

**EGI-Engage**

Fully Integrated MoBrain web portal

D6.4

|  |  |
| --- | --- |
| **Date** | 10 February 2016 |
| **Activity** | WP6 |
| **Lead Partner** | GNUBILA |
| **Document Status** | DRAFT |
| **Document Link** | <https://documents.egi.eu/document/2664> |

Abstract

The MoBrain Web Portal (<http://mobrain.egi.eu>) is the entry point to the MoBrain Competence Center, it is meant to aggregate information from the WeNMR and N4U communities and link to relevant content on the respective sites and other related projects such as INSTRUCT and the WestLife Virtual Research Environment.

The MoBrain portal has been deployed as a Liferay-based portal in the EGI Federated Cloud. Its deployment and configuration are managed using the Software Configuration Management tool Puppet and is called from a contextualization script launched at the end of the Virtual Machine creation. This portal provides the basis from which the integration/aggregation process will be continued under EGI-Engage with input from the various MoBrain partners.

**COPYRIGHT NOTICE**



This work by Parties of the EGI-Engage Consortium is licensed under a Creative Commons Attribution 4.0 International License (http://creativecommons.org/licenses/by/4.0/). The EGI-Engage project is co-funded by the European Union Horizon 2020 programme under grant number 654142.

**DELIVERY SLIP**

|  |  |  |  |
| --- | --- | --- | --- |
|  | ***Name*** | ***Partner/Activity*** | ***Date*** |
| **From:** | Baptiste Grenier | GNUBILA | 14/01/2016 |
| **Moderated by:** | Małgorzata Krakowian | EGI.eu/WP1 |  |
| **Reviewed by** | Sergio Andreozzi  Johan Montagnat | EGI.eu/WP2  FR CNRS IDGC/PMB |  |
| **Approved by:** | AMB and PMB |  |  |

**DOCUMENT LOG**

|  |  |  |  |
| --- | --- | --- | --- |
| ***Issue*** | ***Date*** | ***Comment*** | ***Author/Partner*** |
| **v.1** | 02/02/2016 | Revision | Alexandre Bonvin |
| **v.2** | 02/02/2016 | Revision | Marco Verlato |
| **v.3** | 05/02/2016 | Revision | Jérome Revillard |
| **v.4** | 10/02/2016 | Revision | Jérome Revillard / Alexandre Bonvin |
|  |  |  |  |

**TERMINOLOGY**

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

**Contents**

[1 Introduction 5](#_Toc442874448)

[2 Service architecture 5](#_Toc442874449)

[2.1.1 High-Level Service architecture 5](#_Toc442874450)

[2.1.2 Integration and dependencies 6](#_Toc442874451)

[3 Release notes 12](#_Toc442874452)

[4 Feedback on satisfaction 14](#_Toc442874453)

[5 Future plans 14](#_Toc442874454)

**Executive summary**

The MoBrain Web Portal[[1]](#footnote-2) is the entry point to the MoBrain Competence Center, it is meant to aggregate information from the WeNMR and N4U communities and link to relevant content on the respective sites and other related projects such as INSTRUCT and the WestLife Virtual Research Environment.

The MoBrain portal has been deployed as a Liferay-based portal in the EGI Federated Cloud. Its deployment and configuration are managed using the Software Configuration Management tool Puppet and is called from a contextualization script launched at the end of the Virtual Machine creation. This portal provides the basis from which the integration/aggregation process will be continued under EGI-Engage with input from the various MoBrain partners.

In this document first a short introduction table is presented. Then the service architecture is described provided. Section tree shows the current status of the web portal. Finally, feedback and future plans are briefly be presented.

# Introduction

|  |  |
| --- | --- |
| **Tool name** | *MoBrain Web Portal* |
| **Tool url** | [*https://mobrain.egi.eu*](https://mobrain.egi.eu) |
| **Tool wiki page** | *NA* |
| **Description** | *Portlet-based Web portal by the MoBrain Competence Centre. The portal is the entry point, it will aggregate information from the various web portals of the WeNMR, INSTRUCT and N4U research communities.* |
| **Customer of the tool** | * *Researchers* * *Public* |
| **User of the service** | *TBD – but mainly MoBrain CC partners / users* |
| **User Documentation** | *NA* |
| **Technical Documentation** | *NA* |
| **Product team** | *Gnúbila* |
| **License** | *NA* |
| **Source code** | *NA* |

# Service architecture

The MoBrain Web portal is built on multiple mature, robust and well known technologies. Deployment and configuration of the web portal on the cloud is fully automated enforcing a validated and reproducible deployment procedure.

