





# SeaClouds Project

# D6.4.3 Final evaluation report

Project Acronym	SeaClouds
Project Title	Seamless adaptive multi-cloud management of service-
	based applications
Call identifier	FP7-ICT-2012-10
Grant agreement no.	Collaborative Project
Start Date	1 <sup>st</sup> October 2013
Ending Date	31 <sup>st</sup> March 2016
Work Package	WP6 Case Studies implementation and validation
Due Date.	M29
Submission Date	Date of actual submission – 30 <sup>th</sup> November 2013
Version	1 0
Status	Final
Author(s):	Roi Sucasas (ATOS)

Román Sosa González (ATOS) Christian Tismer (NURO) Andrea Turli (Cloudsoft) Javier Cubo (UMA)

Reviewer(s)



Dissemination Level			
Project co-funded by the European Commission within the Seventh Framework Programme			
PU	PU Public X		
PP	Restricted to other programme participants (including the Commission)		
RE	Restricted to a group specified by the consortium (including the Commission)		
CO	Confidential, only for members of the consortium (including the Commission)		

# Version History

Version	Date	Comments, Changes, Status	Authors, contributors,
			reviewers



# **Table of Contents**

List of figu	res4		
List of tabl	les5		
Executive	Summary6		
1. Int	roduction7		
1.1	Purpose7		
1.2	Glossary of Acronyms8		
2. Eva	aluation Scenarios9		
2.1	Local environment9		
2.2	Cloud environment7		
3. No	n-functional testing17		
3.1	Documentation testing		
3.2	Installation testing		
3.3	Interoperability testing		
3.4	Usability testing		
4. Co	nclusions		
Annex 1: A	Annex 1: ATOS Case Study		
Annex 2: N	Annex 2: NURO Case Study		
Annex 3: T	Annex 3: Testing Tools/Software		
Annex 2: T	Annex 2: Test results report format Error! Bookmark not defined.		
Reference	References		



# .

# List of figures Figure 1 This is figure 1 ..... Error! Bookmark not defined.



### 5

# List of tables

Table 1. Acronyms	
Table 1 This is table 1	Error! Bookmark not defined.



### **Executive Summary**

The task 6.3 of the WP6 is responsible for making an assessment of the quality (effectiveness, efficiency and user satisfaction) of the SeaClouds platform solution, evaluating it from qualitative and quantitative points of view.

D6.4.3 SeaClouds periodic evaluation reports: Definition, in complete detail, of the protocol and procedures that should be followed during the evaluation of the performance of the Case Studies. Collection, categorization and statistical analysis of the feedback gained by the system engineers, through the evaluation protocol procedures, during the Case Studies execution and operation phase. In detail reporting of the experience during the operation of the various showcases. Step-by-step cookbook and methodological guidelines for the adoption of cloud computing platforms and services in real-life scenarios. Notice this deliverable is the last version of the D6.4.X.

Also, additional information about the different testbeds set-up to validate SeaClouds is added.

# 1. Introduction

The task 6.3 of the WP6 is responsible for the evaluation of the SeaClouds software platform. The Deliverable D6.4.2 includes the results of the first cycle of evaluation measurements and analysis of SeaClouds, evaluating both quantitative and qualitative requirements.

After the selection of tests methods and evaluation scenarios, defined in the deliverable D6.4.1 [1], it is now necessary to define a selection of tools and testbed set-ups to execute those tests and evaluate if the proposed SeaClouds solution fulfills the different requirements defined at the beginning of the project. This deliverable is strongly connected to WP2 to WP5:

- WP2 and WP6 define the technical requirements for SeaClouds. The technical requirements are based on a set of use cases, developed by the partners of the project. The WP2 also highlight the SeaClouds high-level architecture.
- WP3 and WP4 are the responsible of the low-level design and implementation of the SeaClouds design-time and run-time tools. Both components are going to be tested in the context of the WP6.
- The WP5 is responsible for the low-level design and implementation of the SeaClouds GUI.

The conclusions of this document will help to see the actual status of the project and what it is still missing to fulfil the requirements and objectives defined at the beginning of it.

This document is organized as follows:

Section 32 introduces the evaluation scenarios. A detailed description of all these tests, and the reason way they are done can be found in deliverable D6.4.1 **[1]**. The objective in this case is to see if SeaClouds fulfils all the functional requirements established at the beginning of the project in deliverable D2.1 **[2]** 

Section 3 presents the results of the non-functional evaluation. A detailed description of all these tests, and the reason way they are done can be found in deliverable D6.4.1 **[1]**. The objective in this case is to see if SeaClouds fulfils all the non-functional requirements established in D6.4.1.

#### 1.1 Purpose

The objective of this deliverable is to verify if the software solution as result of this project fulfils the functional requirements established at the beginning of it.

These functional requirements were defined in the deliverable D2.1 – Resubmission **[2]**. In D2.1 the SeaClouds consortium defined also a list of Use Cases that include a list of steps, which define interactions between actors and the SeaClouds platform as well the internal interactions performed by the SeaClouds platform to provide the overall functionalities.

In this deliverable two different evaluation scenarios (SeaClouds deployment configurations) will be defined to test the SeaClouds functionality: local installation, cloud installation.





# **1.2** Glossary of Acronyms

Acronym	Definition
UC1	Design an Application
UC2	Show Cloud offers
UC3	Produce Deployment Plans
UC4	Generate SLA Agreement
UC5	Deploy an Application (on a laas and on a PaaS)
UC6	Monitor an Application
UC7	Evaluate Management Policies
UC8	Re-plan Application Deployment
UC9	Migrate Application

Table 1. Acronyms

**Comment [RSG1]:** These aren't acronyms but the use cases for the functional evaluation.

#### 2. Evaluation Scenarios

All of the evaluation tests have been done in a local environment, to minimize network issues and to increase the repeatability of the tests. In the cloud environment we have verified the results obtained with fast-iterations on local environments.

The final platform is composed of 11 different services



Figure 1: SeaClouds Platform detail

#### 2.1 Local environment

According to the installation guide available at [8], the installation is done on a single box with the following hardware specifications: Ubuntu Trusty 64bit with 2048 MB RAM and 2 CPUs.

In that server, all of the 11 services that made up the platform can be deployed and used for testing load.

Use Case ID	LUC1
Use Case	Design an Application
Name	
Purpose	The purpose of this test is to design the topology and requirements of an
	application using the SeaClouds GUI.
Initiator	The Software Developer
Primary	The Software Developer
Actor	
Additional	
Actors	
Description	The test will cover the design of the topology of the NURO case study,
	which consists of (also a test for the Webchat application presented
	in Deliverable D5.1.3, although here we are going to focus on the
	explanation of the NURO case study, version 1 – Note in Section 3.4.1



	is presented also the usability testing of a complex version of the		
	NURO case study):		
	<ul> <li>PHP module. The technical requirements are:</li> </ul>		
	<ul> <li>Language: PHP &gt;= 5</li> </ul>		
	<ul> <li>To be deployed on PaaS</li> </ul>		
	<ul> <li>Container: Compressed file (zip/tarball)</li> </ul>		
	• Uses the database 3 times per call in average		
	Database. The technical requirements are:		
	MySQL >= 5		
	To be deployed on JaaS		
	Additionally, the following requirements have been defined:		
	Maximum Response Time: 2500 ms		
	Availability: 98.5%		
	<ul> <li>Budget per month: 400 €</li> </ul>		
	The expected workload of the application is 100 requests/second.		
Pre-	The SeaClouds platform is correctly installed.		
condition	The browser has the SeaClouds Dashboard loaded.		
Post-	The topology of the application described above is correctly defined.		
condition			
Use Case Fun	ctionality		
Sequence	1. Click the "New application" button		
•	2. Fill the application properties		
	a. Fill the application name: NURO		
	b Fill the optimization properties		
	i Response time: 2500		
	ii Availability: 98 5		
	iii Cost: 400		
	iv Workload: 100		
	2 Click port		
	4. Define the topology		
	4. Define the topology		
	a. Click web Application button and fill the properties		
	for the frontend module		
	I. Name: NURO PhpGame		
	II. Language: Python		
	III. Min version: 5		
	iv. Max version: 5		
	v. Code container: Compressed file (zip/tarball)		
	vi. Provider is: PaaS		
	vii. Location: Europe		
	b. Click Add. The module is added		
	c. Click Database button and fill the properties for the		
	database		
	i. Name: NURO MySQL		
	ii. Category: MySQL		
	iv. Max version: 5.6		



	v. Provider is: IaaS
	vi. Location: American
	d. Click Add. The module is added.
	e. Shift+Click on NURO MySQL and drag to NURO MySQL;
	fill the properties for the link
	i. Average number of calls: 3
	iii. Click Edit. The link is added. The result is shown
	in the following figure:
	NURO PhpGame
	±
	NUROMySQL
Alternative	

Use Case ID	LUC2
Use Case	Show Cloud offers
Name	
Purpose	The purpose of this test is to check that the cloud offerings provided
	by the planner match the technical requirements expressed in the
	topology.
Initiator	The Software Developer
Primary	The Software Developer
Actor	
Additional	
Actors	
Description	The test will cover the correctness of the plans generated by the
	planner for the application topology of the NURO case study. The
	generated plans should contain offerings matching the application
	requirements.



Pre-The SeaClouds platform is correctly installed. The browser has the SeaClouds Dashboard loaded. condition We have designed a valid application topology. Post-An Abstract Application Model (AAM) is generated, is specified in TOSCA and contains the technical requirements expressed in the condition topology. We finally have a set of cloud offers that matches the application requirements. For NURO PhpGame module, a PaaS offering Python >= 5 . **Use Case Functionality** Sequence 1. Create application topology as in LUC1. 2. Click next 3. Review generated Abstract Application Model 4. Review offerings provided by planner Alternative The AAM is generated. It contains the technical requirements Result expressed in the topology. The planner return a set of cloud offerings. The providers match the requirements.

