



---

## SeaClouds Project

### D6.4.1 - SeaClouds periodic evaluation reports

---

Project Acronym	SeaClouds
Project Title	Seamless adaptive multi-cloud management of service-based applications
Call identifier	FP7-ICT-2012-10
Grant agreement no.	610531
Start Date	1 <sup>st</sup> October 2013
Ending Date	31 <sup>st</sup> March 2016
Work Package	WP6 Case Studies implementation and Validation
Deliverable code	D6.4.1
Deliverable Title	SeaClouds periodic evaluation reports
Nature	Report
Dissemination Level	Public
Due Date:	M12
Submission Date:	10 <sup>th</sup> October 2014
Version:	1.0
Status	Final
Author(s):	Francesco D'Andria (Atos), Raffaella Mirandola (POLIMI)
Reviewer(s)	Ernesto Pimentel (UMA), Antonio Brogi (UPI)

## Dissemination Level

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

## Table of Contents

1. Executive Summary .....	4
2. Introduction .....	5
3. Scenarios .....	6
4. Criteria.....	7
4.1 Quantitative Analysis .....	7
4.2 Qualitative Analysis.....	7
5. Methods .....	8
5.1 Acceptance Testing/Specification Testing .....	8
5.2 Load Testing .....	8
5.3 Performance/Scalability Testing .....	8
5.4 Stress Testing .....	8
5.6 Documentation Testing.....	9
5.7 Regression Testing .....	9
5.8 Long Term Testing.....	9
5.9 Interoperability Testing.....	10
6. Tasks and Time Schedule .....	11
6.1 Local Evaluation .....	11
6.2 Evaluation as SaaS solution.....	11
6.3 Final Evaluation .....	11
7. Conclusions .....	13
8. References.....	14

## 1. Executive Summary

The purpose of this short deliverable is the description of the evaluation scenarios, criteria and methodology for evaluation of the SeaClouds software.

The requirements for SeaClouds are described in D2.1 [1] and D6.1 [2]. The initial requirements for the SeaClouds platform identified in those deliverables were considered to obtain the initial version of the architecture and design for the SeaClouds platform, which will be required to get the final conceptual SeaClouds reference platform [3].

The project works towards these requirements to support developers and application managers through the creation of an open source platform to support the deployment of applications over multiple-clouds, the monitoring of such deployments, and the migration of application modules across different (both public and private) cloud providers if needed.

The deliverable is structured as follows.

Section 3 introduces the methodology to evaluate whether SeaClouds functional requirements are achieved, while Section 4 lists the criteria to perform quantitative and qualitative analyses of non-functional requirements like usability, portability, and so on.

Section 5 describes the evaluation methodology to assess these requirements.

The clear description of tasks, responsibilities of different project partners and time schedules in Section 6 ensures the observance of the development plan given in the DoW.

## 2. Introduction

Task 6.3 of the WP6 deals with the evaluation of the SeaClouds platform.

Deliverable D6.1 defines the scenarios, criteria and methodology for such platform evaluation. Such work is strongly connected to the technical work implemented in work packages WP2, WP3, WP4 and WP5.

As many tasks are running in parallel to WP6, it is essential to separate between the tasks of testing in WP3 and WP4 and the evaluation of SeaClouds software inside WP6.

The evaluation includes quantitative and qualitative analysis of SeaClouds. Besides the functional requirements, the non-functional requirements like performance, user interface, installation procedures, ease of use, are tested.

At the end of this document a detailed evaluation schedule with detailed tasks and clear responsibilities of the project partners is delivered.

### 3. Scenarios

To evaluate the SeaClouds platform different kind of techniques can be applied. The first class comprises measuring techniques, and relies on quantitative information. Some examples will be discussed in Sections 4 and 5. The second class comprises techniques, in which one investigates how the platform reacts to certain situations. This can be done using scenarios [4]. Different types of scenarios can be used in platform assessments, for example [4]:

- *Use cases*: these are given by the users or can be derived from the requirements.
- *Change cases*: these describe possible ("what-if" like) situations.
- *Stress situations*: they describe extreme conditions under which the system still has to operate, for example, when the number of concurrent users of the system grows unexpectedly.
- *Far-into-the-future scenarios*: these are kind of visionary change cases.

