

D5.13 - FOURTH 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	October 1st, 2018
Dissemination level	Public
Nature	Report
Version	1.0
Work Package	WP5
Leading Partner	ATOS (F. Javier Nieto)
Authors	Burak Karaboğa, Michael Gienger
Internal Reviewers	Andreas Geiges, Paweł Wolniewicz
Keywords	Portal, Tools, CoE Services
Total number of pages:	18

Copyright (c) 2018 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://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	Burak Karaboga	ATOS	01.10.2018
Initial Template	Burak Karaboga	ATOS	01.10.2018
Review Draft (ver. 0.2)	Burak Karaboga, Michael Gienger	ATOS, HLRS	10.10.2018
Final	Burak Karaboga	ATOS	17.10.2018
Approved by	ECM		24.10.2018

Abstract

The fourth release of the CoeGSS Portal introduces some important improvements on top of the previous version. 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 only 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.

Abstract	3
1. Introduction.....	4
2. Implemented Portal Architecture	5
3. Frontend.....	6
4. Authentication & Authorization.....	14
5. Summary.....	16
References	17
List of tables.....	18
List of figures.....	18
List of Abbreviations	18

1. Introduction

This aim of this document is to describe the state of the fourth and the final release of the CoeGSS Portal by focusing only to the changes and updates to the previous release [1] and referring to it where necessary.

The document describes the CoeGSS Portal implementation, detailing several aspects about the implementation with the objective of facilitating the understanding about the implementation and acting as a guideline for those who may want to deploy the Portal components.

In order to do so, Section 2, describes the implemented features and remembers the followed architecture, while Sections 3 and 4 provide information about the implementation, configuration and the testing details of the components that are enhanced or introduced in this version. Finally, Section 5 summarizes the document and provides some conclusions.

The work towards the fourth and the final release has been focused on improving the functionality, usability and the integration of the HPC UI sub-component as well as upgrading the Single-sign-on (SSO) provider, FIWARE IDM [2] [3], to a newer version. This release also introduces some changes to the Matchmaking sub-component of the Frontend.

2. Implemented Portal Architecture

2.1 Implemented Features

The fourth, and the final, release of the CoeGSS portal is comprised of the following changes, new features and enhancements:

- The Single-Sign-On (SSO) component consisting of FIWARE IDM has been upgraded from version 5.4.4 to version 7.0.2. The new version features include but not limited to: support for HTTPS certificate chains, less resource consumption and an easier deployment process.
- HPC Job Submission component has been improved significantly to provide a user friendly HPC job submission process to the user.
- Matchmaking component has been updated with improvements on the user interface and bug fixes.

2.2 Implemented High Level Architecture

The fourth release of the CoeGSS Portal introduces some changes to the architecture described in the third release of the CoeGSS Portal [1] by introducing the HPC UI as a sub-component to the Frontend, which provides a user interface for the HPC Job Submission component, allowing the CoeGSS user to submit jobs to the HPC systems through the Portal UI (see: Figure 1).

The final release introduces a new version of the FIWARE IDM and some improvements over the Matchmaking Tool but both of these changes do not alter the architecture of the CoeGSS Portal with respect to the previous version.

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, once the Portal implementation is stable, its code will be made public. For further details about the implementation of any of the mentioned components, please refer to the next sections.