Use Case ID	LUC3
Use Case Name	Produce Deployment Plans
Purpose	The purpose of this test is checking that a deployment plan expressed in TOSCA is generated for the plan selected by the user.
Initiator	The Software Developer
Primary Actor	The Software Developer
Additional Actors	

l1



Description	The test will cover the generation of a deployment plan following the TOSCA specification, which should declare that each module is going to be deployed in the selected offering, the SLA agreement and the monitoring rules.	
Pre- condition	The SeaClouds platform is correctly installed. The browser has the SeaClouds Dashboard loaded. The user has designed a valid application topology. The user has selected a plan. The user have entered the credentials of the cloud providers	
Post- condition	A Deployable Application Model (DAM) is generated, is specified in TOSCA and contains the cloud offerings selected by the user. The credentials for each provider are included in the DAM. The generated set of monitoring rules is included in the DAM. The generated SLA agreement is included in the DAM. The DAM without the SeaClouds extensions (SeaClouds policy) is deployable using Brooklyn.	
Use Case Functionality		
Sequence	<ol> <li>Select plan as in LUC2.</li> <li>Click next</li> <li>Enter provider credentials</li> <li>Click deploy</li> </ol>	
Alternative		
Result	The DAM is generated and includes the necessary information. The DAM without rules/SLA is deployable by Brooklyn.	

Use Case ID	11164
Use Case	Generate SLA Agreement
Name	
Purpose	The purpose of this test is checking that a WS-Agreement
	agreement is generated.
Initiator	The Software Developer
Primary Actor	The Software Developer
Additional	
Actors	
Description	This test will cover the correctness of the SLA agreement generated
	for the NURO case study.



Pre-condition	The SeaClouds platform is correctly installed. The browser has the SeaClouds Dashboard loaded. The user has designed a valid application topology. The user has selected a plan.
Post-condition	An agreement following WS-Agreement is generated. It contains a guarantee term to assess the desired availability of the application. It contains a guarantee term to assess the desired response time of the application.
Use Case Function	ality
Sequence	<ol> <li>Select plan as in LUC2.</li> <li>Click next</li> <li>Click check SLA agreement</li> </ol>
Alternative	
Result	When the application is deployed, we can check in the Dashboard the SLA agreement.

Use Case ID	LUC5
Use Case Name	Deploy an Application on IaaS/PaaS
Purpose	The purpose of this test is checking the correct deployment of the deployment plan.
Initiator	The Software Developer
Primary Actor	The Software Developer
Additional Actors	
Description	This test will cover the deployment of an application in IaaS and PaaS providers. The topology of the NURO case study defined the PHP module to be deployed on PaaS and MySQL module on IaaS.
Pre- condition	The SeaClouds platform is correctly installed. The browser has the SeaClouds Dashboard loaded.



	The user has designed a valid application topology. The user has selected a plan where providers are IaaS and PaaS according to the definition.	
Post- condition	Module NURO PhpGame is deployed. A MySQL service for the mysql module is created. The MySQL service has been bound to NURO PhpGame. The endpoint, database and credentials of the service have been	
	configured for NURO PhpGame.	
Use Case Functionality		
Sequence	<ol> <li>Select plan as in LUC2.</li> <li>Click next</li> <li>Enter provider credentials</li> <li>Click deploy</li> </ol>	
Alternative		
Result	The application is properly deployed.	

Use Case ID	LUC6
Use Case Name	Monitor an Application
Purpose	The purpose of this test is checking that SeaClouds is able to monitor a deployed application.
Initiator	The Software Developer
Primary Actor	The Software Developer
Additional Actors	
Description	This test will cover the monitoring of the application by Tower 4Clouds and the visualization of the monitoring metrics in the SeaClouds dashboard.
Pre-condition	A deployed application
Post- condition	The status view of the dashboard shows relevant metrics for the NURO case study.
Use Case Function	าลแซง



Sequence	<ol> <li>Deploy application as in LUC5</li> <li>Click Applications -&gt; NURO in top menu.</li> <li>Click Application Monitor in left menu.</li> <li>Select desired metrics</li> <li>Click Metric Viewer</li> </ol>
Alternative	<ol> <li>Deploy application using Brooklyn and a YAML deployment file.</li> <li>Proceed as in Step 2, 3, 4, 5 of the normal sequence</li> </ol>
Result	We are able to check the application is being monitored and visualize the monitored metrics.

Use Case ID	CEUC7	
Use Case Name	Evaluate Repairing Policies	
Purpose	The purpose of this test is checking the repairing policies in the SeaClouds platform.	
Initiator	The Software Developer	
Primary Actor	The Software Developer	
Additional Actors		
Description	This test will cover the application of repairing policies when the application is failing. The test will consist in deploying an application in a IaaS provider (e.g. Amazon) and PaaS provider (e.g. CloudFoundry), stopping manually one of the modules and checking that SeaClouds recognize the situation and is able to restart the component.	
Pre- condition	A deployed application using SeaClouds.	
Post- condition	SeaClouds recognize the application is failing. SeaClouds recover the application status.	
Use Case Func	Use Case Functionality	
Sequence	<ol> <li>Deploy the NURO application as in LUC5.</li> <li>Use the command line client of the cloud provider to stop the corresponding component.</li> </ol>	
Alternative	<ol> <li>Deploy the NURO application as in LUC5.</li> <li>Use the command line client of the cloud provider to delete the</li> </ol>	

	corresponding component.
Result	Adding an autoscaling policy, the application is recovered adding a new
	Instance

Use Case ID	LUC8
Use Case Name	Re-plan Application Deployment
Purpose	The purpose of this test is checking the replanification
	feature of the SeaClouds platform.
Initiator	The Software Developer
Primary Actor	The Software Developer
Additional	
Actors	
Description	
Pre-condition	
Post-condition	
Use Case Functionality	
Sequence	
Alternative	
Result	The feature is not implemented.

Use Case ID	LUC9
Use Case Name	Migrate Application
Purpose	The purpose of this test is checking the migration feature of the SeaClouds platform.
Initiator	The Software Developer
Primary Actor	The Software Developer
Additional Actors	
Description	

Pre-condition	
Post-condition	
Use Case Functionality	
Sequence	
Alternative	
Result	The feature is not implemented.

### 2.2 Cloud environment

The testbed has been prepared in the following Amazon EC2 VMs: Deployer, Dashboard: t2.medium

- 2 CPU (High Frequency Intel Xeon Processors with Turbo up to 3.3GHz)
- 4 GB RAM

Discoverer, Planner: t2.micro

- 1 CPU (High Frequency Intel Xeon Processors with Turbo up to 3.3GHz)
- 1 GB RAM

Monitor: m4.xlarge

- 4 CPUs (2.4 GHz Intel Xeon® E5-2676 v3)
- 16 GB RAM

SLA: m3.large

- 1 CPU (High Frequency Intel Xeon E5-2670 v2 (Ivy Bridge) Processors)
- 4 GB RAM

Use Case ID	CEUC1
Use Case	Design an Application
Name	
Purpose	The purpose of this test is to design the topology and requirements of
	an application using the SeaClouds GUI.
Initiator	The Software Developer
Primary	The Software Developer
Actor	
Additional	
Actors	
Description	The test will cover the design of the topology of the Atos case study,
	which consists of:
	Frontend module. The technical requirements are:
	<ul> <li>Language: Java &gt;= 7</li> </ul>