For SeaClouds, scenarios based on use cases derived from requirements are adopted. Specifically, in Section 6 of D2.1 [1] of WP2 the consortium listed eight different complex use cases used as a basis to define the SeaClouds platform requirements.

- Create a Deployment Plan
- Define Service Level Agreement
- Manage Service Level Agreement
- Monitor periodically
- Monitor on events
- Initialize Application Deployment
- Update Deployed Application
- Application Administrator

The SeaClouds requirements have been summarized in Section 5.1 of the D2.1.

To check whether these complex functional requirements have been achieved, the consortium aims to implement evaluation scenarios based on the complex use cases mentioned before.

The evaluation will be performed in local and public testbeds and it will cover the local installation as well as the usability as SaaS solution.

## 4. Criteria

To evaluate SeaClouds platform it is necessary to perform a quantitative and qualitative analysis.

The quantitative analysis includes the testing of functional requirements and additional performance and stress tests. The qualitative analysis mainly concentrates on non-functional requirements like usability and portability. Examples of target non-functional requirements follow:

### Usability:

- Ease of installation
  - Ease of administration
  - Comparison to existing solutions
- ISVs / Cloud Application provider are interested in an easy to use solution. If the handling of SeaClouds is too complicated, this could impact of the level of acceptance of the solution.

**Reliability:** Once SeaClouds platform is functioning, as specified, and delivered, the reliability characteristic defines the capability of the system to maintain its service provision under defined conditions for defined periods of time. One aspect of this characteristic is fault tolerance that is the ability of a system to withstand component failure.

**Efficiency** as the characteristic that is concerned with the cloud resources consumed when providing the functionalities implemented by the SeaClouds platform.

**Portability** as the ability of the SeaClouds platform to run on numerous platforms.

### 4.1 Quantitative Analysis

The quantitative analysis mainly consists on the test of the functional and (some) non-functional requirements.

The functional requirements are mentioned in the evaluation scenarios of Section 3. In addition to these requirements it is necessary to carry out stress and load tests for SeaClouds platform to show the performance in real environments.

Section 5 contains more details about load and stress testing.

### 4.2 Qualitative Analysis

Besides fulfilling the functional requirements it is essential for a later “market-ready” application of SeaClouds platform to meet non-functional requirements such as usability and portability. For usability studies, for example, we directly observe how customers use technology (or not) to meet their needs. This provides the ability to ask questions, examine the behavior and in case suggest changes to meet the objectives. In this case, differently from the quantitative analysis, the data analysis is usually not mathematical.

## 5. Methods

To check the criteria defined in Sections 3 and 4, the following methods are adopted:

- Acceptance testing/Specification-based testing
- Load testing
- Performance/Scalability testing
- Stress testing
- Recovery testing
- Documentation testing
- Regression testing
- Long term testing
- Interoperability testing

An explanation of objectives and actions behind these methods is provided below.

### 5.1 Acceptance Testing/Specification Testing

Acceptance testing is usually an interactive test. Acceptance testing checks if the system meets the functional requirements as well as the non-functional requirements. A report is written specifying how close the system is to fulfil the requirements list and which changes are necessary to do so.

### 5.2 Load Testing

Load testing models the expected use of SeaClouds by simulating the simultaneous access from multiple users. During the load testing all actions and answers are monitored.

### 5.3 Performance/Scalability Testing

Performance of a system indicates the efficiency of the system while performing tasks. It includes total throughput of an operation as well as memory and disk space efficiency.

However, in SeaClouds we should not consider performance at Cloud Subsystem and Network Subsystem.

### 5.4 Stress Testing

Stress testing determines the behavior of the software while the offered load is in excess of its designed capacity. The system is deliberately stressed by pushing it to and beyond its specified limits. Stress tests are targeted to bring out the problems associated with one or more of the following:

- Memory leaks.
- Buffer allocation and memory carving.



In terms of the project the stress testing should concentrate to the SeaClouds components of the management service.

## 5.5 Recovery Testing

Recovery testing means the capacity to verify the recovery property of the SeaClouds platform during the failure of the software. It will be made in a variety of ways to verify that recovery is properly performed.

## 5.6 Documentation Testing

Documentation testing means verifying the technical accuracy and readability of the user manuals, including possible tutorials or online documentation. This test can be divided into two different sections

- Read Test: In this test documentation is reviewed for clarity, organization, flow, and accuracy without executing the documented instructions on the system.
- Functional Test: The instructions embodied in the documentation are followed to verify that the system works as it has been documented.