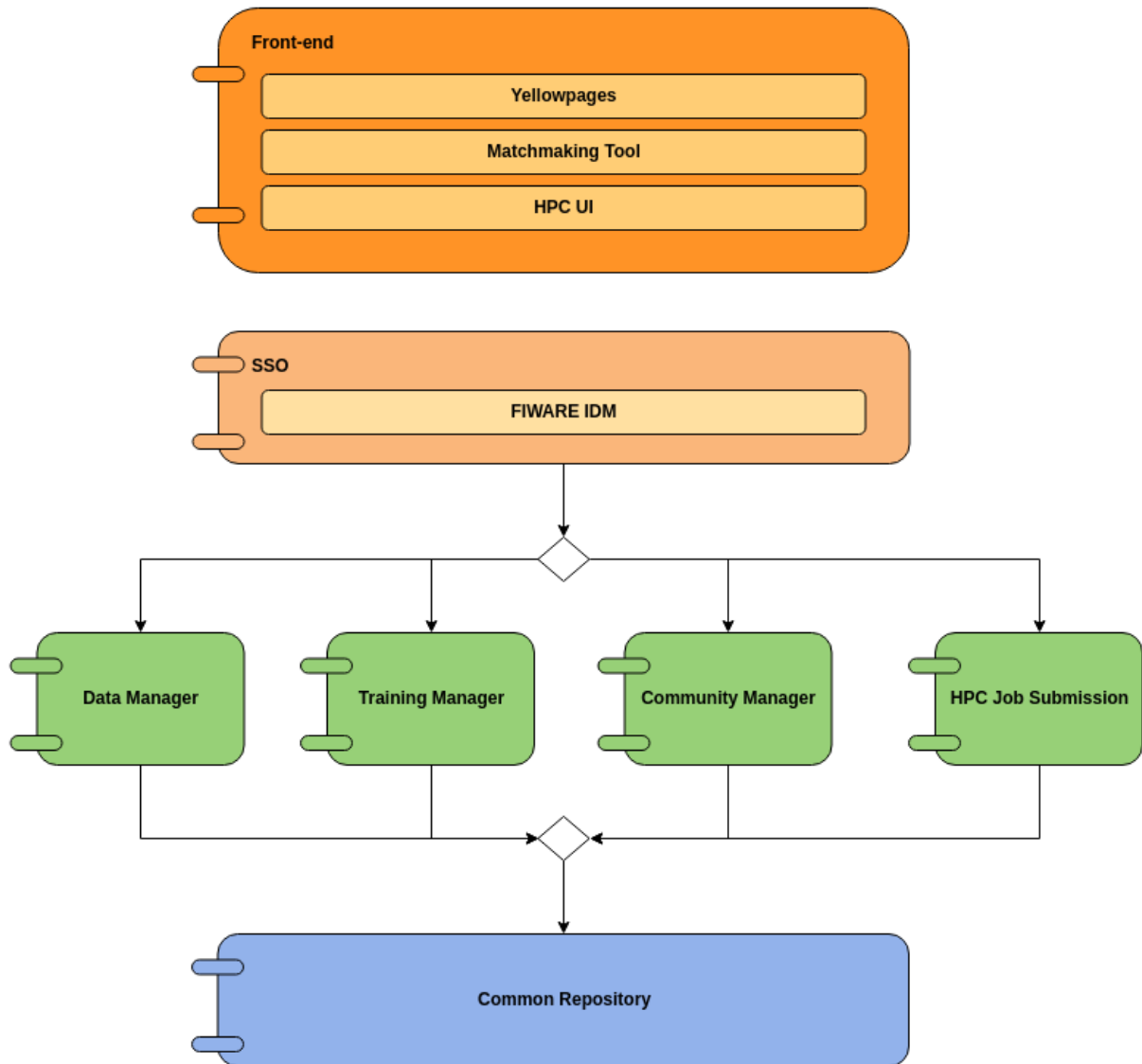


Figure 1. CoeGSS Portal High Level Architecture

3. Frontend

3.1 Component Description

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.

The final release of the portal introduces changes to the configuration and deployment of the component as well as a new sub-component called HPC UI and improvements to the Matchmaking sub-component.

3.2 Sub-Components

3.2.1 HPC UI

Final release of the CoeGSS Portal introduces a new sub-component called HPC UI which is responsible for providing a user interface for the HPC Job Submission component. So far, the blueprints and deployments were managed by the Cloudify Command Line Interface (CLI). HPC UI sub-component fills the gap between the CoeGSS Portal and the HPC Job Submission component, allowing the Portal users to manage their blueprints, deployments and job executions on HPC systems through the CoeGSS Portal.