	<ul> <li>To be deployed on PaaS</li> </ul>
	<ul> <li>Container: Tomcat</li> </ul>
	<ul> <li>Uses the Web Services module 2 times per call in</li> </ul>
	average.
	<ul> <li>Web services module. The technical requirements are:</li> </ul>
	<ul> <li>Language: Java &gt;= 7</li> </ul>
	<ul> <li>To be deployed on PaaS</li> </ul>
	<ul> <li>Container: Tomcat</li> </ul>
	<ul> <li>Uses the database 2.5 times per call in average</li> </ul>
	Database. The technical requirements are:
	o MySQL >= 5
	<ul> <li>To be deployed on laaS</li> </ul>
	<ul> <li>VM: 4 CPUs, 50GB hard disk</li> </ul>
	Additionally, the following requirements have been defined:
	Maximum Response Time: 2000 ms
	Availability: 98%
	<ul> <li>Budget per month: 200 €</li> </ul>
	The expected workload of the application is 50 requests/second.
Pre-	The SeaClouds platform is correctly installed.
condition	The SeaClouds Dashboard is loaded in the browser.
Post-	The topology of the application described above is correctly defined.
condition	
Use Case Fun	ctionality
Sequence	1. Click the "New application" button
	2. Fill the application properties
	a. Fill the application name: Atos
	<ul><li>a. Fill the application name: Atos</li><li>b. Fill the optimization properties</li></ul>
	<ul><li>a. Fill the application name: Atos</li><li>b. Fill the optimization properties</li><li>i. Response time: 2000</li></ul>
	<ul> <li>a. Fill the application name: Atos</li> <li>b. Fill the optimization properties <ol> <li>Response time: 2000</li> <li>Availability: 98</li> </ol> </li> </ul>
	<ul> <li>a. Fill the application name: Atos</li> <li>b. Fill the optimization properties <ol> <li>Response time: 2000</li> <li>Availability: 98</li> <li>Cost: 200</li> </ol> </li> </ul>
	<ul> <li>a. Fill the application name: Atos</li> <li>b. Fill the optimization properties <ol> <li>Response time: 2000</li> <li>Availability: 98</li> <li>Cost: 200</li> <li>Workload: 50</li> </ol> </li> </ul>
	<ul> <li>a. Fill the application name: Atos</li> <li>b. Fill the optimization properties <ol> <li>Response time: 2000</li> <li>Availability: 98</li> <li>Cost: 200</li> <li>Workload: 50</li> </ol> </li> <li>3. Click next</li> </ul>
	<ul> <li>a. Fill the application name: Atos</li> <li>b. Fill the optimization properties <ol> <li>Response time: 2000</li> <li>Availability: 98</li> <li>Cost: 200</li> <li>Cost: 200</li> <li>Workload: 50</li> </ol> </li> <li>3. Click next</li> <li>4. Define the topology</li> </ul>
	<ul> <li>a. Fill the application name: Atos</li> <li>b. Fill the optimization properties <ol> <li>Response time: 2000</li> <li>Availability: 98</li> <li>Cost: 200</li> <li>Cost: 200</li> <li>Workload: 50</li> </ol> </li> <li>3. Click next <ol> <li>Define the topology</li> <li>Click Web Application button and fill the properties for</li> </ol> </li> </ul>
	<ul> <li>a. Fill the application name: Atos</li> <li>b. Fill the optimization properties <ol> <li>Response time: 2000</li> <li>Availability: 98</li> <li>Cost: 200</li> <li>Workload: 50</li> </ol> </li> <li>3. Click next <ol> <li>Define the topology</li> <li>Click Web Application button and fill the properties for the frontend module</li> </ol> </li> </ul>
	<ul> <li>a. Fill the application name: Atos</li> <li>b. Fill the optimization properties <ol> <li>Response time: 2000</li> <li>Availability: 98</li> <li>Cost: 200</li> <li>Workload: 50</li> </ol> </li> <li>3. Click next <ol> <li>Define the topology</li> <li>Click Web Application button and fill the properties for the frontend module <ol> <li>Name: www</li> </ol> </li> </ol></li></ul>
	<ul> <li>a. Fill the application name: Atos</li> <li>b. Fill the optimization properties <ol> <li>Response time: 2000</li> <li>Availability: 98</li> <li>Cost: 200</li> <li>Workload: 50</li> </ol> </li> <li>3. Click next <ol> <li>Define the topology</li> <li>Click Web Application button and fill the properties for the frontend module <ol> <li>Name: www</li> </ol> </li> </ol></li></ul>
	<ul> <li>a. Fill the application name: Atos</li> <li>b. Fill the optimization properties <ol> <li>Response time: 2000</li> <li>Availability: 98</li> <li>Cost: 200</li> <li>Workload: 50</li> </ol> </li> <li>3. Click next <ol> <li>Define the topology</li> <li>Click Web Application button and fill the properties for the frontend module <ol> <li>Name: www</li> <li>Language: Java</li> <li>Min version: 7</li> </ol> </li> </ol></li></ul>
	<ul> <li>a. Fill the application name: Atos</li> <li>b. Fill the optimization properties <ol> <li>Response time: 2000</li> <li>Availability: 98</li> <li>Cost: 200</li> <li>Workload: 50</li> </ol> </li> <li>3. Click next <ol> <li>Define the topology</li> <li>Click Web Application button and fill the properties for the frontend module <ol> <li>Name: www</li> <li>Language: Java</li> <li>Min version: 7</li> <li>Max version: 8</li> </ol> </li> </ol></li></ul>
	<ul> <li>a. Fill the application name: Atos</li> <li>b. Fill the optimization properties <ol> <li>Response time: 2000</li> <li>Availability: 98</li> <li>Cost: 200</li> <li>Workload: 50</li> </ol> </li> <li>3. Click next <ol> <li>Define the topology</li> <li>Click Web Application button and fill the properties for the frontend module <ol> <li>Name: www</li> <li>Language: Java</li> <li>Min version: 7</li> <li>Max version: 8</li> <li>Code container: Tomcat</li> </ol> </li> </ol></li></ul>
	<ul> <li>a. Fill the application name: Atos</li> <li>b. Fill the optimization properties <ol> <li>Response time: 2000</li> <li>Availability: 98</li> <li>Cost: 200</li> <li>Workload: 50</li> </ol> </li> <li>3. Click next <ol> <li>Define the topology</li> <li>Click Web Application button and fill the properties for the frontend module <ol> <li>Name: www</li> <li>Language: Java</li> <li>Min version: 7</li> <li>Max version: 8</li> <li>Code container: Tomcat</li> </ol> </li> </ol></li></ul>
	<ul> <li>a. Fill the application name: Atos</li> <li>b. Fill the optimization properties <ol> <li>Response time: 2000</li> <li>Availability: 98</li> <li>Cost: 200</li> <li>Workload: 50</li> </ol> </li> <li>3. Click next <ol> <li>Define the topology</li> <li>Click Web Application button and fill the properties for the frontend module <ol> <li>Name: www</li> <li>Language: Java</li> <li>Min version: 7</li> <li>Max version: 8</li> <li>Code container: Tomcat</li> <li>Provider is: PaaS</li> </ol> </li> </ol></li></ul>
	<ul> <li>a. Fill the application name: Atos</li> <li>b. Fill the optimization properties <ol> <li>Response time: 2000</li> <li>Availability: 98</li> <li>Cost: 200</li> <li>Workload: 50</li> </ol> </li> <li>3. Click next <ol> <li>Define the topology</li> <li>Click Web Application button and fill the properties for the frontend module <ol> <li>Name: www</li> <li>Language: Java</li> <li>Min version: 7</li> <li>Max version: 8</li> <li>Code container: Tomcat</li> <li>Provider is: PaaS</li> <li>Location: None</li> </ol> </li> </ol></li></ul>
	<ul> <li>a. Fill the application name: Atos</li> <li>b. Fill the optimization properties <ol> <li>Response time: 2000</li> <li>Availability: 98</li> <li>Cost: 200</li> <li>Workload: 50</li> </ol> </li> <li>3. Click next <ol> <li>Define the topology</li> <li>Click Web Application button and fill the properties for the frontend module <ol> <li>Name: www</li> <li>Language: Java</li> <li>Min version: 7</li> <li>Max version: 8</li> <li>Code container: Tomcat</li> <li>Provider is: PaaS</li> <li>Location: None</li> </ol> </li> </ol></li></ul>
	<ul> <li>a. Fill the application name: Atos</li> <li>b. Fill the optimization properties <ol> <li>Response time: 2000</li> <li>Availability: 98</li> <li>Cost: 200</li> <li>Workload: 50</li> </ol> </li> <li>3. Click next <ol> <li>Define the topology</li> <li>Click Web Application button and fill the properties for the frontend module <ol> <li>Name: www</li> <li>Language: Java</li> <li>Min version: 7</li> <li>Max version: 8</li> <li>Code container: Tomcat</li> <li>Provider is: PaaS</li> <li>Location: None</li> </ol> </li> <li>b. Click Web Application button and fill the properties for the web cenvices module</li> </ol></li></ul>



Use Case ID	CEUC2		
Use Case	Show Cloud offers		
Name			
Purpose	The purpose of this test is to check that the cloud offerings provided by the		
	planner match the technical requirements expressed in the topology.		
Initiator	The Software Developer		
Primary	The Software Developer		
Actor			
Addition			
al Actors			
Descripti	The test will cover the correctness of the plans generated by the planner		
on	for the application topology of the Atos case study. The generated plans		
	should contain offerings matching the application requirements.		
Pre-	The SeaClouds platform is correctly installed.		
conditio	The SeaClouds Dashboard is loaded in the browser.		
n	We have designed a valid application topology.		
Post-	An Abstract Application Model (AAM) is generated, is specified in TOSCA		
conditio	and contains the technical requirements expressed in the topology.		
n	We finally have a set of cloud offers that matches the application		
	requirements.		
	• For www module, a PaaS offering Java >= 7		
	• For ws module, a PaaS offering Java >= /		
	<ul> <li>For mysql module, a laas VM with CPUs &gt;= 4 and &gt;= 50GB hard disk</li> </ul>		
Use Case F	unctionality		
Sequenc	1. Create application topology as in CEUC1.		
е	2. Click next		
	3. Review generated Abstract Application Model		
	4. Review offerings provided by planner		
Alternati			
ve			
Result	The AAM is generated. It contains the technical requirements expressed in		
	the topology.		
	The planner returns a set of cloud offerings. The providers match the		
	requirements.		
	The result is shown below. The web applications are hosted in		
	CloudFoundry, and db in Amazon c3.xlarge or c3.2xlarge VMs (both satisfy		
	CPU and hard disk size constraints).		

2 location/s solution:		2 location/s solution:		2 location/s so
Cloud_Foundry - (Cost per hou 0.04) Properties ~ Hosted Services softcare-gui evaluation-ws	r (€):	Cloud_Foundry - (Cost per ho 0.04) Properties ~ Hosted Services softcare-gui evaluation-ws	our (€):	Cloud_Founda 0.04) Properties ~ Hosted Services softcare-gui evaluation-ws
Amazon_EC2_c3_xlarge_sa_ea (Cost per hour (€): 0.24) Properties ▼ Hosted Services evaluation-db	ist_1 -	Amazon_EC2_c3_xlarge_ap_ - (Cost per hour (€): 0.24) Properties ▼ Hosted Services evaluation-db	southeast_2	Amazon_EC2_ - (Cost per hou Properties ▼ Hosted Services evaluation-db

Use Case	CEUC3
ID	
Use Case	Produce Deployment Plans
Name	
Purpose	The purpose of this test is checking that a deployment plan expressed in
	TOSCA is generated for the plan selected by the user.
Initiator	The Software Developer
Primary	The Software Developer
Actor	
Addition	
al Actors	
Descripti	The test will cover the generation of a deployment plan following the
on	TOSCA specification, which should declare that each module is going to be
	deployed in the selected offering, the SLA agreement and the monitoring
	rules.
Pre-	The SeaClouds platform is correctly installed.
conditio	The SeaClouds Dashboard is loaded in the browser.
n	The user has designed a valid application topology.
	The user has selected a plan.
	The user have entered the credentials of the cloud providers
Post-	A Deployable Application Model (DAM) is generated, is specified in TOSCA
conditio	and contains the cloud offerings selected by the user.
n	The credentials for each provider are included in the DAM.
	The generated set of monitoring rules is included in the DAM.
	The generated SLA agreement is included in the DAM.
	The DAM without the SeaClouds extensions (SeaClouds policy) is
	deployable using Brooklyn.
Use Case F	unctionality
Sequenc	1. Select plan as in CEUC2.