### High-Level Service architecture

The MoBrain web portal is a Liferay portal instance deployed in the EGI.eu cloud using tools compatible with the Open Cloud Computing Interface (OCCI) that provides abstraction over multiple cloud providers. In fact during the deployment tests, the web portal was first published at INFN-PADOVA-STACK cloud site that was an OpenStack-based IaaS and is now in production at CESENT-MetaCloud, build using OpenNebula.

The Liferay portal is configured to be used by the MoBrain Competence Centre, using the mobrain.egi.eu domain name. A certificate for mobrain.egi.eu is also used to provide access to the portal over HTTPS only, thus the portal is accessible at https://mobrain.egi.eu.

A MoBrain-specific theme is deployed allowing quickly identifying and efficiently presenting the MoBrain Competence Centre.

### Integration and dependencies

The MoBrain Web Portal has been deployed in a Virtual Machine running at CESNET MetaCloud, one of the sites integrated into the EGI Federated Cloud. The resources available to the enmr.eu Virtual Organization (VO) have been found using the EGI Applications database.

An overview of the deployment is shown by Figure 1.

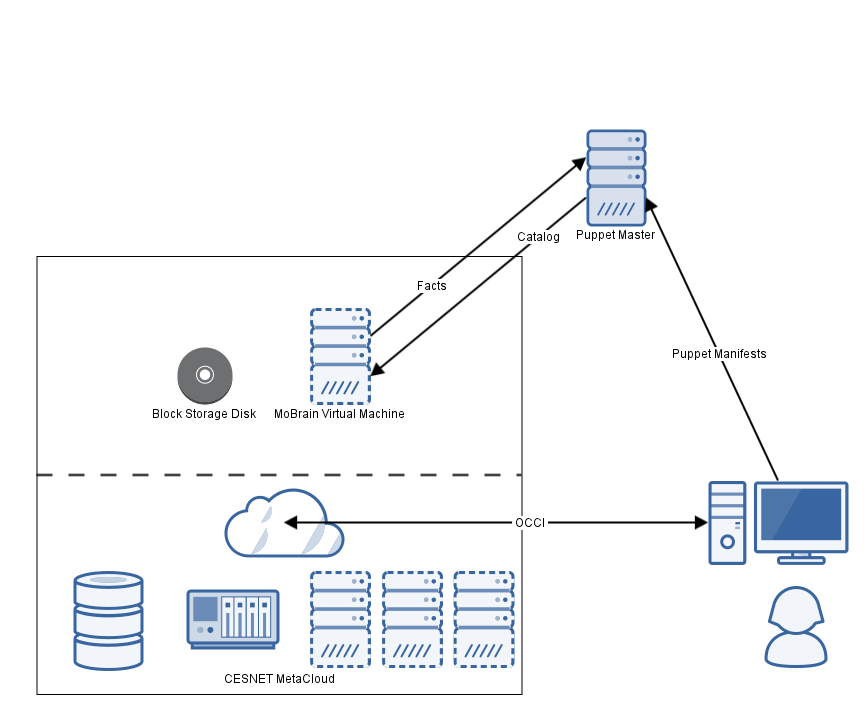


Figure 1: Deployment overview

#### EGI federated cloud: CESNET MetaCloud site

The EGI federated cloud infrastructure is an e-infrastructure federating resources from multiple academic private clouds accessible using open standards such as OCCI. Multiple sites provide cloud resources (mostly storage, computing and network resources) shared among multiple Virtual Organizations, allowing their users to easily manage it using well-known interfaces and compatible tools.

Thanks to the OCCI compatibility, multiple heterogeneous infrastructures (some are OpenStack-based, others are OpenNebula-based) can be managed using a common set of tools, such as the rOCCI client whose usage is documented in the HOWTO11 on the EGI.eu wiki[[2]](#footnote-3).

CESNET-MetaCloud site was chosen as it provides the required storage, computing and network resources along with the desired OS template as well as compatible rules of usages for public services, allowing long lived and public services to be operated.

#### EGI Applications Database

The EGI Applications Database (AppDB)[[3]](#footnote-4) is a central service allowing describing and sharing information about software solutions, people and publications in the context of EGI and the Federated Cloud.

The MoBrain Web portal is published on resources that were found using the AppDB: once the running site was chosen, the AppDB was used to find the OS and resource templates available on this site.

#### The resource template: large

The resource template is the template defining what will be the hardware resources available to the Virtual Machine, it defines the number of Virtual CPUs and memory that will be assigned to the Virtual Machine instance.

As the Liferay portal is java-based and uses a local MySQL database, it will benefit from running on a powerful Virtual Machine having multiple cores and a large amount of memory, so the large resource template was chosen as it is powered by 4 CPU cores and 8 GB of memory.

#### EGI Federated Cloud Storage: block storage

The storage block is a 50GB virtual disk that is mounted in the Virtual Machine and where Liferay and MySQL data is stored. That storage block can be mounted on another Virtual Machine if needed. (To use a more recent OS template or to use a more powerful resource template).

