SeaClouds Project
D2.1 Requirements for the SeaClouds Platform

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	1st October 2013
Ending Date	31st March 2016

Work Package	WP2, WP Requirements Analysis, overall Architecture and Standardization
Deliverable code	D2.1
Deliverable Title	Requirements for the SeaClouds Platform
Nature	Report
Dissemination Level	Public
Due Date:	M5
Submission Date:	28st February 2014
Author(s):	Antonio Brogi (UPI), Javier Cubo (UMA), Francesco D'Andria (ATOS), Elisabetta Di Nitto, Luca Florio and Raffaela Mirandola (POLIMI), Christian Tismer (NURO), Andrea Turli (CloudSoft)
Reviewer(s)	Roman Sosa (ATOS), PengWei Wang, Michela Fazzolari (UPI)
Status	Final

Table of Contents

Executive Summary 4
1. Introduction 5
   1.1 Glossary of Terms 7
2. The requirement analysis approach 11
   2.1 Business goals, domain assumptions and requirements 11
   2.2 Domain description 13
   2.3 Use cases 14
3. SeaClouds business goals 14
   3.1 SeaClouds Business Goals Summary 20
4. SeaClouds domain assumptions 21
   4.1 SeaClouds Domain Assumptions Summary 25
5. SeaClouds requirements 25
   5.1 SeaClouds Requirements Summary 34
6. Use cases for the SeaClouds Platform 36
   6.1 SeaClouds Use Cases Summary 44
7. SeaClouds Domain description 45
8. Ideal Scenario - How the requirements will be mapped in a real scenario 46
9. Conclusion 48
Annex 1: Details on important concepts 49
References 52

List of figures

Figure 1 The relationship between the world and a system to be developed (the machine). 12

Figure 3 Use Case Diagram 43

Figure 3 Domain Description 45

List of tables

Table 1 Glossary of Terms 7

Table 2 Business Goal, Domain Assumption and Requirements description templates 12

Table 3 Use Case Description template 14

Table 4 SeaClouds Business Goals Summary 20

Table 5 SeaClouds Domain Assumptions Summary 25

Table 6 SeaClouds Requirements Summary 34

Table 7 SeaClouds Use Cases Summary 44

Executive Summary

The current document describes the main requirements for the SeaClouds Platform. Requirements are collected in a structured form. In particular, we identify the main business goals driving the SeaClouds development taking into account the needs of both our end users and other external stakeholders potentially interested in the project results; we set the boundaries of our platform by defining a set of domain assumptions and we translate business goals in requirements taking into account these domain assumptions.

In order to better clarify the domain space and the way our platform is going to be used, we provide also a formalization of the domain and a number of use cases.

1. Introduction

SeaClouds focuses mainly on supporting design and execution of service compositions on multiple clouds. The main scenario we have in mind is the following:

> *The Application Designer creates an application as a composition of services and defines the application topology, that is, a model describing the application modules together with inter-module relationships. He/she then exploits the SeaClouds Planner to associate to the service composition some cloud capabilities. These can be: containers that support the execution of application services, communication middleware, storage services, SaaS (Software as a Service) already available on the cloud. In doing its work, the planner takes into consideration the non-functional requirements (Mylopoulos et al 92) that the Application Designer associates to the composition and to each of its parts. The outcome of the planner is a Deployment Plan that is deployed using state of the art tools.*
>
> *The Application Administrator deploys the application by running the Deployment Plan on the SeaClouds Deployer. Moreover, he/she is going to exploit the SeaClouds Monitor to monitor the status of the system. The Monitor is able to automatically trigger a reconfiguration of the application in case of need.*

As concerns the main human actors of the SeaClouds Platform, the following have been devised so far:

- *Application Designer*: this is the one who designs a service composition and interacts with the planner to obtain a Deployment Plan.
- *Cloud provider*: this is the one providing some Cloud services. It does not necessarily interact directly with the SeaClouds Platform, but the services offered are exploited by the platform to run service compositions.
- *Application administrator*: this is the one who oversees to the correct execution of the service composition deployed on multiple clouds.

The following of this document describes the main requirements for the SeaClouds Platform. The approach we use to elicit and describe these requirements is based on the classical Requirements Engineering literature (Jackson 1995) and leverages from the results achieved by NEXOF-RA (NEXOF-RA 2009) and S-Cube (S-Cube, 2009).