е	2. Click next	
-	3 Enter provider credentials	
	4. Deste DANA (with sut Cas Claude a	
	4. Paste DAM (without Seaciouds p	folicy) in Brooklyn and deploy.
Alternati		
ve		
Result	The DAM is generated and includes the without rules/SLA is deployable by Broo below:	necessary information. The DAM klyn. The generated DAM is shown
	Deployable Application Model	TOSCA Deployable Application Model
	Amazon_EC2_ceviluation-db evaluation-db evaluation-db evaluation-db evaluation-db	<pre>tosca_definitions_version: tosca_simple_yd description: evaluation imports:</pre>

Use Case	CEUC4
Use Case	Generate SLA Agreement
Name	
Purpose	The purpose of this test is checking that a WS-Agreement agreement is generated.
Initiator	The Software Developer
Primary	The Software Developer
Actor	
Addition	
al Actors	
Descripti	This test will cover the correctness of the SLA agreement generated for the
on	ATOS case study.
Pre-	The SeaClouds platform is correctly installed.
conditio	The SeaClouds Dashboard is loaded in the browser.
n	The user has designed a valid application topology.
	The user has selected a plan.
Post-	An agreement following WS-Agreement is generated.
conditio	It contains a guarantee term to assess the desired availability of the
n	application.
	It contains a guarantee term to assess the desired response time of the

	a subsection of the second s
	application.
Use Case	Functionality
Sequenc	1. Select plan as in CEUC2.
е	2. Click next
	3. Check SLA agreement
Alternati	
ve	
Result	When the application is deployed, we can check in the Dashboard there is
	an SLA agreement with the two terms. See picture below:
	SLA Viewer View RAW Agreement
	Name: user-seaclouds-evaluation Service Provider: AgreementResponder Status: FUI FUI FD Expiration Time: Apr
	Guaranteed Terms
	respTimeSLARuleSoftcareGUI appAvailableSLARuleSoftcareGUI
	Status: FULFILLED Constrain: ResponseTimeViolation_SoftcareGUI NOT_EXISTS QoS: METRIC LE 200.0
	Violations
	Show ventries Search:
	ID Date Value
	No data available in table
	Showing 0 to 0 of 0 entries

Use Case	CEUC5
	Deuleu en Annliestian
Use Case	Deploy an Application
Name	
Purpose	The purpose of this test is checking the correct deployment of the
	deployment plan.
Initiator	The Software Developer
Primary	The Software Developer
Actor	
Addition	
al Actors	
Descripti	This test will cover the deployment of an application in PaaS and IaaS
on	providers. The topology specifies the two webapps to be deployed on PaaS
	and the database to be deployed on IaaS.
Pre-	The SeaClouds platform is correctly installed.
conditio	The SeaClouds Dashboard is loaded in the browser.
n	The user has designed a valid application topology.
Post-	Module www is deployed.
conditio	Module ws is deployed.
n	A VM and a MySQL database for the mysql module are created.
	The endpoint of ws has been configured for www

	The MySQL service has been bound to ws.	
	The endpoint, database and credentials of the service have been	
	configured for ws.	
	The agreement is stored in the SLA Service and its evaluation is started.	
	The monitoring rules are stored in the Monitor.	
Use Case F	unctionality	
Sequenc	1. Select plan as in CEUC2.	
е	2. Click next	
	3. Enter provider credentials	
	4. Click deploy	
Alternati		
ve		
Result	The application is properly deployed. The result is shown below.	
	seaclouds.app.iqe3bVGW Topology & Status Diagram Application Sensors	
	Topology & Status	
	Application Monitor	
	SLA	
	modacioudstre_evaluation-db3	
	Remove Application	
	Amazon_E02_c3_vlarge_eu_wes evaluation-ws3 evaluation-db3	
	seaclouds app(ge3bVGW	
	softcare-gui3	

	051400
Use Case	CEUC6
ID	
Use Case	Monitor an Application
Name	
Purpose	The purpose of this test is checking that SeaClouds is able to monitor a deployed application.
Initiator	The Software Developer
Primary	The Software Developer
Actor	
Addition	
al Actors	
Descripti	This test will cover the monitoring of the application by Tower 4Clouds and
on	the visualization of the monitoring metrics in the SeaClouds dashboard.
Pre-	A deployed application
conditio	
n	
Post-	The status view of the dashboard show relevant metrics for the Atos case

conditio	study.
n	
Use Case F	unctionality
Sequenc e	<ol> <li>Deploy application as in CEUC5</li> <li>Click Applications -&gt; Atos in top menu</li> <li>Click Application Monitor in left menu.</li> <li>Select desired metrics</li> <li>Click Metric Viewer</li> </ol>
Alternati ve	<ol> <li>Deploy application using Brooklyn and a YAML deployment file.</li> <li>Proceed as in Step 2, 3, 4, 5 of the normal sequence</li> </ol>
Result	We are able to check the application is being monitored and visualize the monitored metrics. A screenshot from the Grafana dashboard is shown below.
	AverageCpuUtilization_Amazon_EC2_c3_large_us_west_2
	Sign out         AverageRamUtilization_Amazon_EC2_c3_large_us_west_2           324 Mil         2016.04.07 11:57:57           - AverageRamUtilization_Amazon_EC2_c3_large_us_west_2 mean:         316.85 Mil           330 Mil         - AverageRamUtilization_Amazon_EC2_c3_large_us_west_2 mean:           316 Mil         - 11:56           11:54         11:58         12:00         12:02         12:04         12:08         11:08
	11:54 11:56 11:58 12:00 12:02 12:04 12:06 — AverageRamUtilization_Amazon_EC2_c3_large_us_west_2.mean

Use Case	CEUC7
ID	
Use Case	Evaluate Repairing Policies
Name	
Purpose	The purpose of this test is checking the repairing policies in the SeaClouds
	platform.
Initiator	The Software Developer
Primary	The Software Developer
Actor	
Addition	
al Actors	
Descripti	This test will cover the application of repairing policies in PaaS providers
on	when the application is failing. This test will cover the application of
	repairing policies when the application is failing. The test will consist in