In order to access the HPC UI, the user has to sign in to the CoeGSS Portal and select HPC Job Submission menu item from the Services menu (see: Figure 2)

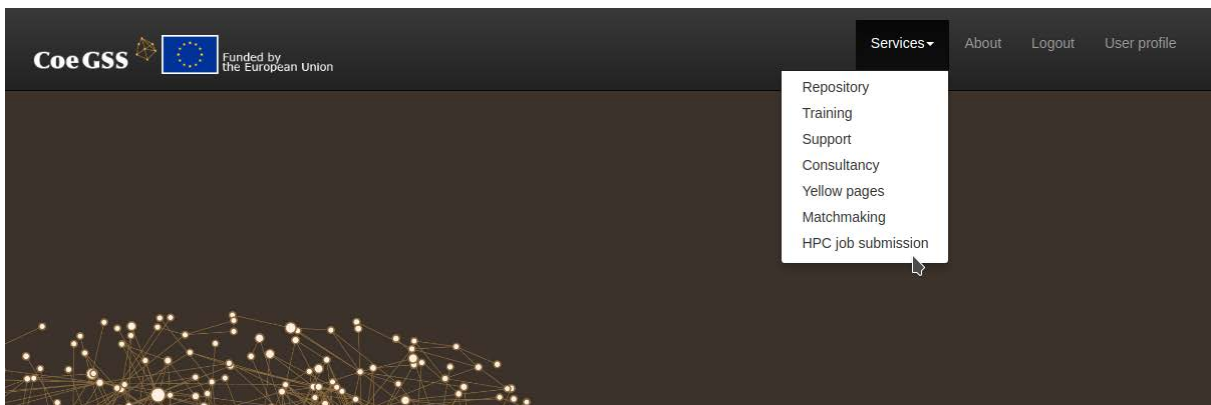


Figure 2. HPC UI menu access

HPC UI consists of two main tabs called *Blueprints* and *Deployments* and the user is greeted with the active Blueprints tab upon access to the sub-component (see: Figure 3).



Figure 3. Blueprints tab

The Blueprints tab consists of a table listing the blueprints created by the user, a form to create new blueprints and another form to deploy the selected blueprint to the HPC systems. Both of these forms are mapped to the buttons that can be found below the table: *Deploy selected blueprint* and *Create new blueprint*.

In order to create a new blueprint, the user should access the blueprint creation form (see: Figure 4) by clicking *Create new blueprint* button.

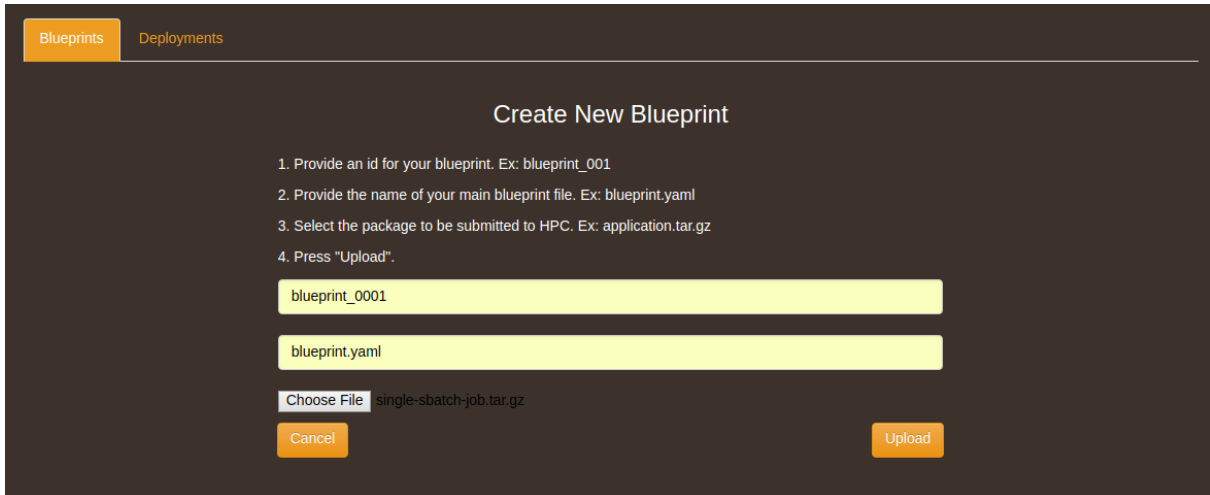


Figure 4. Blueprint creation form

Blueprint creation form expects a unique Blueprint ID, the name of the main blueprint file in the blueprint package and finally the blueprint package itself which must be a *.tar.gz* package containing the main blueprint file in its root and other script files if necessary. Once all the fields are filled with the required information the user can click on the *Upload* button which sends the necessary data to the HPC Job Submission component to create a blueprint ready to be deployed. Once the operation is complete, the form closes automatically and the user is presented with the blueprints listing updated with the recently uploaded package.