This deliverable is structured as follows:

- In Section 2 we provide some more details on the approach we adopt for requirement description.
- Sections from 3, 4 and 6 are the core of the deliverable and contain a definition of business goals, domain assumptions, and requirements for the SeaClouds Platform.

- Section 5 describes the domain in which the SeaClouds Platform is supposed to be used.
- Section 7 offers a description of how the system can be used by the various stakeholders.
- Finally, Section 8 describes an ideal scenario of usage of the SeaClouds platform and Section 9 concludes the deliverable.

For the sake of clarity, all terms introduced in all sections of this deliverable are defined in the Glossary section available below.

1.1 Glossary of Terms

Table 1 Glossary of Terms

Term	Meaning
SeaClouds Platform (SCP):	*The system/platform to be developed in the project*
Requirements	*The set of requirements for the SCP*
Business goals	*The objectives pursued by the SCP*
Domain Assumptions	*The assumptions and constraints that are used as premises by the SCP*
Application	*The software that is given by an Application Designer to the SCP, in order to be deployed on different clouds.*
Application Design	The action of composing different modules to create an application
Application Modules	*The set of modules that compose an application*
Application Topology	*Application modules together with inter-module relationships*
Deployment Plan	*The description of how the modules of an application should be deployed on the clouds*
Orchestration Specification	*Specification of the application topology plus the deployment plan*
User Input	*The input given by the user to the SCP. It should include a) the set of application modules (i.e. the application) b) the inter-module relationships c) QoS requirements and technology requirements*
Inter-Module Relationships	*The way the application modules are connected among them*
QoS Requirements	*The set of QoS conditions specified by the*

	Application Designer that the SCP should fulfill
Technology Requirements	*Features the applications (modules) need to run correctly*
Application Administrator	*The person who oversees to the correct execution of the application deployed on multiple clouds*
Application Designer	*The person who designs an application and interacts with the SCP to obtain a deployment plan*
Service Level Agreement (SLA)	*A contract between a cloud service provider and a customer that specifies (in measurable terms) what services the cloud provider will furnish*
Dashboard	*The instruments for administration of services distributed between cloud platforms (deploy, stop, start, and update applications)*
SeaClouds API	*The unified API (and universal metrics) for monitoring and verifying functional and non-functional properties. It will allow the reusing and modularity, deployment and monitoring operations (performed by the SeaClouds Controller).*
Standardization activities	*The activities of alignment of SeaClouds's architecture with major standards for cloud interoperability, particularly OASIS' CAMP and TOSCA, promoting them in research and industrial communities*
SeaClouds Planner	*The SeaClouds functionality (it acts as only one component plus the orchestration specification) in charge of implementing planning policy to orchestrate the multi-cloud deployment of the application modules*

SeaClouds Controller	*The SeaClouds functionality in charge of implementing the multi-cloud deployment of the application modules and SeaClouds monitoring policy. It is composed of the SeaClouds Monitor, SeaClouds Analyzer and the SeaClouds Deployer*
SeaClouds Deployer	*The SeaClouds functionality in charge of taking the input the orchestration specification generated by the Planner, and deploying (by exploiting the Multi-Cloud Deployment API) the application modules on the specified clouds*
SeaClouds Monitor	*The SeaClouds functionality in charge of monitoring (by exploiting the Monitoring API) that the QoS properties of the application modules are not violated by the clouds in which they were deployed, and that the whole application satisfies the QoS properties specified for the whole application*
SeaClouds Analyzer	*The SeaClouds functionality in charge of generating the reconfiguration suggestions (if needed) to be passed as inputs to the Planner module to trigger the generation of a new adaptive orchestration plan*
Reconfiguration Suggestions	*The suggestions to inform the Planner regarding cloud functionalities that need to be replaced to satisfy (technology and QoS) requirements.*
Violation Notification	*The information that an SLA has been violated.*
Discovery API	*The API to discover the capabilities and add-ons featured by available clouds*
Deployment API	*The API to deploy the application modules on the specified clouds according to the orchestration specification*

Monitoring API	*The API to monitor the QoS properties of the modules of the applications, of the whole application, and of the platform, as well as the contract specified in the SLA to check the required and agreed compliance*
Boom scenario	
Burst scenario	

2. The requirement analysis approach

2.1 Business goals, domain assumptions and requirements

The main objectives of the requirements analysis are (Jackson 1995):

1. To identify the business goals the system to be developed is going to address and the detailed requirements it has to fulfill.
2. To understand the application domain for which the system is going to be developed
3. To determine the boundaries of the system to be developed with respect to the external world.