	deploying an application in a PaaS provider (e.g. CloudFoundry), and manually stress the application (making a high number of requests per second) and checking that SeaClouds recognize the situation and is able to scale the affected component.			
Pre-	A deployed application using Sea(	Jouds		
anditio	A deployed application using seac			
conditio				
n				
Post-	SeaClouds recognize the abnorma	I situation.		
conditio	SeaClouds reacts and adds a new	instance.		
n				
Use Case	Functionality			
Sequenc	1 Deploy the Atos applicatio	n as in CELICS		
Sequenc	1. Deploy the Atos applicatio	load of the application		
e	2. Use jineter to increase the	ioau of the application.		
Alternati				
ve				
Result	The application is recovered adding a new instance to the frontend module. The code that adds the autoscaling policy to the DAM for PaaS modules is not integrated yet, so the policy had to be added manually.			
	The code that adds the autoscalin not integrated yet, so the policy h	ad to be added manually.	aas modules is	
	The code that adds the autoscalin not integrated yet, so the policy h The picture below shows an addee	ad to be added manually. d instance when the appl	ication is stressed.	
	The code that adds the autoscalin not integrated yet, so the policy h The picture below shows an adder Applications • C	ad to be added manually. d instance when the appl Summary Sensors Effectors	ication is stressed.	
	The code that adds the autoscalin not integrated yet, so the policy h The picture below shows an adder Applications • C seaclouds app.1/286wsSH	ad to be added manually. d instance when the appl Summary Sensors Effectors	ication is stressed.	
	The code that adds the autoscalin not integrated yet, so the policy h The picture below shows an adder Applications • C seaclouds.app.JZ86wsSH	ad to be added manually. d instance when the appl Summary Sensors Effectors	ication is stressed. Policies Activity Ad	
	The code that adds the autoscalin not integrated yet, so the policy h The picture below shows an adder Applications seaclouds.app.JZ86wsSH Amazon_EC2_c3_xlarge_us_east_1	ad to be added manually. d instance when the appl Summary Sensors Effectors	ication is stressed. Policies Activity Ad Value https://cf-app-YydjG7MB-do	
	The code that adds the autoscalin not integrated yet, so the policy h The picture below shows an adder Applications seaclouds.app.JZ86wsSH Amazon_EC2_c3_xlarge_us_east_1 evaluation-db	ad to be added manually. d instance when the appl Summary Sensors Effectors Name app.monitor.url app.resource.durationsum	ication is stressed. Policies Activity Ad Value https://cf-app-YydjG7MB-do 63	
	The code that adds the autoscalin not integrated yet, so the policy h The picture below shows an adder Applications • C seaclouds.app.JZ86wsSH • Amazon_EC2_c3_xlarge_us_east_1 • evaluation-db	ad to be added manually. d instance when the appl Summary Sensors Effectors Name app.monitor.url app.resource.durationsum app.resource.hits	ication is stressed. Policies Activity Ad Value https://cf-app-YydjG7MB-do 63 211	
	The code that adds the autoscalin not integrated yet, so the policy h The picture below shows an adder Applications • C seaclouds.app.JZ86wsSH Amazon_EC2_c3_xlarge_us_east_1 e evaluation-db modacloudsDc_evaluation-db	ad to be added manually. d instance when the appl Summary Sensors Effectors Name app.monitor.url app.resource.durationsum app.resource.hits app.resource.latency	ication is stressed. Policies Activity Ad Value https://cf-app-YydjG7MB-do 63 211 0.3	
	The code that adds the autoscalin not integrated yet, so the policy h The picture below shows an adder Applications • C seaclouds.app.JZ86wsSH Amazon_EC2_c3_xlarge_us_east_1 evaluation-db seacloudsDc_evaluation-db seacloudsDc_evaluation-db seacloudsDc_evaluation-db histances which are used to run the	ad to be added manually. d instance when the appl Summary Sensors Effectors Name app.monitor.url app.resource.hits app.resource.latency app.running.disk	ication is stressed. Policies Activity Ad Value https://cf-app-YydjG7MB-do 63 211 0.3 1024	
	The code that adds the autoscalin not integrated yet, so the policy h The picture below shows an adder Applications • C • seaclouds.app.JZ86wsSH • Amazon_EC2_c3_xlarge_us_east_1 • evaluation-db • seacloudsDc_evaluation-db • seacloudsDc_evaluation-db • seacloudsDc_evaluation-db • seacloudsDc_evaluation-db • evaluation-with are used to run the • application	g policy to the DAM for P ad to be added manually. d instance when the appl Summary Sensors Effectors Name app.monitor.url app.resource.durationsum app.resource.latency app.running.disk app.running.instances	ication is stressed. Policies Activity Ad Value https://cf-app-YydjG7MB-do 63 211 0.3 1024 2	
	The code that adds the autoscalin not integrated yet, so the policy h The picture below shows an adder Applications • C • seaclouds.app.JZ86wsSH • Amazon_EC2_c3_xlarge_us_east_1 • evaluation-db • seacloudsDc_evaluation-db • seacloudsDc_evaluation-db • seacloudsDc_evaluation-db • seacloudsDc_evaluation-db • seacloudsDc_evaluation-db • seacloudsDc_evaluation-db • seacloudsDc_evaluation-db	g policy to the DAM for P ad to be added manually. d instance when the appl Summary Sensors Effectors Name app.monitor.url app.resource.durationsum app.resource.durationsum app.resource.latency app.running.instances app.running.instances app.running.ram	aas modules is ication is stressed. Policies Activity Ad value https://cf-app-YydjG7MB-do 63 211 0.3 1024 2 512	
	The code that adds the autoscalin not integrated yet, so the policy h The picture below shows an adder Applications • C • seaclouds.app.JZ86wsSH • Amazon_EC2_c3_xlarge_us_east_1 • evaluation-db • modacloudsDc_evaluation-db • seacloudsDc_evaluation-db • seacloudsDc_evaluation-db	g policy to the DAM for P ad to be added manually. d instance when the appl Summary Sensors Effectors Name app.monitor.url app.resource.durationsum app.resource.latency app.running.instances app.running.instances app.running.ram app.server.latency	aas modules is ication is stressed. Policies Activity Ad Value https://cf-app-YydjG7MB-do 63 211 0.3 1024 2 512 15.6558	
	The code that adds the autoscalin not integrated yet, so the policy h The picture below shows an adder Applications • C • seaclouds.app.JZ86wsSH • Amazon_EC2_c3_xlarge_us_east_1 • evaluation-db • modacloudsDc_evaluation-db • seacloudsDc_evaluation-db • seacloudsDc_evaluation-db	g policy to the DAM for P ad to be added manually. d instance when the appl Summary Sensors Effectors Name app.monitor.url app.resource.durationsum app.resource.hits app.resource.latency app.running.disk app.running.instances app.running.ram app.server.latency app.server.latency app.server.latency	aas modules is ication is stressed. Policies Activity Ad Value https://cf-app-YydjG7MB-do 63 211 0.3 1024 2 512 15.6558 55253	
	The code that adds the autoscalin not integrated yet, so the policy h The picture below shows an adder Applications • C • seaclouds.app.JZ86wsSH • Amazon_EC2_c3_xlarge_us_east_1 • evaluation-db • modacloudsDc_evaluation-db • seacloud Researched in the application (Integer) • softcare-gui	g policy to the DAM for P ad to be added manually. d instance when the appl Summary Sensors Effectors Name app.monitor.url app.resource.durationsum app.resource.hits app.resource.latency app.running.instances app.running.instances app.running.ram app.server.latency app.server.latency app.server.requestpersecond	aas modules is ication is stressed. Policies Activity Ad Value https://cf-app-YydjG7MB-do 63 211 0.3 1024 2 512 15.6558 55253 7 2614	
	The code that adds the autoscalin not integrated yet, so the policy h The picture below shows an adder Applications • C • seaclouds.app.JZ86wsSH • Amazon_EC2_c3_xlarge_us_east_1 • evaluation-db • modacloudsDc_evaluation-db • seacloud-Breast which are used to run the • application • evaluation-ws (Integer) • softcare-gui	g policy to the DAM for P ad to be added manually. d instance when the appl Summary Sensors Effectors Name app.monitor.ul app.resource.durationsum app.resource.hits app.resource.hits app.resource.latency app.running.disk app.running.instances app.running.ram app.server.latency app.server.request app.server.requests app.server.requests	aas modules is ication is stressed. Policies Activity Ad value https://cf-app-YydjG7MB-do 63 211 0.3 1024 2 512 15.6558 55253 7 3531	
	The code that adds the autoscalin not integrated yet, so the policy h The picture below shows an adder Applications • C • seaclouds.app.JZ86wsSH • Amazon_EC2_c3_xlarge_us_east_1 • evaluation-db • modacloudsDc_evaluation-db • seacloud-Breastwhich are used to run the application • evaluation-wis (Integer) • softcare-gui	g policy to the DAM for P ad to be added manually. d instance when the appl Summary Sensors Effectors Name app.monitor.ul app.resource.durationsum app.resource.hits app.resource.hits app.resource.latency app.running.disk app.running.instances app.running.ram app.server.latency app.server.requests app.usedmemory app.usedmemory	aas modules is ication is stressed. Policies Activity Ad value https://cf-app-YydjG7MB-do 63 211 0.3 1024 2 512 15.6558 55253 7 3531 120911440 P	
	The code that adds the autoscalin not integrated yet, so the policy h The picture below shows an adder Applications • C • seaclouds.app.JZ86wsSH • Amazon_EC2_c3_xlarge_us_east_1 • evaluation-db • modacloudsDc_evaluation-db • seacloud Instances which are used to run the application (Integer) • softcare-gui	g polity to the DAM for P ad to be added manually. d instance when the appl Summary Sensors Effectors Name app.monitor.ul app.resource.durationsum app.resource.hits app.resource.hits app.running.disk app.running.disk app.running.ram app.server.latency app.server.requests app.server.requests app.usedmemory cloudFoundry.webapp.boundServices reain.ui	Aas modules is ication is stressed. Policies Activity Ad Value https://cf-app-YydjG7MB-do 63 211 0.3 1024 2 512 15.6558 55253 7 3531 120911440 D bttps://cf app YudiC 7440 d to	
	The code that adds the autoscalin not integrated yet, so the policy h The picture below shows an adder Applications • C seaclouds.app.JZ86wsSH • Amazon_EC2_c3_xlarge_us_east_1 • evaluation-db • modacloudsDc_evaluation-db • seacloud Instances which are used to run the application (Integer) • softcare-gui	g polity to the DAM for P ad to be added manually. d instance when the appl Summary Sensors Effectors Name app.monitor.ul app.resource.durationsum app.resource.hits app.resource.hits app.running.disk app.running.instances app.running.ram app.server.latency app.server.requests app.server.requests app.usedmemory cloudFoundry.webapp.boundServices main.uri service is In	AaS modules is ication is stressed. Policies Activity Ad Value https://cf-app-YydjG7MB-do 63 211 0.3 1024 2 512 15.6558 55253 7 3531 120911440 I https://cf-app-YydjG7MB-do	
	The code that adds the autoscalin not integrated yet, so the policy h The picture below shows an adder Applications • C seaclouds.app.JZ86wsSH • Amazon_EC2_c3_xlarge_us_east_1 • evaluation-db • modacloudsDc_evaluation-db • seacloud fistances which are used to run the • application (Integer) • softcare-gui	g polity to the DAM for P ad to be added manually. d instance when the appl Summary Sensors Effectors Name app.monitor.ul app.resource.durationsum app.resource.hits app.resource.hits app.resource.hits app.resource.latency app.running.instances app.running.instances app.running.ram app.server.latency app.server.requests app.usedmemory cloudFoundry.webapp.boundServices main.uri service.isUp service.netUb indicators	AaS modules is ication is stressed. Policies Activity Ad Value https://cf-app-YydjG7MB-do 63 211 0.3 1024 2 512 15.6558 55253 7 3531 120911440 I https://cf-app-YydjG7MB-do true 0	
	The code that adds the autoscalin not integrated yet, so the policy h The picture below shows an added Applications • C seaclouds.app.JZ86wsSH • Amazon_EC2_c3_xlarge_us_east_1 • evaluation-db • modacloudsDc_evaluation-db • seacloud-fistances which are used to run the • application (Integer) • softcare-gui	g polity to the DAM for P ad to be added manually. d instance when the appl Summary Sensors Effectors Name app.monitor.ul app.resource.durationsum app.resource.hits app.resource.hits app.resource.hits app.resource.latency app.resource.latency app.running.instances app.running.instances app.running.ram app.server.latency app.server.requests app.usedmemory cloudFoundry.webapp.boundServices main.uri service.isUp service.notUp.indicators service.notUp.indicators	Aas modules is ication is stressed. Policies Activity Ad Value https://cf-app-YydjG7MB-do 63 211 0.3 1024 2 512 15.6558 55253 7 3531 120911440 I https://cf-app-YydjG7MB-do true Q true	

Use Case ID	CEUC8
Use Case Name	Re-plan Application Deployment
Purpose	The purpose of this test is checking the replanification feature of the
	SeaClouds platform.
Initiator	The Software Developer
<b>Primary Actor</b>	The Software Developer

Additional	
Actors	
Description	
Pre-condition	
Post-condition	
Use Case Functio	nality
Sequence	
Alternative	
Result	The feature is not implemented.

Use Case ID	CEUC9
Use Case Name	Migrate Application
Purpose	The purpose of this test is checking the migration feature of the
	SeaClouds platform.
Initiator	The Software Developer
Primary Actor	The Software Developer
Additional	
Actors	
Description	
Pre-condition	
Post-condition	
Use Case Function	nality
Sequence	
Alternative	
Result	The feature is not implemented.

# 3. Non-functional testing

A widely accepted definition for non-functional requirements is the following: "In systems engineering and **requirements** engineering, a **non-functional requirement** is a **requirement** that specifies criteria that can be used to judge the operation of a system, rather than specific behaviors. They are contrasted with **functional requirements** that define specific behavior or functions." [Wikipedia]

In the SeaClouds project those non-functional requirements were defined in chapter 5 of D6.4.1. However, at project months 30 (PM30) the SeaClouds platform is not mature enough to be fully evaluated according to the set of requirements listed in D6.4.1.

