Secure Provisioning of Cloud Services
based on SLA Management

SPECS Project - Deliverable 1.6.1

SPECS Testbed Beta

Version 1.1
15 February 2016

The activities reported in this deliverable are partially supported
by the European Community's Seventh Framework Programme under grant agreement no. 610795.

Deliverable information

Deliverable no.:	D1.6.1
Deliverable title:	SPECS Testbed Beta
Deliverable nature:	Prototype
Dissemination level:	Public
Contractual delivery:	18 February 2016
Actual delivery date:	18 February 2016
Author(s):	Silviu Panica (IeAT)
Contributors:	Massimiliano Rak (CeRICT), Valentina Casola (CeRICT)
Reviewers:	Madalina Erascu (IeAT), Umberto Villano (CeRICT)
Task contributing to the deliverable:	T1.6
Total number of pages:	17

Executive summary

This deliverable is associated with the prototype implementation of the SPECS Testbed (Task 1.6) whose goal is to provide and maintain a running platform to host the SPECS Services.
The SPECS Testbed provides the SPECS Enabling Platform as already illustrated in Deliverables D1.2, D1.1.1 and D1.1.2; it offers all the components needed to provide and manage all SPECS services in their life cycle.
It is currently deployed in the servers of the IeAT partner, being accessible to all partners during the project life.
The design of the SPECS Enabling Platform module is presented in details in the deliverable D1.1.2, the goal of this document is to: (i) report the status of implementation activities, (ii) give the initial instructions on how to access the testbed and use it.
Finally, a plan for the future implementation activities is discussed.

Table of contents

Index of figures

Index of tables

1. Introduction

The *SPECS Testbed* represents the runtime environment and the supporting infrastructure where the *SPECS SLA Platform and Core Services* can be deployed and started. In the project architecture, this environment is grouped under the name of *Enabling Platform*.
The Enabling Platform implementation consists of two main parts: the *SPECS Enabling Platform* and the *SPECS Testbed supporting infrastructure*.

The *SPECS Enabling Platform* is a software stack (operating system, associated bootstrap services, etc.) that enables the functionalities of the SPECS Platform (software packages, components, services,....). It consists of the following technological components:

- *a Custom Operating System* (mOS) (the underlying operating system started on the infrastructure),
- *mOS Bootstrapper* (a collection of services able to customize a mOS in order to match the deployment requirements),
- *Cluster Manager* (a resource allocator and deployment service able to easily deploy and configure a collection of resources).

These components are briefly described in the following sections with the primary goal of describing the status of activities, the link to repositories, the installation prerequisites and their configuration and usage. It should be noted that, unlike the other SPECS components, the Enabling Platform sources will not be provided on Bitbucket, but it will be available as hosted services delivered by the two physical infrastructures deployed in the data center of IeAT and, at the time of the next iteration of this report, also in the data center of the EMC partner.

2. Relationship with other deliverables

The work presented in this document is related mainly to activities of all Tasks inWP1. In particular, in deliverable D1.2 we discussed the requirements that the Enabling Platform should respect, and in deliverables D1.1.1 and D1.1.2 we motivated and designed the solution to provide an independent Enabling Platform to manage the whole life cycle of all SPECS Services.

The next iteration of this deliverable (D1.6.2 at month 30) will present the final set of services needed to deploy and manage the SPECS Platform, and a second different Testbed in order to prove the service independence.
Figure 1shows the relationships described above. The platform described in this deliverable enables the execution of all SPECS services. However, for readability's sake we do not report here the complete list of all implementation tasks (and related deliverables) that use the services provided by the Enabling Platform.