In order to deploy an already created blueprint, the user should access the deployment form (see: Figure 5) by clicking on the *Deploy selected blueprint* button. This button is activated only when a blueprint is selected from the blueprints listing.

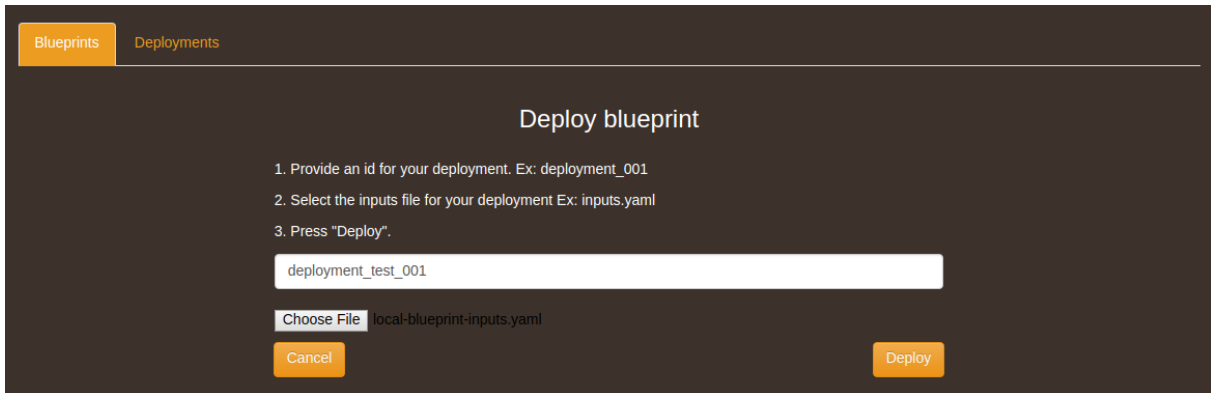
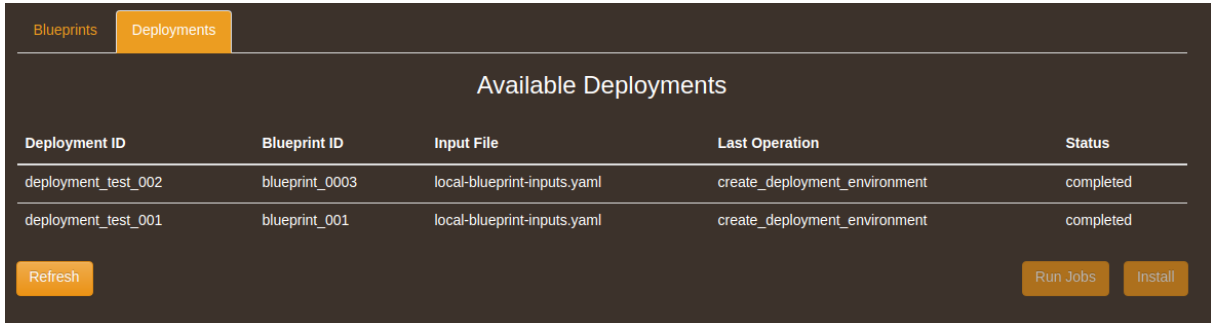


Figure 5. Deployment form

Deployment form expects a unique Deployment ID and an input file containing the target HPC clusters and user’s HPC credentials in YAML format. Once the deployment id and the input file are provided, the user can click on the Deploy button which takes the associated blueprint and creates a deployment for it on the target HPC cluster. Once the deployment operation is successfully started, the form is closed automatically and the user is directed to the Deployments tab where all the deployments created are listed, as seen in Figure 6.



Deployment ID	Blueprint ID	Input File	Last Operation	Status
deployment_test_002	blueprint_0003	local-blueprint-inputs.yaml	create_deployment_environment	completed
deployment_test_001	blueprint_001	local-blueprint-inputs.yaml	create_deployment_environment	completed

Figure 6. Deployments tab

The Deployments tab consists of a table listing the deployments created by the user, a button to refresh the view to check the deployment statuses, a button to install the deployments and finally another button to run the jobs of an installed deployment.