The following table summarizes the results of the measurable metrices at project months 30.

# Non-Functional Requirements tests at M30

Performance/Scalability Testing
Stress Testing
Documentation Testing
Local and in Cloud Installation Testing
Regression Testing
Long Term Testing
Final Interoperability testing
Early Usability Testing

Some of the predefined testing strategies couldn't be performed as the SeaClouds platform has to be considered a PoC and not a production-ready system. Most of the above criteria require a stable platform, fully documented and with a good testing coverage since the beginning of the project, but at least, three months before the end of the project: integration tests, stress tests, performance tests are meaningful only starting from a solid testable codebase, otherwise the outcomes of that testing strategies may be very misleading.

### 3.1 Documentation testing

Documentation testing means verifying that the SeaClouds documentation user manuals, including guidelines, tutorials and on-line documentation- are easy to read and understand, grammatically correct and technically accurate.

Test ID	Docu (DT)	Jme	ntation	Testing	Date	11/04/2016
Tester	Andr	ea 1	Turli (Clou	dsoft)	Testbed name	Local Testbed and Cloud Testbed.
SeaCloud Platform Version	eaClouds 1.0.0 latform /ersion					
SeaClouds REA documentatio n Version ME		REA <u>https</u> <u>ME.</u> 1	DME.md fro :://github.co md	om SeaClo m/SeaClou	udsPlatform v.1. dsEU/SeaCloudsP	0.0 latform/blob/master/READ
	Test Results					
Involved SeaClouds Dashboa Components SeaClouds Discovere		rd, SeaClouds r, SeaClouds Plar	SLA, SeaClouds Monitor, nner, SeaClouds Deployer			
Interaction Between Components			Not tested here.			

Passed?	Yes	Bug ID	na
Problems: general observations			
Required Changes: specific changes to be made			
Cost Estimation			
Comments			

# 3.2 Installation testing

Installation testing verifies the correct work of the installation procedure of SeaClouds in different configuration environments.

# 3.2.1 Local installation testing

Test ID	Local Ir	stallation Testing (LIT)	Date	11/04/2016		
Tester	Andrea Turli (Cloudsoft)		Testbed name	Local environment/Ubuntu		
				12.04.1		
SeaCloud	ls	1.0.0				
Platform						
Version						
Test Res	ults					
Involved		Any SeaClouds compor	ents deployed / lau	nched using Apache Brooklyn. V	Ne	
Compon	ents	currently support deplo	oyments against Brir	ng Your Own Nodes (BYON) and	to	
		all the IaaS provider su	pported by Apache	jclouds.		
Environn	nent	Following				
characte	ristics	<u>https://github.com/</u>	<u>/SeaCloudsEU/Se</u>	aCloudsPlatform/blob/mc	<u>1st</u>	
		er/README.md#de	ploy-seaclouds t	he installation is done on a sing	le	
		box with the following	spec:			
		box: ubuntu/trusty64				
ram: 2048		ram: 2048				
cpus: 2						
Pre-		virtualbox				
requirem	nents	https://www.virtuall	<u>oox.org/wiki/Dov</u>	<u>vnloads</u>		
		vagrant				
		https://docs.vagrantup.com/v2/installation/				
			• · · · · · · · · · · · · · · · · · · ·			
		The Apache Brooklyn 0.9.	U server setup and th	le subsequent bootstrap is entirely	'	
		SeaCloudsELL/SeaCloudsE	Platform nublic reposi	tory on github com		

Installation	Point your favourit	e browser at	http://10.10.10.100:8081
steps	Select SeaClouds Pl	atform on BYC	N application from Apache Brooklyn
	dropdown menu		
	Click on Finish but	ton	
Passed?	Yes		
Problems			
Required			
Changes			
<b>Cost Estimation</b>			
Comments			

# 3.2.2 Cloud installation testing

Test ID	Ins	tallation Testing (IT)	Date	11/04/2016
Tester	An	drea Turli (Cloudsoft)	Testbed name	Cloud Testbed.
SeaClouds		1.0.0		
Platform				
Version				
Test Result	s			
Involved		SeaClouds Dashboard,	SeaClouds SLA, Sea	aClouds Monitorr SeaClouds,
Componen	ts	SeaClouds Discoverer,	SeaClouds Planner	, SeaClouds Deployer
Environme	nt	Cloud Testbed: AWS-E	C2 m3.medium ins	tances (one for each of the
characteris	tic	involved components).		
S				
Interaction				
Between		Not tested here.		
Componen	ts			
Installation		The steps followed are	those reported in	the README available at
steps		https://github.com/Sea	aCloudsEU/SeaClo	udsPlatform/blob/master/REA
		DME.md		
Passed?		YES		
Problems		NONE		
Required				
Changes				
Cost				
Estimation				
Comments		All the components we	re successfully inst	talled and were reachable
		installing the platform	on AWS-EC2.	

# 3.3 Interoperability testing

Interoperability is about testing whether a software is compatible with others and promotes interoperable functionalities. SeaClouds was designed as standard-based platform just to ensure interoperability, adoption and re-usability. In this context the role of standards is primary and then we should consider if:

• SeaClouds (partially) uses standardized open initiative like OASIS TOSCA and OASIS CAMP.

• Some of the SeaClouds members are contributing to standardize OASIS CAMP and OASIS TOSCA.

Internally SeaClouds consortium agreed on implementing the software platform using a microservices architecture. A well-known definition of Microservice has been given by A. Cockcroft "Loosely coupled service oriented architecture with bounded context" This architectural choice has implications also on the interoperability testing, as it should simplify the testing of the internal components which expose well-defined API limiting their boundaries. Most of the SeaClouds component tend to be TOSCA(-ish) compliant: in particular, a lot of effort has been invested on the SeaClouds Deployer to enable TOSCA compliance.

OASIS CAMP input	Yes, users can directly run manually prepared OASIS
	CAMP plans via the Dashboard
OASIS TOSCA input	Yes, users can directly run manually prepared OASIS
	TOSCA plans via the Dashboard
OASIS CAMP output	When using the drag and drop UI the user cannot export
	OASIS CAMP plans.
OASIS TOSCA output	When using the drag and drop UI the user can export
	OASIS TOSCA plans.

#### 3.3.1 Dashboard interoperability testing

#### 3.3.2 Planner interoperability testing

OASIS CAMP input	No, users cannot directly input manually prepared OASIS CAMP plans.
OASIS TOSCA input	Yes, users can directly run manually prepared OASIS TOSCA plans.
OASIS CAMP output	No, the planner cannot produce OASIS CAMP plan.
OASIS TOSCA output	Yes, the planner can produce OASIS CAMP plan. Anyway those plans are not complete as they miss the credentials to access the cloud provider. Those needs to be filled from the SeaClouds Dashboard, tightly-coupling the 2 components, and reducing independent re- usability of the components in standalone mode.

#### 3.3.3 Discoverer interoperability testing

OASIS CAMP input	N/A – discoverer passively responds to external
	requests using REST API. The query language to discover
	offerings is not following any particular standard.
OASIS TOSCA input	N/A – as above
OASIS CAMP output	No, Discoverer can't produces cloud offerings formatted
	as OASIS CAMP types.
OASIS TOSCA output	Discoverer produces cloud offerings formatted as OASIS
	TOSCA NodeTemplates.

3.3.4 Deployer interope	erability testing
OASIS CAMP input	Yes – Deployer accepts OASIS CAMP plan as application
	model to be deployed.
OASIS TOSCA input	Yes – Deployer accepts OASIS TOSCA plan as application
	model to be deployed.
OASIS CAMP output	N/A –deployer output is an application deployment not
	a description of its topology.
OASIS TOSCA output	N/A –same as above.

### 3.3.5 Monitor interoperability testing

OASIS CAMP input	No - Monitoring rules are expressed in their own format
	and SeaClouds doesn't provide any transformer to re-
	map CAMP policies into Tower4Clouds monitoring rules.
OASIS TOSCA input	No - Monitoring rules are expressed in their own
	format. SeaClouds provides a series of transformers to
	re-map TOSCA policies into Tower4Clouds monitoring
	rules.
OASIS CAMP output	No – Monitoring system doesn't produce metrics CAMP-
	compliant.
OASIS TOSCA output	No – Monitoring system doesn't produce metrics TOSCA-
	compliant.

# 3.3.6 SLA interoperability testing

OASIS TOSCA input	The SLA Service needs a TOSCA file (the DAM) to
	generate the templates and agreements of the
	application
WS-Agreement output	The SLA Service generates templates and agreements
	following the WS-Agreement specification.

#### 3.4 **Usability testing**

Through time many definitions for usability have been proposed. Two of the most established definitions can be found in international standard for the evaluation of software ISO 9241-11 [1] and ISO 9126 [4].

The Guidance on usability in ISO 9241-11 outlines the usability as "the level to which a (software) product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use".

On the other hand, in the standard ISO 9126, usability is defined as "the capability of a software product to be understood, learned, used and attractive for the user, when it is used under specified conditions". In depth, usability studies relate to evaluating a product by testing it on representative users while they focus not only on how well users can learn and use a product to achieve their goals but also on how satisfied users are with that process. This can be seen as an irreplaceable usability practice since it gives direct input on how real users use the system. Usability studies examine three principles: effectiveness, efficiency and overall satisfaction of the user. [6]

In the context of the SeaClouds project usability testing is a perceptual test depending of the tester. No tasks are done in an automatic way. It is important to note that the

testers could use one of the testbeds to perform the usability tests or to install the SeaClouds platform in a virtual machine (similar to the installation test). Due to the subjectivity of this test, it is going to be performance by at least two different partners of the project.

# 3.4.1 NURO usability testing

As announced in D6.3.3 in this section NURO describes a complete design and deploy session of a medium complex NURO use case scenario with three PHP Components and one database.