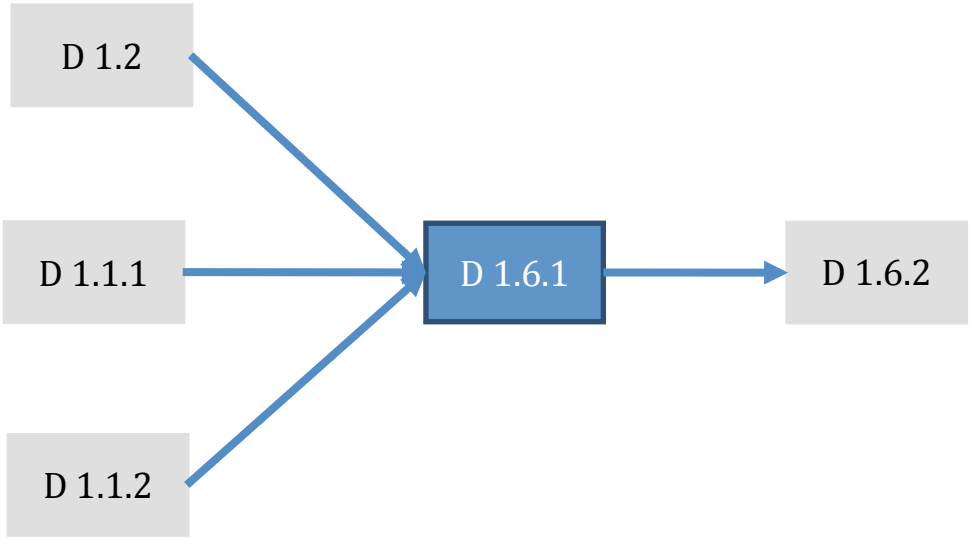

Figure 1.Relationships with other deliverables

3. SPECS Enabling Platform

The Enabling Platform implementation consists of two main parts: the *SPECS Enabling Platform* and the *SPECS Testbed supporting infrastructure*.

The *SPECS Enabling Platform* is a software stack that enables the SPECS platform deployment and its management at infrastructure level. It has three main components: *Custom Operating System*, *mOS Bootstrapper* and *SPECS Cluster Manager*; these three components are described in the next paragraphs.

The *Custom Operating System* (cloud multi-purpose operating system) is a custom lightweight operating system based on the openSUSE 13.1 Linux Distribution [1] . mOS is a Custom OS (see D1.1.2) and it was built with the following goals: to create an operating system with a small footprint and short booting time, easily customizable to support any targeted cloud environment. As fare as customization is concerned, the most important issue is the flexibility of the kernel to support different virtualization technologies adopted by the cloud providers. This translates into compatibility with multiple cloud providers. The openSUSE Linux distribution is a mature Linux distribution that incorporates the most advanced operating technologies in terms of user experience, data security and kernel compatibility. But it has a big disadvantage: it is heavily preloaded with a lot of software packages unnecessary for running in a cloud environment. In order to maintain our goals, we built our own openSUSE distribution, a lightweight version that incorporates only the necessary software packages to make it working in a cloud environment. Furthermore, we added a set of special services on top of it, able to customize mOS based on the deployment requirements. The services are grouped under the name of *mOS Bootstrapper*.

The *mOS Bootstrapper* represents a collection of services that run on top of mOS in order to customize a mOS instance to fulfil the requirements, in terms of runtime services deployment, configuration and management, of the current running platform environment. It implements the two components designed in deliverable D1.1.2, namely the *Node Bootstrapper* and the *Component Logging*. In particular, these services are specialized in:

- automatic bootstraping of the software packages;
- resource management: resource registration and discovery of specific information using a custom distributed DNS-based service;
- automatic resource configuration based on a resource descriptor;
- offering a http interface to easily access various services information (logging, output data, configuration files and parameters);
- offering support for data migration using a specialized file system with built-in remote synchronization support.

The *Cluster Manager* is a specialized component in charge of the resource allocation (virtual machines) from a cloud provider and the deployment (installation, configuration and management) of specific components on the allocated resources. It has the following two subcomponents:

- *Cloud Resource Allocator*[7] – a REST web service responsible of cloud resource allocation from cloud providers;
- *Chef Deployment Solution*[8] – a collection of services in charge of software deployment;

The *Cluster Manager* consists of a set of components (see D1.1.2) used to deploy a distributed runtime, in an automatic way, using bootstrapping descriptors, where the entire SPECS Platform will be hosted.

The *SPECS Enabling Platform* is available and compatible with Amazon EC2 and Eucalyptus Cloud Software[3] .

The ***SPECS Testbed supporting infrastructure*** represents the execution environment for the *SPECS Enabling Platform* . The *SPECS Testbed supporting infrastructure* is offered in two variants: local (all-in-one limited solution) and hosted, with full support, that simulates a real cloud environment.

The *local solution* is based on Oracle VM VirtualBox [2] virtualization technology. It consists of a VirtualBox machine that the user is able to download and import on its own VirtualBox instance. With this solution the user is able to start a limited set of services to test, debug and develop the application that needs to be deployed on the top of the SPECS Platform. This variant of the SPECS Testbed supporting infrastructure will be described in the next release of the SPECS Testbed solution.