In order to install a deployment, the user should select a deployment from the list and click install. Once the installation is completed, the corresponding deployment in the list should have *install* as its Last Operation and *completed* as its status. The list can be refreshed by using the refresh button.

In order to run the jobs of an installed deployment, the user should select a deployment which has *install* as its Last Operation and *completed* as its status. If the selected deployment

has a different state then *Run Jobs* button will stay disabled. Once the running of the jobs is complete, the corresponding deployment in the list should have *run_jobs* as its Last Operation and *completed* as its status.

HPC UI uses the Python package *cloudify-rest-client* [4] in order to communicate with the HPC Job Submission component which is based on a deployment of Cloudify [5] extended by a custom plugin developed within the context of the CoeGSS Project. For a detailed description of the HPC Job Submission component please refer to the previous version of this deliverable [1].

3.2.2 Matchmaking Tool

The final release of the Portal and its sub-components target operating safety, therefore, the Matchmaking component does not introduce any new functionality for the final release. Consequently, software testing and improvements have been focused in order to guarantee functional reliability.

Within the past six months, the Portal with its new Identity Management System has been evaluated and optimised. The Matchmaking tool does not interact intensively with the other components, in general only a valid database connection is required in conjunction with authentication functionality, both of which are natively offered by the Django framework. However, especially the change of the Identity Management System affected the behaviour of the Matchmaking tool in such a way that refactoring and adaptation of the login management has been required. In this context, also various smaller bugs have been resolved that did not hamper the user experience, but contribute to a stable service offering.

Besides the functional adjustments, also the Graphical User Interface (GUI) of the Matchmaking tool has been improved. These improvements are based on the updated Cascade Style Sheets (CSS), but also a slightly modified button design in order to represent a more attractive solution to the end user. Figure 7 illustrates the final version of the Matchmaking tool and shows in particular the questions for generating matches as well as the resulting user interactions.

