order to evaluate and validate the mechanisms.

9.4 Hosting Testing

The entire infrastructure is checked with the help of the Nagios framework, which is completely configurable and extendable. In addition to the automated tests, manual tests on the command line and via graphical user interfaces have been performed to ensure that all components work as expected.

10. Summary

This document has described the current implementation of the CoeGSS Portal, providing information about the components deployed, their configurations and the testing performed in order to guarantee that this first release is operative.

All the expected features for the first release are available (data management, training and users' community management) together with some required features such as authentication and the frontend for enabling easy access to all the functionalities. Therefore, the proposed architecture during the design phase is still valid according to the current implementation.

It is important to highlight that these 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 feedback from the pilots and as we identify other potential functionalities that could be useful for them.

Also, other additional features are present in the hosting environment (i.e. monitoring), so it will be possible to increase the Portal features in a near future. Therefore, during the next months, the CoeGSS Portal architecture will be improved, in order to adapt it for providing the features listed in the original roadmap.

References

- [1] C. Consortium, "Initial Portal Design," 2016.
- [2] C. Consortium, "First Report on Pilot Requirements," 2016.
- [3] T. O. K. Foundation, "CKAN - The open source data portal software," [Online]. Available: <http://ckan.org/>.
- [4] moodle, "Moodle Website," [Online]. Available: <https://moodle.org/>.
- [5] "Askbot - Create your own Q&A Forum," [Online]. Available: <https://askbot.com/>.
- [6] "Nagios Monitoring System," [Online]. Available: <https://www.nagios.org>.
- [7] "Apache Virtual Host documentation," [Online]. Available: <https://httpd.apache.org/docs/current/vhosts/>.

List of tables

Table 1: LDAP user scheme 12
 Table 2: Hosting environment specification 20

List of figures

Figure 1. Current high level architecture of the CoeGSS Portal..... 7
 Figure 2. Subcomponents involved in the Community Manager 17

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
HPDA	High Performance Data Analysis
HTTP	Hypertext Transfer Protocol
LDAP	Lightweight Directory Access Protocol
LDIF	LDAP Data Interchange Format
Q&A	Questions and Answers
SEO	Search Engine Optimization
VM	Virtual Machine
WP	Work Package