The *hosted solution* is based on Eucalyptus Cloud Software [3] . This hosted solution uses the Eucalyptus instance deployed at the servers of the IeAT partner. Eucalyptus Cloud Software delivers a private cloud environment that can be installed on a custom hardware infrastructure. It is compatible with Amazon EC2 (cloud compute instances) and Amazon S3 (object storage) in terms of API and part of the core functionalities. This feature will ensure that any developed solution compatible with Eucalyptus Cloud Software will be also compatible with Amazon EC2 (cloud compute instances) and Amazon S3 (object storage).

The Eucalyptus Cloud Software has a layered architecture consisting of five main components that enable the management of the cluster and of all cloud services. The components are briefly described as follows:

- Cloud Controller (CLC): the global orchestration service for the entire cloud software;
- Walrus: the bucket and object storage service similar to the Amazon S3 service;
- Cluster Controllers (CC): the local cluster orchestrator that manages the local clusters of resources;
- Storage Controllers (SC): the service in charge with volume storage management;
- Node Controllers (NC): the service in charge with the virtualization management; it runs on the local resources;

The IeAT instance of Eucalyptus deployment architecture is illustrated in Figure 2.

The Eucalyptus Cloud Software deployed uses version 4.1.0 of the cloud software and the components are deployed on IBM Blade Center hardware technology, as follows:

- *Service Node*: a set of collocated services: CLC, SC, CC and Walrus;
- *Cluster Node*: each NC blade runs its own NC service instance;
- *Storage Node*: available remotely from a dedicated storage server;

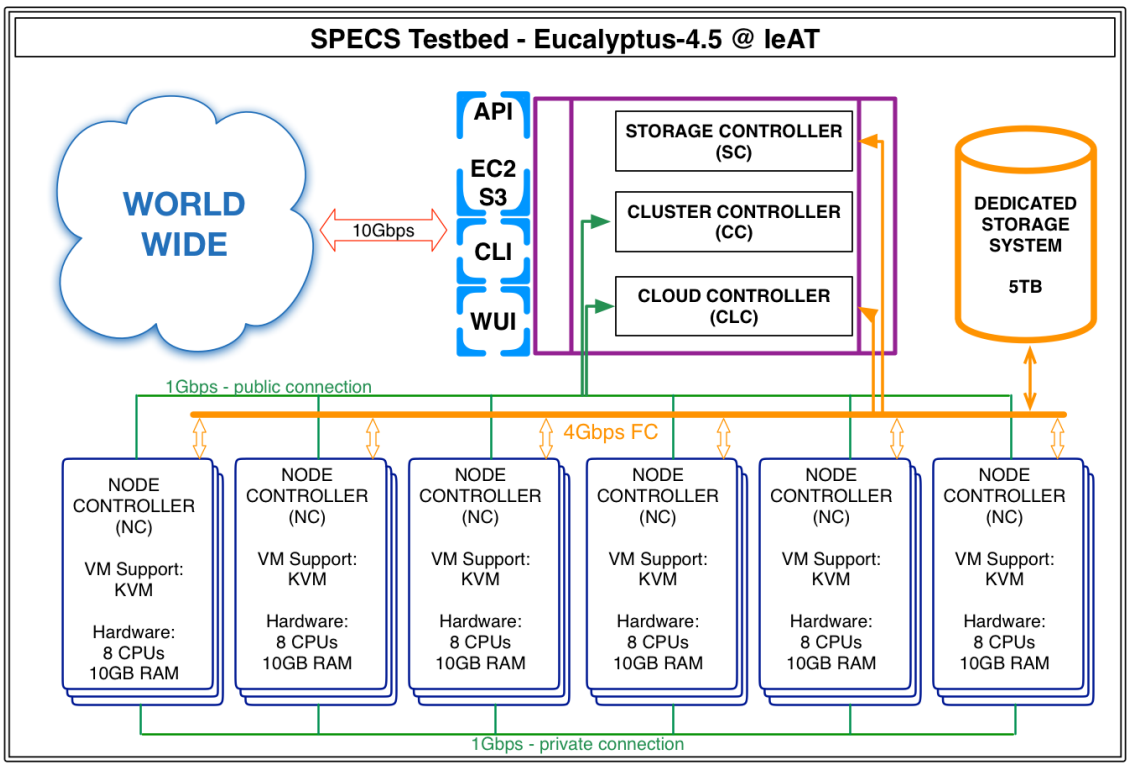

Figure 2.Eucalyptus architecture deployed at IeAT

The following table explains the type and the maximum number of instances supported by the IeAT instance of the Eucalyptus Cloud Software:

Instance type	CPU	RAM Memory (MB)	Disk storage (GB)	Maximum available
m1.small	1	256	5	36
m1.medium	1	512	5	36
c1.medium	1	512	10	18
m1.large	1	1024	10	18
c1.xlarge	1	2048	10	18
m2.2xlarge	2	2048	10	18
m1.xlarge	1	1024	15	12
m2.xlarge	1	2048	15	12
m3.xlarge	2	2048	15	12
m3.2xlarge	2	4096	30	6
cc1.4xlarge	4	4096	30	6

Table 1. IeAT Eucalyptus available resources

3.1. Status of development activities

In Table 2 we report the list of SPECS software components under development associated with the SPECS Enabling Platform, as discussed in D1.2 and D1.1.2, together with the requirements they respectively cover.

Enabling Platform Requirements	SPECS Software Components											
	Custom OS	***Components Logging***	***Node Bootstrapper***	***Node Logging***	***Node discovery***	***Node controller***	***Component Discover Sys.***	***Components Controller***	***Artifact Repository***	***Component Operational REST API***	***Node Operational REST API***	***Cluster Manager***
ENPL_R1				X		X				X		
ENPL_R2			X							X		X
ENPL_R3			X		X					X		X
ENPL_R4					X	X				X		
ENPL_R5					X	X				X		
ENPL_R6	X				X					X		
ENPL_R7	X									X		
ENPL_R8		X										
ENPL_R9								X			X	
ENPL_R10								X			X	
ENPL_R11								X			X	
ENPL_R12							X	X			X	
ENPL_R13							X	X			X	
ENPL_R14							X	X			X	
ENPL_R15		X									X	
ENPL_R16								X			X	
ENPL_R17								X			X	
ENPL_R18							X		X		X	
ENPL_R19									X			

Table 2. SPECS Components related to the Enabling Platform and related requirements

Almost all 19 requirements have been covered, even if the implementation of some components is still in progress. The 100% coverage will be provided in the second release of this report at the end of the project.

In Table 3, we report the actual development status of all SPECS artifacts associated with the Enabling Platform.

Module	Artefacts under development	Status
Enabling Platform	Components:Custom OS	Available
	Components:Components Logging	Available
	Components:NodeBootstrapper	Available
	Components:Node Logging	Available
	Components:Node discovery	Available
	Components:Node controller	Available
	Components:Component Discover Sys.	Work in progress

	Components:Components Controller	Work in progress
	Components:Artifact Repository	Work in progress
	Components:Component Operational REST API	Work in progress
	Components:Node Operational REST API	Work in progress
	Components:Cluster Manager	Work in progress

Table 3. Enabling Platform Implementation Status

We are currently working on the not-yet-finalized components and to a second testbed to be deployed in the EMC partner data center in order to prove the independence of the Enabling Platform from all services to be hosted.

3.2. Installation

The SPECS Testbed is offered in two variants: local version and hosted version. The local version is enabled by the VirtualBox technology that needs to be installed on the users' system; afterwards, the mOS exported virtual image must be imported into the VirtualBox system and started as a normal virtual machine. This variant of the SPECS Testbed will be deeply covered in the next version of the SPECS Testbed.
The hosted SPECS Testbed, using the Eucalyptus Cloud System, is installed at he IeAT partner data center and follows the official Eucalyptus installation procedure, available at [4] . Individual users will have to either setup a CLI (command line interface) environment, if they have chosen the CLI approach, or the web user interface, for better experience. The next subsection will cover the usage aspects of the hosted Eucalyptus instance at the IeAT partner.

3.3. Usage

In order to obtain access to the hosted **SPECS Enabling Platform**, a partner representative must send an email to support@specs-project.eu with the subject "[Eucalyptus@IeAT] request institution access" which contains:

- institution name;
- name, surname and email address for each user that needs Eucalyptus access.

Based on this resource request, an organisation account will be created, and for each user apart individual access accounts.

Each individual user will receive a confirmation email that contains the organization ID, username and password pair for web user interface access, and a link from which the user must download its credentials bundle (the usage of this bundle will be covered in this chapter).

The hosted SPECS Testbed can be used in two ways: using the CLI (command line interface) or the web user interface (WUI) available at [5] . Both entry points offer almost the same functionalities and access to the same resources from a cloud user perspective.