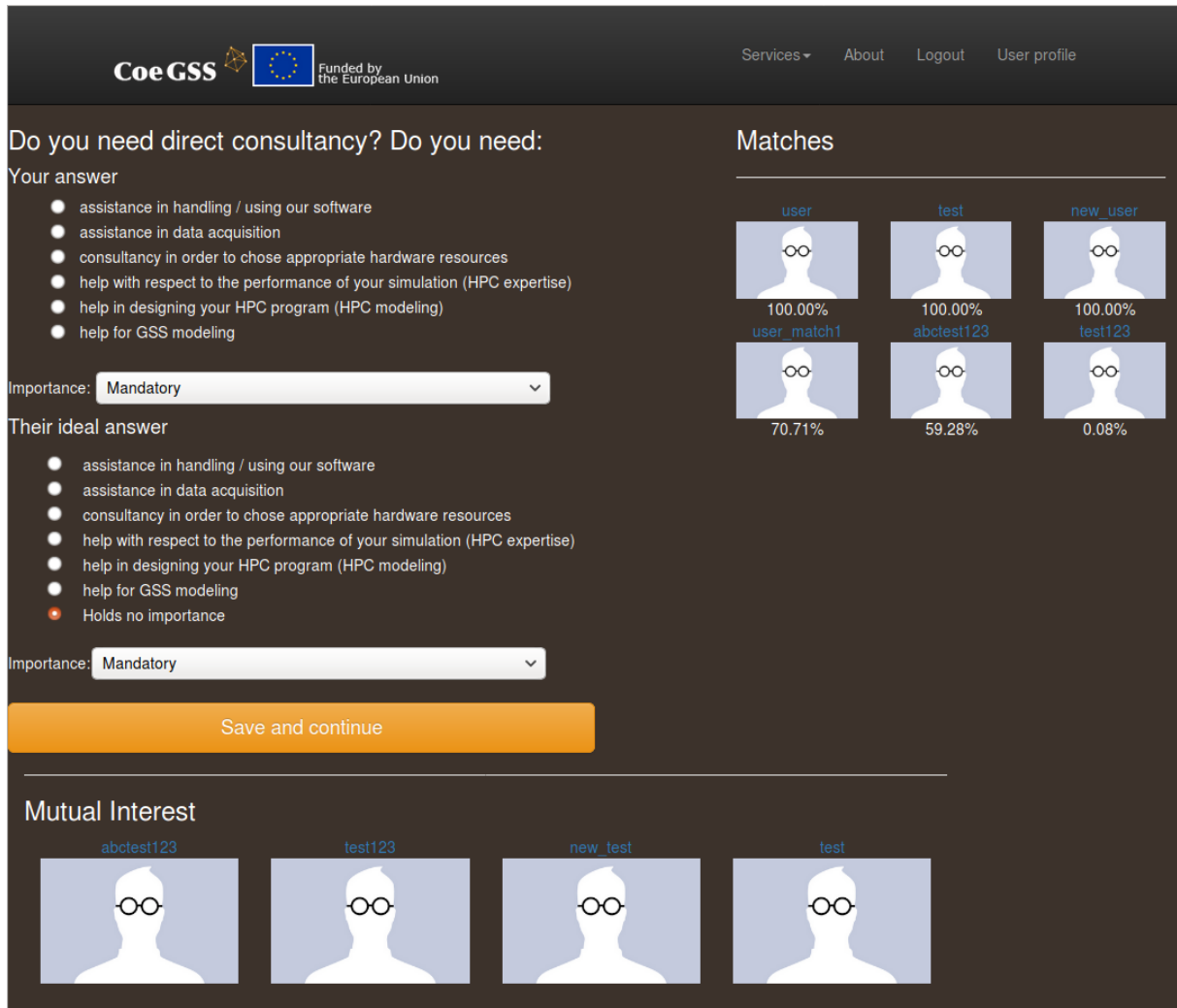


Figure 7. Matchmaking Tool

3.3 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 backend as well as FIWARE IDM [2] [3] configuration can be managed, Database connection can be configured and other Django apps can be added to the web application.

The current version of the CoeGSS portal introduces new configuration points which are required by the new version of the FIWARE IDM [2] [3]. The current settings for the configuration points listed above in *settings.py* are listed as follows:

```
BASE_DIR = os.path.dirname(os.path.dirname(os.path.abspath(__file__)))
STATIC_ROOT = os.path.join(BASE_DIR, 'static')
INSTALLED_APPS = [
    'sso.apps.SsoConfig',
```

```
'registration',
'django.contrib.admin',
'django.contrib.auth',
'django.contrib.contenttypes',
'django.contrib.sessions',
'django.contrib.messages',
'django.contrib.staticfiles',
'frontend',
'matchmaking',
'tags_input',
'social_django',
]
AUTHENTICATION_BACKENDS = [
    'sso.backends.keyrock.KeyrockOAuth2',
    'django.contrib.auth.backends.ModelBackend',
]
SOCIAL_AUTH_PIPELINE = (
    'social_core.pipeline.social_auth.social_details',
    'social_core.pipeline.social_auth.social_uid',
    'social_core.pipeline.social_auth.auth_allowed',
    'social_core.pipeline.social_auth.social_user',
    'social_core.pipeline.user.get_username',
    'social_core.pipeline.social_auth.associate_by_email',
    'social_core.pipeline.user.create_user',
    'social_core.pipeline.social_auth.associate_user',
    'social_core.pipeline.social_auth.load_extra_data',
    'social_core.pipeline.user.user_details',
)
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.postgresql_psycopg2',
        'NAME': 'coegss_portal',
        'USER': 'portal',
        'PASSWORD': 'PASSWORD',
        'HOST': 'localhost',
    }
}
MIDDLEWARE_CLASSES = [
    'django.middleware.security.SecurityMiddleware',
    'django.contrib.sessions.middleware.SessionMiddleware',
    'django.middleware.common.CommonMiddleware',
    'django.middleware.csrf.CsrfViewMiddleware',
    'django.contrib.auth.middleware.AuthenticationMiddleware',
    'django.contrib.messages.middleware.MessageMiddleware',
    'django.middleware.clickjacking.XFrameOptionsMiddleware',
    'social_django.middleware.SocialAuthExceptionMiddleware',
    'coegss_portal.middleware.RedirectOnCancelMiddleware'
]
TEMPLATES = [
    {
        'BACKEND': 'django.template.backends.django.DjangoTemplates',
        'DIRS': [TEMPLATE_PATH],
        'APP_DIRS': True,
```

```
'OPTIONS': {
    'context_processors': [
        'django.template.context_processors.debug',
        'django.template.context_processors.request',
        'django.contrib.auth.context_processors.auth',
        'django.contrib.messages.context_processors.messages',

        'social_django.context_processors.backends',
        'social_django.context_processors.login_redirect',
    ],
},
],
}],
]
EMAIL_BACKEND = 'django.core.mail.backends.smtp.EmailBackend'
EMAIL_HOST = 'localhost'
EMAIL_PORT = 25
DEFAULT_FROM_EMAIL = 'no-reply@portal.coegss.hlrs.de'
CONTACT_EMAIL = 'gienger@hlrs.de'
EMAIL_HOST_USER = ''
EMAIL_HOST_PASSWORD = ''
EMAIL_USE_TLS = False

LOGIN_URL = 'login'
LOGOUT_URL = 'logout'
LOGOUT_REDIRECT_URL = '/frontend'
LOGIN_REDIRECT_URL = 'home'

CKAN_URL = config('CKAN_URL')

ORCHESTRATOR_HOST = config('ORCHESTRATOR_HOST')
ORCHESTRATOR_USER = config('ORCHESTRATOR_USER')
ORCHESTRATOR_PASS = config('ORCHESTRATOR_PASS')
ORCHESTRATOR_TENANT = config('ORCHESTRATOR_TENANT')

FIWARE_IDM_ENDPOINT = config('FIWARE_IDM_ENDPOINT')
SOCIAL_AUTH_FIWARE_KEY = config('SOCIAL_AUTH_FIWARE_KEY')
SOCIAL_AUTH_FIWARE_SECRET = config('SOCIAL_AUTH_FIWARE_SECRET')

SOCIAL_AUTH_LOGIN_ERROR_URL = '/login_error/'
SOCIAL_AUTH_LOGIN_REDIRECT_URL = '/'
SOCIAL_AUTH_RAISE_EXCEPTIONS = False
```