Figure 1 describes the distinction between the system to be (the machine) and the world in which the system is going to operate. The set of *phenomena* occurring in the world together with the laws that regulate such world (e.g., physical laws, social rules, conventions that need to be respected) define the *application domain*. In the case a software system (a machine) is needed in order to fulfil certain business goals, such machine needs to have an impact on the world. Thus, the two corresponding domains have to partially overlap. The phenomena that are at the intersection between the world and the machine are called *shared phenomena*. These can be either controlled by the world and observed by the machine, or, conversely, controlled by the machine and observed by the world. Understanding the application domain means that we need to identify the phenomena in the world that are relevant for our system to be (in the example of Figure 1 these are: the occurrence of incidents, the fact that the involved persons or the spectators call for help, the fact that such calls are encoded in the system, the fact that ambulances move in the city and that the system is able to track the current location of an ambulance).

Business goals for our system are prescriptive assertions that are expressed in terms of the world phenomena and that should be fulfilled when the system will be in operation (in the example of Figure 1 the goal could be “An ambulance should move to the incident as soon as this is notified by some citizen”). *Requirements* are similar to business goals, but they are expressed only in terms of the phenomena that are in the intersection between the world and the machine (again, in the example of Figure 1 the requirement could be “As soon as the outcome of a call is encoded into the system, then this last one should identify the ambulance that is closest to the incident location and should inform the ambulance operators about the incident”). Requirements express the desires of the customer concerning the application. *Domain assumptions* are descriptive assertions assumed to hold in the world and intended to describe the laws that hold in the world and are relevant to the application. For instance, in the example of Figure 1 a domain assumption is the following “the information about a call for help are always correctly encoded in the system”.

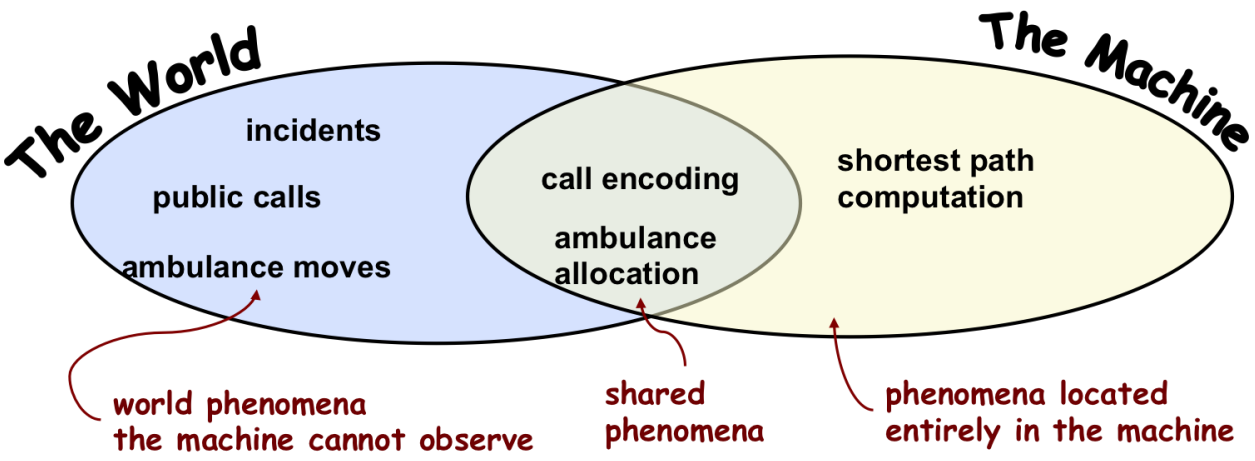

Figure 1 The relationship between the world and a system to be developed (the machine).

The requirements R are complete if R ensure satisfaction of the goals G in the context of the domain properties D (R and D |= G), given that G adequately capture all the stakeholders needs and D represents valid properties/assumptions about the world.