## 5.7 Regression Testing

Regression testing means a fully automated test of the software. A regression test deck contains a number of software tests including component tests as well as whole system tests. It is necessary to reproduce the results. Therefore the results are compared to stored reference solutions. The regression runs regular in predefined intervals. The size of the interval depends strongly on the time needed to perform the tests.

As the regression addresses component tests too, it should be done by WP3/WP4 or WP5. At the end of these WPs changes in one component should not affect other components. Therefore a set of automatic tests in a regression is eligible.

## 5.8 Long Term Testing

Long term testing is designed to ensure that the system remains stable for a long period of time under a high load. A system might function flawlessly when tested about some specific aspect, however, when a system runs for a long period of time without restarting, a number of problems are likely to occur: the system slows down, the system encounters functionality problems, the system silently fails over, and the systems crashes.

## 5.9 Interoperability Testing

Interoperability is the “ability to work with other systems”; in SeaClouds it means that we have to be able to guarantee component integration with both IaaS and PaaS unified APIs and third-party external application management (Add-ons).

## 6. Tasks and Time Schedule

The subsections 6.1, 6.2 and 6.3 give a clear description of tasks, responsibilities of different project partners and time schedules to ensure the observance of the development plan given in the DoW.

### 6.1 Local Evaluation

As prerequisite of task 6.1 the evaluation scenarios were defined and the tools for load testing, stress testing and long term testing are available. The local evaluation includes the following tasks:

- Acceptance testing
- Usability testing
- Load testing
- Stress testing
- Long term testing
- Security testing
- Documentation testing
- Example Application Management Test based on the Softcare App and the Cloud Gaming App (still preliminary version of the Case Studies)

Cloudsoft with the help of UMA and ATOS will lead the local evaluation of the SeaClouds platform. The outcome of this analysis will be presented in the deliverable D6.4.2 “Second SeaClouds periodic evaluation reports” delivered by the M24 of the project.

### 6.2 Evaluation as SaaS solution

- Installation
- Acceptance testing
- Documentation testing
- Usability testing

Cloudsoft will lead the evaluation as SaaS solution with the main help of ATOS. The outcome of this analysis will be presented in the deliverable D6.4.3 “Final SeaClouds periodic evaluation reports” delivered by the M30 of the project.

### 6.3 Final Evaluation

- Load testing
- Documentation testing
- Definition of example cases for the Softcare App
- Definition of example cases for the Cloud Gaming App

Cloudsoft will lead the final evaluation of the SeaClouds platform. The outcome of this analysis will be presented in the deliverable D6.4.3 “Final SeaClouds periodic evaluation reports” delivered by the M30 of the project.

## 7. Conclusions

The Deliverable 6.4.1 is the first document of D6.4.x saga (the D6.4.2 will be delivered on M24 and D6.4.3 will be delivered on M30).

The purpose of these deliverables is: on one hand the definition and detailed description of the evaluation scenarios, criteria and methodology; on the other hand report about the evaluation analysis.

For evaluation it is necessary to test quantitative criteria-functional requirements and qualitative criteria/non-functional requirements. The evaluation scenarios are based on the functional requirements and use cases defined in D2.1 [1] and D6.1 [2].

## 8. References

- [1]. Requirements for the SeaClouds Platform: [http://www.seaclouds-project.eu/deliverables/SeaClouds-D2.1-Requirements\\_for\\_the\\_SeaClouds\\_Platform.pdf](http://www.seaclouds-project.eu/deliverables/SeaClouds-D2.1-Requirements_for_the_SeaClouds_Platform.pdf)
- [2]. Case study extended description [http://www.seaclouds-project.eu/deliverables/SeaClouds-D6.1-Case\\_study\\_extended\\_description.pdf](http://www.seaclouds-project.eu/deliverables/SeaClouds-D6.1-Case_study_extended_description.pdf)
- [3]. Initial architecture and design of the SeaClouds Platform [http://www.seaclouds-project.eu/deliverables/SeaClouds-D2\\_2-Initial\\_architecture\\_and\\_design\\_of\\_the\\_SeaClouds\\_platform.pdf](http://www.seaclouds-project.eu/deliverables/SeaClouds-D2_2-Initial_architecture_and_design_of_the_SeaClouds_platform.pdf)