3.4 Component Deployment

This final 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 [https:// portal.coegss.hlrs.de](https://portal.coegss.hlrs.de)

3.5 Component Testing

As the previous version [1], the final version of the component has been tested manually.

4. Authentication & Authorization

Within this section of the deliverable, the CoeGSS authentication and authorization mechanism, which is handled by a deployment of FIWARE IDM [2] [3] 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.

4.1 Component Description

In the previous release, this component was composed of a deployment of FIWARE IDM version 5.4.4 which depended on Openstack Keystone and Horizon [6] [7] [8]. This release upgrades the FIWARE IDM to version 7.0.2 which is fully implemented in Node.js, reducing the resource requirements, improving the performance and introducing the HTTPS certificate chain support.

Besides the improvements over the performance, there have been some minor changes from the user perspective and for this reason; the authentication flow has been re-described below.

The new authentication flow based on the new version of FIWARE IDM deployment is as follows:

1. User accesses a restricted resource on CoeGSS Portal and the portal backend makes an authorization request to FIWARE IDM
 - a. If the user is has not logged in yet, they are redirected to FIWARE IDM and are asked for their login credentials
 - b. If the user is already logged in, or the user successfully logs in, FIWARE IDM creates a session, creates an authorization token for the user and redirects them back to CoeGSS portal
2. CoeGSS Portal validates the token, creates a session for the user and sends them to the requested resource