The main purpose of the requirement analysis phase within SeaClouds is to identify G, D, and R. To support this identification, we have defined a template for describing them (see Table 1). The description includes the involved stakeholders, the rationale, the priority and the material supporting the description, if any.

Table 2 Business Goal, Domain Assumption and Requirements description templates

Field	**Description**
Unique ID	Give a unique ID for this goal/requirement/assumption (name of partner + Goal/Requirement/Assumption + number)
Short name	Give a short name for this goal/requirement/assumption
Type	One of the following: ❏ Business goal ❏ Domain Assumption ❏ Requirement
Business goal this requirement refers to	This is only for requirements and allows to define a traceability between business goals and requirements

Description	Specify the intention of the goal/requirement/assumption
Rationale	Give a justification of the goal/requirement/assumption
Involved Stakeholder (only for business goals and requirements)	Stakeholders involved in the business goal/requirement
Supporting materials	Give a pointer to documents that illustrates and explain this goal/requirement/assumption (in particular those of domain analysis)
Priority of accomplishment (only for business goals and requirements)	One of the following: ❏ Must have: The system must implement this goal/requirement to be accepted. ❏ Should have: The system should implement this goal/requirement: some deviation from the goal/requirement as stated may be acceptable. ❏ Could have: The system should implement this goal/requirement, but may be accepted without it.
Tentative scheduling (only for business goals and requirements)	Tentative scheduling of accomplishment.

2.2 Domain description

As we have already mentioned, the study of world phenomena is particularly important in the requirement analysis phase since they are useful to define the interface between the machine and the world. Of course, shared phenomena (and therefore scenarios) can be understood in the context of the world in which the machine will work and of the laws governing the world. In general, phenomena are related with each other and it is important to make such relations explicit. For the above reasons, it is important to include in the requirement analysis deliverable a definition of such application domain. *Class diagrams* are usually a good tool for this purpose since they allow the engineer to identify main phenomena as classes and to express several kinds of relationships between these. Entity-relationship diagrams as

well as semantic networks for our purposes have an expressive power that is similar to class diagrams and therefore can be used as well.

2.3 Use cases

The phenomena shared between the world and the machine can be detailed through use cases descriptions. Use cases have an operational flavour in the sense that they describe the steps that need to be followed by the machine and the world entities in order to accomplish a certain task.

Table 2 describes how use cases should be detailed and described, and it should be used as a template for any single use case description. Here, a use case is described using information about the business goals or the domain assumptions they refer to, the operational description of the use case, the possible problems involved and the supporting material.

Table 3 Use Case Description template

Field	**Description**
Unique ID	Give a unique ID for this use case (name of partner + UseCase + number)
Short name	Give a short name for this use case
Participating actors	Specify the actors involved in the use case
Flow of events	Describe the flow of event characterizing the use case
Exceptions	Specify possible exceptions that should be handled
Non-functional Requirements	Describe the non-functional requirements of the use use

3. SeaClouds business goals

Field	**Description**
Unique ID	Goal1

Short name	OrchestrationPlanning
Type	Business goal
Description	The SeaClouds Platform, in particular, the Planner, must be able to support the Application Designer in the identification of the cloud capabilities suitable for a certain application (expressed as an orchestration of services). We also call this activity Matchmaking. Specifically, SeaClouds planner will input the specification by the Application Designer and exploit Discovery API to generate a standard specification for distributing Application Modules onto available clouds. We call this *Deployment Plan*. Such a specification will be passed to the SeaClouds Controller (WP4), and, in particular, to the SeaClouds Deployer that will deploy the Application Modules of the application over multiple clouds according to the Deployment Plan.
Rationale	SeaClouds planner will be in charge of determining a distribution of Application Modules onto multiple available cloud capabilities so that QoS and Technology Requirements for individual group of Application Modules are satisfied.
Involved Stakeholder	The Application Designer
Supporting materials	WP3 description
Priority of accomplishment (only for business goals and requirements)	Must have: The system must implement this goal to be accepted.
Tentative scheduling (only for business goals and requirements)	Task 3.3. Planning the multi-cloud deployment (M6-M22)