# Further details on the NURO CaseStudy can be found in the D6.3.x deliverable family and D6.1.

Test ID	Usability	Date	M30
	Testing 1 (UT1)		
Tester	Christian	Testbed name	Cloud Testbed
	Tismer (Nuro)		http://52.49.41.119:8000/
SeaClouds Plat	form Version		Final
Test Results			
Involved	All SeaClouds Pla	atform Componen	ts, NURO CaseStudy Components
Components			
Interaction	The integration	between the Com	ponents stabilized at testing time and an
Between	end to end test	with different com	plex topologies was made.
Components	SeaClouds Dash	board	
	SeaClouds Dashboard - A	eactouds Seamle	
	← → C ↑ ☐ 52.49.41.119:800	0/#/applications	☆ Ø 6 ≡
	SeaClouds Dashboard A	pplications - Grafana Dashboard	API About
	Œ	seaclouds.ap status B	pi.q5WJJtX
	Liguro 2. SooClo	ude Dashbaard	research project EC-FP7-ICT-610531 Seaclouds.
	Figure 2: Seacio	uas Dasnooara - S	
	On start the Das	hboard is interact	ing with the deployer and monitor.
	Application dep	loyments and state	us is shown.
	A click on the pl	us starts the Appli	cation Design Wizard. It is intuitively and
	fun to use, mod	ules and paramete	ers are suitable for this research level

![](_page_34_Figure_1.jpeg)

Passed?	Yes	Bug ID		
Problems	NONE			
Required	NONE			
Changes				
Cost	NONE			
Estimation				
Comments	The final SeaCle	ouds evaluatio	n and testing succeeded on the major propo	sed
	features of the	Project. In adv	vance to the previous evaluation, the integrat	tion
	of components	was sufficient	and human interaction was not necessary w	hile
	the evaluation.	On extended	testing sessions sometimes human intervent	ion
	was needed. So	ome cloud offe	rings did not support all promised features.	

#### Alternative topologies tested and additional evaluation:

Please	define the topology of NURO CaseStudy Simple
	NURO GUI

Figure 6 NURO CaseStudy - Simple, standalone GUI

The SeaClouds Project outcome was tested with different complex topologies. The possible simplest deployment is a single module deployment of the NURO Case Study's GUI. It is designed to run also standalone.

![](_page_35_Picture_6.jpeg)

Figure 7: NURO CaseStudy - Previous Topology

The path to a NURO CaseStudy API deployment can be given by the web frontend and the Benchmarking tool can also be used to verify erformance of complete independent web services.

Beside is the topology of the previous evaluation report.

NURO GUI and API are bundled with one deployment package and deployed together as one module.

The administrative tools module was not available at that time.

If the deployment topology and maturity of the project advanced, NURO was

convinced by the flexibility of mixed IaaS and PaaS deployments the SeaClouds system offered.

seaclouds.app.I12v9nQh	Topology & Status Diagram	Application Sensors
Topology & Status		
Application Monitor		
SLA		seaclfudsDc_db
Remove Application		Amazon_EC2_c3_large_us_east_1
		ab seacloudd aco, 1999nQh
		nuro-pma nuro-gui

The following image series documents the design and deployment of this D6.4.3 evaluation deployment on the cloud testbed of SeaClouds.

SeaClouds Dashboard - A <sub>1</sub> ×	seaccouds	Seamless ads	ptive multi-cloud man	
→ C f 52.49.41.119	9:8000/#/applications			😭 🕺 🖉
SeaClouds Dashboard	Applications - Grafana [	Dashboard		API Abo
E	Đ			
re 9: NURO CaseStu	This work is part o udy - Evaluation Star	f the ongoing European resear	ch project EC-FP7-ICT-610531 SeaClouds	5.
SeaClouds Dashboard	Applications - Grafana Dash	board		API A
0	2	3	4	5
Application properties	Design topology	Optimize & Plan	Configuration summary	Process Summary & Deploy
Application Name	Nuro-Complex			
Fill these proper	ties if you want to o	ptimize your appli	ication	
Desired	application availability (%)	99		
Maximum	n estimated cost (€/month):	500		

![](_page_36_Figure_5.jpeg)

Figure 8: NURO CaseStudy - D6.4.3 Topology

27

SeaClouds Dashboard Applicatio	ns - Grafana Dashboard	API A
	Database	
1 Application properties	Description V	5 Process Summary & Deploy
	Name	
	Module name	
Please define the topology of	N This is the frontend component	
	No \$	Available Modules
	Technological Requirements 🗸	Database
	Non-functional Requirements 🗸	Databage
	Provider Infrastructure 🗸	Web application
	Close Add	

Figure 11: NURO CaseStudy - Evaluation Step 2.1 Add a Database

db		br00k11n				
This is the frontend component		Non-functional Require	ements 🗸			
No	\$	Benchmarked response	time (ms)			
Technological Requirements 🗸		14				
Category		Benchmark platform				
Category		Amazon AWS medium in	nstance			÷
MySql	•	QoS				
Min version		Metric name	Operator	Thrashold	Actions	
5	•		operator	meanoid	Acaona	
Max version		Availability \$	> \$	99	\$	×
5.6	\$					+
Initialization script						
https://www.dropbox.com/s/rp7owcgyhaa692f/create.sql?di=1		Autoscale component				
Database		No				\$
database1		Provider Infrastructure	<b>~</b>			
User		Provider is Any	aS Paat	2		
brooklyn				~		
Password		Num CPUs				
br00k11n		2				
		Min Disk Size (GB)				
Non-functional Requirements 🗸		15				
Benchmarked response time (ms)		Location None S	tatic 🕞 Dyr	amic		
14						
Benchmark platform						_
Amazon AWS medium Instance	0				Close	Add

Figure 12: NURO CaseStudy - Evaluation Step 2.2 Configure Database

Neb application	This is the frontend com	ponent			
Zescription ✓	No				
lame					¢
	Technological Requirem	nents 🗸			
nuro-api	Non-functional Require	ments 🗸			
his is the frontend component	Renchmarked recourse	time (me)			
No	4	unio (ms)			
echnological Requirements 🗸	Benchmark platform				
Language	Amazon AWS medium in	stance			¢
Php \$	QoS				
din version	Metric name	Operator	Threshold	Actions	
5.1 \$	Availability \$	> \$	99	•	×
fax version					
5.5 \$					+
ode container	Autoscale component				
Apache \$	No				¢
utifact	Provider Infrastructure	~			
west-2.amazonaws.com/seaclouds-usescases-artifacts/php/paas/splitted/nuro-api.zip					
Ion-functional Requirements V	Provider is Any Ia	aS 💿 Paas	6		
Provider Infrastructure	Location None St	atic 💮 Dyn	amic		

Figure 13: NURO CaseStudy - Evaluation Step 3-5 PHP Modules Configuration

SeaClouds Dashboard	Applications - Grafana Dashbo	ard		API About
1 Application properties	2 Design topology	3 Optimize & Plan	4 Configuration summary	5 Process Summary & Deploy
Please define the top	bology of Nuro-Complex			Available Modules
				Database Web application
		db nuro-pma		
	nuro-gui	nuro-api		•
Back Next				

Figure 14: NURO CaseStudy - Evaluation Step 6 Modules added

Please define the topology of Nuro-	Complex	Available Modules
		Database
	do Received and Antonia	Web applicatio
	nuro-gui nuro-api	
gure 15: NURO CaseStudy - Evaluat	tion Step 7-9 Connecting	
Link	×	
Description		
Operation type		
PHP-db connection	\$	
Env var to store endpoint URL		
db_connection_uri		
Average number of calls		
1		
	Close	
Link	×	
Description		
Operation type		
	\$	
HITP connection		
Env var to store endpoint URL		
Env var to store endpoint URL		
Env var to store endpoint URL nuro_api_uri Average number of calls		
In P connection  Env var to store endpoint URL      nuro_api_uri  Average number of calls  1		
HIP connection       Env var to store endpoint URL       nuro_api_ur(       Average number of calls       1		

Figure 16: NURO CaseStudy - Evaluation Step 7-9 Connection Configuration

Application properties Please define the topolo	Design topology	Optimize & Plan	4 Configuration summ	nary Process Summar	ry & Deploy
Please define the topolo	ogy of Nuro-Comple				
	nur	PA	nuro-gui	Available D Web	a Modules
x Next 17: NURO CaseStud SeaClouds Dashboard Ap 1 Application properties	ly - Evaluation Ste plications - Grafana Das 2 Design topology	ep 10 Design Ready hboard Optimize & Pian	4 Configuration summa	5 ry Process Summary &	API i Deploy
Please choose one of t	the proposed setups	by SeaClouds	I	Previous configurations     Next configuration	rations >
2 location/s solution: Cloud_Foundry - (Cost p 0.04) Properties ~ Hosted Services	per hour (6):	2 location/s solution: nuro-pma Amazon_EC2_m3_large - (Cost per hour (€): 0.18 Properties +	ap_northeast_1	2 location/s solution: Hosted Services nuro-apl nuro-gui nuro-pma	

![](_page_40_Figure_2.jpeg)

31

![](_page_41_Figure_1.jpeg)

Application properties	2 Design topology	Optimize & Plan	4 Configuration summary	Process Summary & Deploy
				, , , , , , , , , , , , , , , , , , , ,
Deployment log	A	Application status		
Starting the deployment p	process Done.			
The application deploymer	nt process was tri-			
* Please notice that alth	hough the wizard f			

Figure 20: NURO CaseStudy - Evaluation Step 13 Deployment in Progress

eaclouds.app.112v9nQh	Topology & Status Diagram Applic	ation Sensors		
pology & Status				
plication Monitor		(R)		
A		modacloudsDo_do	db	
move Application			. ()	
		Amazon eue co large us east	nuro-api	
		/		

Figure 21: NURO CaseStudy - Evaluation Step 14

Deployment succeeded.

![](_page_43_Figure_1.jpeg)

The latest and complexes topology deployment is shown in the next figure.

Figure 22: NURO CaseStudy - Complex Database Deployment

![](_page_43_Figure_4.jpeg)

Beside the continuous evaluation of advances in the SeaClouds Project, NURO spent much effort on research and testing of cloud based scaling solutions for their MySQL based approach. With sobering conclusion the actual generation of MySQL offerings is premature for seamless database scaling.