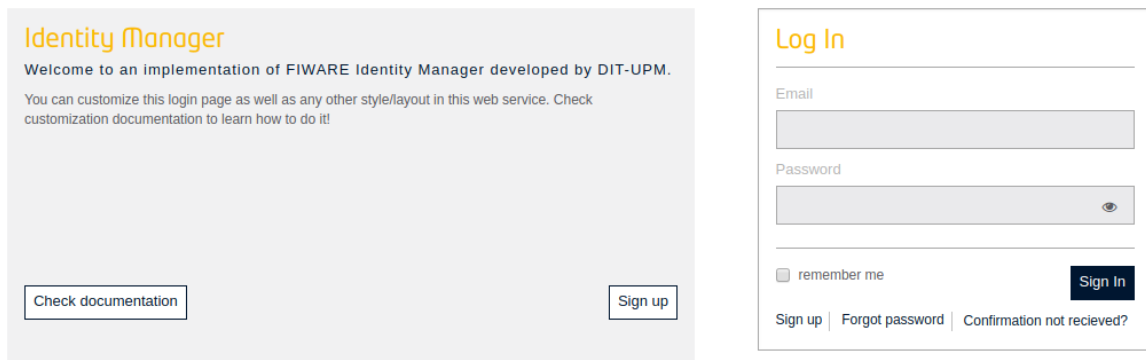


Figure 8. FIWARE IDM login page

4.2 Component Configuration

The new version of the component introduces a new method of configuration where all the required configuration parameters are taken from an `.env` file in the same directory as the `docker-compose.yaml` file which is used to deploy an instance of FIWARE IDM.

All the values that can be configured via `.env` configuration file are listed below. Please note that all values denoted in `<...>` should be replaced with actual values and all values denoted in `#...#` are optional.

```
IDM_HOST=idm.coegss.hlrs.de
IDM_PORT=
IDM_HTTPS=true
IDM_CONFIG=./config.js

SMTP_HOST=172.18.10.43
SMTP_PORT=25
SMTP_SECURE=false
SMTP_FROM=no-reply@portal.coegss.hlrs.de
SMTP_USER= #smtp_user#
SMTP_PASS= #smtp_password#

MYSQL_ROOT_PASSWORD= <mysql_password>
MYSQL_DATA=./mysql_data
```

4.3 Component Deployment

FIWARE IDM is hosted by its own virtual machine at the High Performance Computing Centre Stuttgart. The virtual machine hosting FIWARE IDM is based on a standard Ubuntu 16.04 installation which is extended to provide the ability to store Secure Shell (SSH) public keys. The component is publicly reachable via URL <http://idm.coegss.hlrs.de>

4.4 Component Testing

This component has been tested manually.

5. Summary

This document has described the fourth and the final release 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 [1].

Initially, several open source products from FIWARE project was planned to be evaluated for integration to the CoeGSS Portal such as: WStore, Repository GE and the Identity Management. During the lifetime of the project, WStore and Repository GE were deprecated and replaced by a single application called FIWARE Marketplace yet the final release of CoeGSS Portal does not integrate this component. FIWARE IDM was successfully integrated to the CoeGSS Portal as a single sign-on solution reducing the development overhead significantly.

The goal of the task 5.3 of WP5 was to deliver the CoeGSS Portal that connects and hosts the services as well as the resources of the project. Specifically, the CoeGSS Portal was planned to provide: an efficient and secure platform covering the whole service lifecycle, a direct access point for getting support and fostering community building. The current release of the portal meets all these goals by bringing the CoeGSS service together to a single point of access in a secure manner.

References

- [1] C. Consortium, *D5.12 Third Portal Release*, 2018.
- [2] "FIWARE," [Online]. Available: <https://www.fiware.org/>. [Accessed October 2018].
- [3] "FIWARE IDM," [Online]. Available: <https://catalogue-server.fiware.org/enablers/identity-management-keyrock>. [Accessed October 2018].
- [4] "Cloudify Rest Client," [Online]. Available: <https://cloudify-rest-client.readthedocs.io/en/3.3/>. [Accessed October 2018].
- [5] Cloudify. [Online]. Available: <https://cloudify.co/>. [Accessed October 2018].
- [6] "Openstack," [Online]. Available: <https://www.openstack.org/>. [Accessed October 2018].
- [7] "Openstack Keystone," [Online]. Available: <https://docs.openstack.org/keystone/pike/>. [Accessed October 2018].
- [8] "Openstack Horizon," [Online]. Available: <https://docs.openstack.org/horizon/pike/>. [Accessed October 2018].

List of tables

None

List of figures

Figure 1. CoeGSS Portal High Level Architecture	6
Figure 2. HPC UI menu access	7
Figure 3. Blueprints tab	7
Figure 4. Blueprint creation form.....	8
Figure 5. Deployment form	9
Figure 6. Deployments tab	9
Figure 7. Matchmaking Tool	11
Figure 8. FIWARE IDM login page.....	15

List of Abbreviations

CLI	Command Line Interface
CSS	Cascaded Style Sheets
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
IDM	Identity Manager
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