Field	Description
Unique ID	Goal2
Short name	Monitoring and Analysis
Type	Business goal
Description	SeaClouds must monitor and analyze the current performance of the Application Modules which are deployed and running in different clouds, as well as of the whole application.
Rationale	SeaClouds controls the cloud-to-cloud migration upon monitored QoS violations. A violation is detected by comparing the established QoS properties and the actual performance values obtained by monitoring the application execution and the platform. It has to be noted that the QoS properties can be specified for the whole application and for the individual Application Modules. The system can, for example, detect the need of load-balancing or distribution of Cloud services on several Cloud providers.
Involved Stakeholder	The Application Administrator
Supporting materials	WP4 description (DoW - Workplan Tables, Page 22)
Priority of accomplishment (only for business goals and requirements)	Must have: The system must implement this goal to be accepted.
Tentative scheduling (only for business goals and requirements)	Task 4.1. Monitoring, verification and traceability of QoS violations (M6-M22)

Field	Description
Unique ID	Goal3
Short name	Governance
Type	Business goal
Description	SeaClouds should manage capabilities across multiple Cloud providers. Moreover, it should manage the complete lifecycle of an Application Module (deployment, configuration, management and migration)
Rationale	We aim at fully support the execution of multi-cloud orchestrations by controlling the capabilities it is using as well as the execution of its Application Modules.
Involved Stakeholder (only for business goals and requirements)	Application administrator
Supporting materials	See the description of WP4 in the DoW
Priority of accomplishment (only for business goals and requirements)	Must have: The system must implement this goal/requirement to be accepted.
Tentative scheduling (only for business goals and requirements)	To be defined.

Field	Description
Unique ID	Goal4
Short name	Reconfiguration

Type	Business goal
Description	SeaClouds must support the process of migrating Application Modules of a cloud application distributed in heterogeneous cloud platforms.
Rationale	SeaClouds must generate suggestions for reconfiguration plans when the monitoring or the Application Administrator detects that properties, such as QoS, are not respected or whenever an offer for a new, more convenient, cloud capability is advertised. If the application administration accepts the reconfiguration suggestions, SeaClouds reconfigures applications dynamically by changing the orchestration of the services (rescheduling, migration and re-execution of the Application Modules). A possible reconfiguration is the automatic addition/deletion of cloud capabilities. In boom scenarios resources must be added immediately without human interaction to ensure QoS. In burst scenarios automatic release of resources is needed to avoid unneeded payment. This automation reduces administrative effort and waste of resources.
Involved Stakeholder	The Application Administrator (via internal Application Modules)
Supporting materials	WP4 description (DoW - Workplan Tables, Page 22), WP6. (D6.1, section 4.2)
Priority of accomplishment (only for business goals and requirements)	Must have: The system must implement this goal to be accepted.
Tentative scheduling (only for business goals and requirements)	Task 4.2. Dynamic reconfiguration of multi-cloud applications (M6-M22)

Field	Description
Unique ID	Goal5
Short name	SeaClouds API
Type	Business goal
Description	SeaClouds will manage applications deployed on several technologically diverse Cloud platforms, unifying operations such as monitoring and lifecycle management, promoting the adoption of standards for cloud interoperability. Hence, SeaClouds will leverage on the OASIS CAMP and TOSCA specifications for what concerns the interaction of the cloud platforms on top of which the SeaClouds Orchestration Specifications will be deployed and run. SeaClouds will actively contribute to the standardization effort of CAMP by implementing a CAMP-compliant interface towards PaaS providers for the orchestration and the monitoring of cloud-enabled applications over multiple-Clouds infrastructure. On the other hand, SeaClouds will also focus on assessing existing implementation of the TOSCA specification by developing as well new functionalities (that are deliberately out of scope of the specification) to solve issues about policies for the dynamic management of service orchestrations.
Rationale	We want to make sure that we adopt existing standards to increase the impact of the project. Nevertheless, SeaClouds will not depend on the level of success of CAMP and/or TOSCA, since although they will be main standards considered by SeaClouds, other standardization efforts could be also considered if needed. Therefore, SeaClouds will not be affected by a possible

	delay or even failure of CAMP and/or TOSCA standardization activities.
Involved Stakeholder (only for business goals and requirements)	Application Administrator, Cloud provider
Supporting materials	WP6. (D6.1, section 4.2)
Priority of accomplishment (only for business goals and requirements)	Must have: The system must implement this goal/requirement to be accepted.
Tentative scheduling (only for business goals and requirements)	To be defined.