The design on the left with an alternative handmade approach to solve the database scaling obstacles was initiated in the end of the project.

In the end of the project NURO if the researches result for a suitable PaaS or Database as a Service provider was insufficient, NURO started to try hand made experiments on this.

Some options as p-mysql and the end 2015 announced MaxScale/MariaDB are promising approaches but were to late discovered to extend the SeaClouds Project at the Project's runtime.

To keep track on this topic NURO joined the SeaClouds Alliance as described in D7.4 The SeaClouds System itself reached impressive maturity.

# 3.4.2 ATOS usability testing

Usability Testi	ng 2 (UT2)	Date	31/03/2016	
Roi Sucasas (A	TOS)	Testbed	Local Testbed and Cloud Testbed	
		name		
SeaClouds Pla	tform Versio	<b>n</b> 0.9.0		
Test Results				
Involved	SeaClouds d	lashboard (SeaClouds deployer, SeaClouds		
Components	monitoring	tools)		
Interaction	The integrat	ion between the components was finalized at testing		
Between	time. The da	ashboard includes all the expected functionalities like		
Components	the topology	y design, and the components deployment and		
	monitoring.			
	SeaClouds	Impression		
	Dashboard	mpression		
	Wizard	Intuitive and ea	auto uso and understan	d
	navigation	intuitive and ea	sy to use and understan	u.
	Topology	Also intuitive a	nd easy to use. It offers a	lot of
	design	options in the c	lefinition of each of the a	application
		components.		
		LINK	^	6
		Description Softa		
		Operation type		
		Env var to store endpoint URL		
		softcarews		Softcar
		ForumDB ForumDB		
		2		Softca
		Close Edit		
		Figure 1: ATOS case study topology		
		Application properties	Design topology Optimize & Plan Configuratio	n summary O Generating
		Please choose one of th	e proposed setups by SeaClouds	AAM.
		3 location/s solution:	3 location/s solution:	3 location/s solution:
		Cloud_Foundry - (Cost p 0.04)	rr hour (€): Cloud_Foundry - (Cost per hour (€): 0.04)	Cloud_Foundry - (Cost per H 0.04)
		Properties - Hosted Services	Properties • Hosted Services	Properties • Hosted Services
		SoftcareWS	SoftcareWS	SoftcareWS
		SoftcareForum	SoftcareRMedia	SoftcareRMedia
		4	, SoftcareGU	•
		Back Next		
		Figure 2: List of	different Paas / Jaas on	tions for the
		final denlovme	nt	
		mai depioyille		

![](_page_46_Figure_1.jpeg)

Passed?	Yes	Bug ID
Problems	No	
Required	-	
Changes		
Cost	-	
Estimation		
Comments	We had to modify manually the generated DAM in order to add a specific <i>buildpack</i> <sup>1</sup> needed by one of the applications that was deployed in Pivotal / Cloud Foundry. Anyways this is a very particular case for a concrete PaaS provider.	

### 4. Conclusions

The deliverable D6.4.3 is the third version of the D.6.4.x. It reports the final evaluation measurements and analysis of the SeaClouds platform, by evaluating both quantitative and qualitative requirements. The information has been separated into two main sections; the section 2 introduced the different settings to perform a functional evaluation analysis while the section 3 details some non-functional evaluation analysis. Due to the fact the SeaClouds platform has to be considered a PoC; the consortium cancelled some non-functional tests to M30, as the platform didn't reach the necessary maturity level required to perform some of the non-functional testing activities.

Moreover, in this document (in the Annex B) a collection of tools to be used during the testing and validation phase of the SeaClouds project has been presented.

In summary, the final version of the SeaClouds platform proved a great part of the functionalities described in the deliverable D2.1 [2] although it is still missing some key features, like replanning.

From the point of view of chapter 3 (non-functional testing), the final release of SeaClouds shows still several severe stability issues that still prevent performing Performance/Scalability tests as well as Stress tests and Long-Term tests.

SeaClouds Platform can be considered a proof-of-concept of an interesting intuition. The final implementation is far from being a product that can be used in production.

<sup>&</sup>lt;sup>1</sup> A buildpack provides framework and runtime support for an application **→** <u>https://docs.cloudfoundry.org/buildpacks/</u>

# Annex 1: ATOS Case Study

The ATOS case study is about an e-health and social networking application system composed by several applications and modules that aim to easy the lives of elderly people, and also the work of the social workers and doctors that take care of them. The applications that compose this solution are the following:

### • Desktop application:

This .*NET* desktop application will be used by each one of the elderly users. It is ready to be deployed in PCs or small devices, and it is responsible for collecting the medical of these elderlies. This application is also responsible for offering them all the multimedia and social content of the solution.

### • Web services application:

This java Web application is responsible of the main logic of the application components. It is also responsible for the connections with the main database.

### • SoftCare Web GUI application:

This java Web application will be used by social workers and doctors in order to do the follow-up of the elderly people, and also to assign them social and multimedia content.

### • SoftCare Database:

This database stores the data of all users, including the medical data of the elderlies. This implies that the database has to be stored in a private environment that ensures a correct management of the privacy and confidentiality of the stored data.

# • Forum Web application & database:

This java Web application is responsible for maintaining a forum service for elderly people, their families and friends.

# • Multimedia repository application:

Finally, this application is responsible for the management of the multimedia content that is offered to the elderly people.

The architecture of this solution is depicted in the next image:

![](_page_49_Figure_1.jpeg)

Figure 23: ATOS Case Study architecture – SoftCare solution

The SeaClouds platform will be used to design, deploy and manage all the previous described Softcare applications / components, except the desktop application for elderly people, which is out of the SeaClouds scope.

# Annex 2: NURO Case Study

cused to find by the SeaClouds System a simple to use tool to find the best deployment solution for the game and adjust it to the games' needs. A mix of private and multi cloud resources should be possible.

To evaluate this NURO implemented a simplified server plugin based on their engines. In the simple setup it consists of a "Database" and a "PHP" module. More complex topologies have been evaluated. The final version has three PHP modules (NURO API, NURO GUI, and a Database Backend) and up to three databases.

For this project NURO developed some components to simulate load scenarios and to provide an API to interact with the seaclouds system see D6.3.3.

Based on a flexible implementation all components can be also accessed by any web browser. The response is HTML or JSON, we recommend the JSONview plugin to display JSON responses in a human friendly way.

D6.4.3. Deliverable Name - Annex 2: NURO Case Study 40

![](_page_50_Picture_1.jpeg)

The D6.3.3 major Components are:

- simulator.php NURO GUI Scenario Simulator and Monitor
- benchmark.php NURO GUI for Apache benchmarking tool
- sensor.php NURO API Sensor, returns server metrics
- effector.php NURO API Effector, accepts event requests
- analytics.php NURO API Analytics, returns runtime analytics
- chart.php NURO API Chart generator, returns performance charts

Also a quiz game server and client have been developed, included this components and others of NUROs engines.

1
👫 SeaClouds   × 👘 SeaClouds   × 🕒 52.31.203.17 × 📑 SeaClouds   ×
← → C f b52.31.203.170/nuro-casestudy/analytics.php?interval=
<pre>{     header: {         project: "SeaCloudsProject",         component: "NURO/analytics.php",         milestone: "MS9",         sourceversion: "D6.3.2",         protocol: "TBD",         version: "TBD",         release: "2015-09-30",         microtime: 1447324969.2982,         date: "Thu, 12 Nov 2015 10:42:49 +0000"     },     result: {         text: null,         analytics: [             + {},             microtime: 1447324960.2082,             date: "Thu, 12 Nov 2015 10:42:49 +00000"             }             - result: {</pre>
<pre>- {     time_group: "2015-11-12 10:35",     requests: "1065",     users: "1",     avg_run_time: "5.922210762310476",     messages: "5imulatorStart,Violation,SimulatorEnd"     },     + {-,,     + {-,,     }</pre>

Figure 24: NURO CaseStudy - analytics.php response

Figure 10 is an analytics result after a simulation with a SLA violation.

Both the "NURO simulator" and the "SeaClouds Monitoring" use the same effector.php API to report events.

- SimulatorStart
- Violation
- SimulatorEnd

These events are reported by the analytics.php at node: result.analytics[3].messages Based on this data charts are generated. The final pmplementation of the NURO CaseStudy has a simulation and performance monitoring GUI implemented.

Several usage scenarios can be tested and whatched with this component.

![](_page_51_Figure_11.jpeg)

Figure 25: NUROS CaseStudy - Simulator

D6.4.3. Deliverable Name - References 42

#### References

- [1]. SeaClouds D6.4.1 SeaClouds periodic evaluation reports
   <u>http://www.seaclouds-project.eu/deliverables/SEACLOUDS-</u>
   <u>D6.4.1 SeaClouds periodic evaluation reports.pdf</u>
- [2]. SeaClouds D2.1. Requirements for the SeaClouds Platfrom: <u>http://www.seaclouds-project.eu/deliverables/SeaClouds-D2.1-</u> <u>Requirements for the SeaClouds Platform.pdf</u>
- [3]. SeaClouds D6.1. Case study extended description <u>http://www.seaclouds-</u> project.eu/deliverables/SeaClouds-D6.1-Case study extended description.pdf
- [4]. ISO 9241-11:1998, Ergonomic requirements for office work with visual display terminals (VDTs) – Part 11: Guidance on usability, Retrieved from http://www.iso.org/iso/home/store/catalogue\_tc/catalogue\_detail.htm?csnum ber=16883.
- [5]. ISO/IEC 9126:1991. Information Technology Software Product Evaluation -Quality Characteristics and Guidelines for the User.
- [6]. https://en.wikipedia.org/wiki/Usability\_testing
- [7]. Initial architecture and design of the SeaClouds Platform
   <u>http://www.seaclouds-project.eu/deliverables/SeaClouds-D2\_2-</u>
   <u>Initial architecture and design of the SeaClouds platform.pdf</u>
- [8]. https://github.com/SeaCloudsEU/SeaCloudsPlatform/blob/master/ README.md#deploy-seaclouds