In order to setup a CLI environment a user must download and install Eucalyptus Euca2ools (the CLI commands) following the official procedure, documented in[6] . The CLI environment is suitable for Linux or Apple Mac users. After a successful installation, the credentials bundle must be copied into a custom directory and the file "*eucarc*" must be sourced before using the tools. More comprehensive tutorials on how to use the CLI environment, will be presented in the next release of the SPECS Testbed.

The web user interface does not require special requirements; the user must have a recent browser installed (recommended: Google Chrome, Mozilla Firefox) and he/she has to point at the WUI address, shown in[5] .

The usage of the hosted SPECS Testbed is split into two parts:

- the environment setup;
- virtual resources management.

Regarding the environment setup, after a successful login using the WUI [5] the user has access to Eucalyptus Dashboard (Figure 3). Before using the cloud resources, the user must define the remotepassword-lessauthentication credentials and the access security groups.

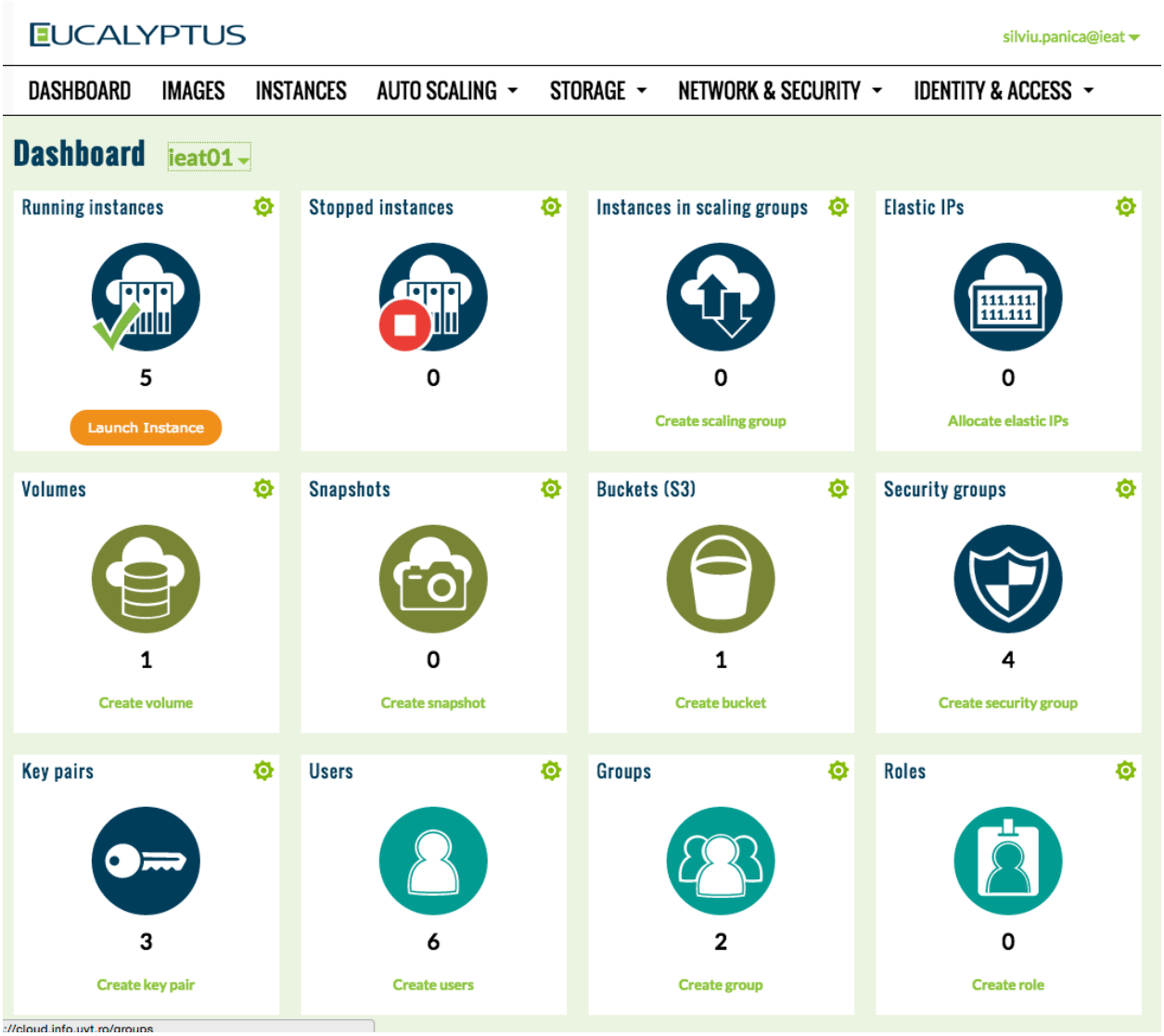

Figure 3. Eucalyptus web user interface – Dashboard

The remote password-less authentication credentials are a set of key pairs (private and public) used for password-less remote login on the cloud resources using the SSH protocol.