The storage block is managed via the OCCI interface and is seen as a local disk. It should be partitioned and mounted in exactly the same way.

#### The OS Template: EGI CentOS 6 base image

The OS template is an image of an Operating System installation. It has to be deployed on a resource template providing a predefined set of computing resources.

The EGI CentOS 6 base image is an Operating System template created and endorsed by the EGI team. It provides a clean and solid base image that can be used to deploy services confidently on the EGI Federated Cloud.

The EGI CentOS 6 OS template has been chosen as it provides a mature, stable, tested and well known RedHat-based Operating System with an extremely long support period ending in November 2020. As this image was created and validated by EGI.eu, it offers some quality guaranties.

#### Software dependencies

As shown in Figure 2, multiple components are used to publish and maintain the MoBrain web portal.

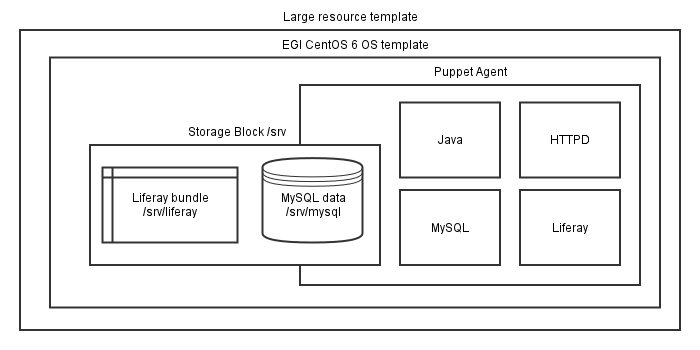
**

Figure 2: Overview of MoBrain web portal

The MoBrain portal is build using the following software components:

* Liferay Portal 6.2 bundled with Apache Tomcat 7 servlet and JSP container
* MySQL 5.1 relational database
* Apache HTTPD 2.2 http server
* OpenJDK 1.7 providing Java 7 execution environment
* Puppet 4 for automating the deployment and configuration of the portal

The last piece of software used is backupninja, a set of shell scripts allowing to create incremental backups of the MySQL database and Liferay files on a server operated by gnúbila and located outside of the CESNET cloud site. The backup data could be used to quickly deploy another instance of the MoBrain portal in case of trouble.

#### Deployment plan

When possible the deployment has been automatized, only two manual steps remain to be done using the rOCCI tool:

* Creation of a storage block using the EGI Federated Cloud Storage.
* Creation and launch of a Virtual Machine using the previously selected OS and resource templates and accessing the storage block.

When the Virtual Machine is started, a cloud-init contextualization script is run and is responsible for bootstrapping the MoBrain portal.

##### Creation of the block storage device

and t

The block storage device will be mounted in the Virtual Machine filesystem on /srv, and the Liferay bundle and MySQL files will be stored under /srv, allowing to easily separate the data from the operating system.

To create a 50 GB storage block titled mobrain\_data using rOCCI, type:

occi -n x509 -x /tmp/x509up\_u0 -X -e https://carach5.ics.muni.cz:11443 \

--action create \

--resource storage \

--attribute occi.storage.size='num(50)',occi.core.title=mobrain\_data

This command returns the storage ID, in the form of /storage/<id>, usable to reference the storage block when linking it on the virtual machine.

##### Creation of the Virtual Machine

Once the Storage Block has been created, the Virtual Machine, that will be using the storage block and using a contextualization script to bootstrap the installation, can be instantiated. The Virtual Machine is created using the OS and resource templates found using the EGI Applications Database.

The rOCCI command to create the Virtual Machine, link the storage block and run the contextualization script is:

occi -n x509 -x /tmp/x509up\_u0 -X -e https://carach5.ics.muni.cz:11443 \

--action create \

--resource compute \

--mixin 'os\_tpl#uuid\_enmr\_egi\_XXXXX\_fedcloud\_warg\_135' \

--mixin 'resource\_tpl#large' \

--attribute occi.core.title="mobrain" \

--link /storage/<id>

--context user\_data="file:///data/mobrain-bootstrap.cfg"

where /storage/<id> is the identifier of the block storage device created previsouly and XXXXX stands for …?

Three points are noticeable in this command line example. The first one is the usage of the mixin parameter to specify the OS template and the resource template specifying the hardware resource and Operating System image to be used by the Virtual Machine. The second important point is the linking of the previously created storage block, which will allow accessing the storage block as a local disk. The last important point is the usage of a contextualization script available locally: mobrain-bootstrap.cfg. This script will be responsible for bootstrapping the MoBrain Web Portal.

##### Bootstrap using cloud-init contextualization script

The cloud-init script mobrain-bootstrap.cfg, used at the end of the Virtual Machine creation to configure the MoBrain Web Portal is in fact not the script that will do all the hard work. This script will only do the following:

* Creating a cloudadm administrator user, allowing him to execute all commands using sudo, and associating to it the ssh keys of gnúbila’s administrators, allowing them to connect securely on the Virtual Machine over ssh.
* Creating and formatting a partition on the block storage device, only if not already done.
* Mounting the storage partition on /srv.
* Downloading a bootstrap.sh script from gnúbila’s server.
* Launching the bootstraps.sh script using the appropriate parameters: the role (liferay\_mobrain) and location (cesnet) of the Virtual Machine.

Here we can see that even if at the end of the contextualization script the Virtual Machine will be configured to run the MoBrain Web Portal, the bootstrap code is applied by the bootstrap.sh script.

The bootstrap.sh script sets up the required components: mostly installing and running the puppet agent to deploy the configuration on the server.

Puppet is an IT automation ecosystem, enforcing the state of an infrastructure using a declarative language. Software Configuration Management allows to specify what software needs to be installed on a particular server and how it should be configured. Puppet is following an agent/master approach: an agent runs on the node (server, virtual machine…), this agent collects information about the local node (the so called facts, ranging from network and kernel information to Operating System details and the location and role facts mentioned earlier) and connects securely, to a master (puppet.gnubila.fr for the time being). Then the master using the information provided by the agent will build and send back to the agent a catalog of resources with their configuration and interdependencies. Eventually the agent will apply the catalog on the local node, putting every resource (package, file, service…) in the state described in the catalog. So once the puppet run is over, the state of the virtual machine should reflect the desired state that was described and configured on the puppet master.

The puppet configuration manages the configuration of everything that needs to be configured, from the hostname to the firewall rules and configuration of the installed applications.

The puppet agent will download a Liferay tarball, deploy it under /srv on the block storage device, install MySQL, install and configure an Apache HTTP server that will do HTTPs in front of the Apache Tomcat servlet container hosting the Liferay portal, bootstrap the initial Liferay configuration. It will also setup backups and apply some security-related configuration, by blocking all unnecessary ports, adjusting SSH configuration and so on. Once puppet agent runs will be over, the MoBrain Web Portal will be deployed and setup on the Virtual Machine.

At this point the only missing thing is connecting the MoBrain Web Portal domain name, mobrain.egi.eu to the IP address that was assigned to the Virtual Machine upon creation.

##### Updating the public IP of the MoBrain Web Portal

During the Virtual Machine creation a public IP is added to the host and thanks to the use of a CNAME alias the mobrain.egi.eu configuration can be quickly updated without having to modify the egi.eu zone: mobrain.egi.eu is an alias point to mobrain.gnubila.fr. The mobrain.gnubila.fr domain points to the public IP and can be quickly updated on deployment using a homemade script and the appropriate API provided by OVH, the registrar of the gnúbila.fr domain.

Updating the domain name with the public IP of the compute resource:

curl -L -O https://web.gnubila.fr/ovh/update\_ip.sh

chmod +x update\_ip.sh

./update\_ip.sh XXXXX XXXXX XXXXX mobrain.gnubila.fr XXX.XXX.XXX.XXX

Here the first three XXXXX blocks are the secret API key and the last XXX.XXX.XXX.XXX argument is the IP address that was added to the node, and that can be retrieved using the OCCI interface and the describe action on the compute resource (the Virtual Machine).

Once this has been done, <https://mobrain.egi.eu> can be used to access the MoBrain Web Portal.

# Release notes

The web site has been populated with some content and is organised with the following sections:

* Main Page: This page contains an overview of the project as well as a pointer to the N4U and WeNMR communities. Last but not least, a Blog module has been added in order to be able to publish news about the project.
* Partners: contains the name of the project partners
* Documentation: For the moment, this section is decomposed in two sub-section:
  + Technical: will contain project specific technical documentation
  + Project: contains the project description as well as the different tasks and deliverables planned during the project.
* Contact Us: This is a pointer to the community contacts as well as support pages.



Figure 3: Web site main page overview



Figure 4: Contact page

# Feedback on satisfaction

Some project partners have been using the editing tools to successfully add and modify content.

Also, an email has been circulated to the project mailing list to ask feedback on the web site content.

# Future plans

The content of the web site will grow regularly. Also, on user requests, some new modules can be added to add more functionality (i.e wiki, forum etc…).

1. <http://mobrain.egi.eu> [↑](#footnote-ref-2)
2. <https://wiki.egi.eu/wiki/HOWTO11> [↑](#footnote-ref-3)
3. <https://appdb.egi.eu> [↑](#footnote-ref-4)