In order to create a key pair an user must:

1. access "*Network&Security*" tab
2. select "*Key pairs*"
3. click on "*Create New Key Pair*"

4. specify a custom key pair name
5. click "*Create and Download*"

At this point a private key will be downloaded from the server. This private key must be used by the SSH client, used to connect to the remote created resources, to login into the resource without the need of a username/password pair.

The next step is to create an access security group. The security groups are a set of access control rules for limiting the access to the cloud resources (like firewall rules for filtering the remote access). By default all cloud resources are filtered from outside. In order to create a security group the user must:

1. access "*Network&Security*" tab
2. select "*Security Groups*"
3. click on "*Create New Security Group*"
4. specify a custom "*Name*" and "*Description*" for the security group
5. select a predefined "*Protocol*" or "*Custom TCP/UDP*" (for defining you own range of access control)
6. specify a port range for which you want to grant access
7. select "*Allow traffic from*" and specify a remote range of IP addresses (or use 0.0.0.0/0 to allow any IP address)
8. click "*Add rule*" and repeat from step 5 for additional rules to add

The resource management represents the procedures forstarting up, shutting down or remotely access the virtual machines (instances).In order to start up an instance, the user must:

1. login the web user interface
2. select "*Instances*" tab
3. choose an image that wants to run and click "*Select*"
4. select "*Instance type*" and click "Next"
5. select already created "*Key name*" and "*Security group*"
6. click "*Launch instance*"

At this point the "*Instances*" tab will be populated with information regarding the newly created instance. Using the gear icon attached to the instance entry, the user can control various aspects of the started instance. To terminate the current instance, the user can use the "*Terminate*" command under the gear icon.

A running instance can be accessed using the SSH protocol by using a suitable SSH client (it is available by default under most Linux/Apple Mac environments; in the Windows environment, Putty can be used as SSH client). The default username for SSH login, is "*root*". The SSH private key generated when the environment was setup must be used for remote authentication (see above "The environment setup").

This tutorial is meant as an introduction and provides minimal instructions on how to access the hosted SPECS Testbed using the Eucalyptus Cloud Software. For more advanced features offered by the Eucalyptus, please refer to the official Eucalyptus Cloud Software user documentation [4] .

4. Conclusions

This document presents the first implementation of the SPECS Testbed to host all SPECS services during the project life.
In this document we summarized the status of implementation and integration activities, and reported the current coverage of the requirements that were located during the requirement analysis and design phases of the Enabling Platform.

Figure 4 illustrates the current status of the implementation activities.

Module	Component	M13 – M18	M19 – M30
Enabling Platform	Custom OS		
	Components Logging		
	Node Bootstrapper		
	Node Logging		
	Node discovery		
	Node controller		
	Component Discover Sys.		
	Components Controller		
	Artifact Repository		
	Component Operational REST API		
	Node Operational REST API		
	Cluster Manager		

Figure 4.Enabling Platform implementation plan

As illustrated, all components are under development, and half of them have been completed and reported in this document. The running platform is available, hosted by the IeAT partner and remotely accessible.

5. Bibliography

[1] OpenSUSE Operating System, https://www.opensuse.org/, as seen on 13.03.2015
[2] Virtualization technology: Oracle VM VirtualBox, https://www.virtualbox.org/, as seen on 20.04.2015
[3] Eucalyptus Cloud Software, http://www.eucalyptus.com/, as seen on 14.03.2015
[4] Eucalyptus Cloud Software installation procedure, https://www.eucalyptus.com/docs/eucalyptus/4.1.0/index.html, as seen on 20.05.2015
[5] Hosted SPECS Testbed web user interface, https://cloud.info.uvt.ro/
[6] Eucalyptus Euca2ools CLI tools, https://www.eucalyptus.com/download/euca2ools
[7] SPECS Cloud Resource Allocator, https://bitbucket.org/specs-team/specs-cloud-resource-allocator
[8] Chef Deployment solution, https://chef.io/