3.1 SeaClouds Business Goals Summary

Table 4 SeaClouds Business Goals Summary

Unique ID	Short name	Stakeholder	Priority	Tentative scheduling
Goal1	Orchestration Planning	Application Designer	Must have	Task 3.3. Planning the multi-cloud deployment (M6-M22)
Goal2	Monitoring and Analysis	Application Administrator	Must have	Task 4.1. Monitoring, verification and traceability of QoS violations (M6-M22)
Goal3	Governance	Application Administrator	Must have	To be defined.
Goal4	Reconfiguration	Application Administrator (via	Must have	Task 4.2. Dynamic reconfiguration of

		internal Application Modules)		multi-cloud applications (M6-M22)
Goal5	SeaClouds API	Application Administrator, Cloud provider	Must have	To be defined.

4. SeaClouds domain assumptions

Field	**Description**
Unique ID	Assumption1
Short name	Specification Of Orchestration
Type	Domain Assumption
Description	The Application topology is specified by exploiting a standard specification, such as TOSCA (by means of a CSAR, Cloud Service Archives) or CAMP (via the PDP, Platform Deployment Package).
Rationale	This assumption is required because SeaClouds does not focus on the definition of application topologies, but use them as inputs.
Supporting materials	TOSCA and CAMP specifications. The Task 2.2. Architecture and design of the SeaClouds Platform (M4-M9) develops the architecture and design of the SeaClouds Platform, and among the activities of this task is included the decision about the Orchestration Specification. Therefore, the specification to be used to describe the Deployment Plan will be taken in this task, previous to the Task 3.3 Planning the multi-cloud deployment (M6-M22), where the SeaClouds Planner (knowing the specification to be used) has to be designed and developed.

	Section B.1.5. Significant risks and contingency plan (DoW - Part B, Page 45) - Risk 8 - WP2 - Specifications (mainly CAMP and TOSCA) adopted by SeaClouds projects are obsolete, while new standadrs should be considered in the SeaClouds for different tasks, including the specification of the Deployment Plan (Probability of this risk: Low)).

Field	**Description**
Unique ID	Assumption2
Short name	Specification of QoS requirements and properties
Type	Domain Assumption
Description	The Designer API will allow Application Designer to specify desired (hard/soft) QoS and Technology requirements for each group of Application Modules as well as (hard/soft) QoS requirements for whole application. It will also allow Application Designer to specify which Application Modules should be deployed on multiple cloud. The formalism to express QoS properties should be WS-Policy compatible. The Application Designer characterizes the application to be deployed on multiple clouds a1)The Application Designer MUST provide the Application Modules that compose the application, together with the set of inter-module relationships (e.g., Module A *communicates with* Module B, Module A *is hosted on* Module B). (a2) For each (group of) Application Modules the Application Designer SHOULD provide a set of (hard/soft) technology and QoS requirements.

	(b) The Application Designer MAY also provide QoS requirements for the whole application. /* (b) can be executed before (a1) and/or (a2). */
Rationale	SeaClouds Platform must give the Application Designer the ability to describe the need and resources consumption depending on QoS rules. This includes (hard/soft) QoS and Technology requirements, groups of application module and inter-module relationships. Technology requirement should also allow to set a minimum and a maximum for automatic consumption and automatic release of resources. It should also allow definition of notifications, warnings and alerts depending on triggers (fixed threshold values or extreme delta values).
Supporting materials	Task 3.2 Specification of properties and requirements (M6-M22) develops Designer API to suitably express the Application Designer input.

Field	**Description**
Unique ID	Assumption3
Short name	Availability of Tools For Monitoring
Type	Domain Assumption
Description	The cloud providers have tools to monitor several kind of properties and information (CPU, latency, memory, resources).
Rationale	SeaClouds is not focusing on developing tools to provide monitoring.
Supporting materials	External tools like VMware vFabric Hyperic, ProdEagle, JMX, Sigar, Collectl, and others.

Field	Description
Unique ID	Assumption4
Short name	Security delegated to Cloud Providers
Type	Domain Assumption
Description	We expect system hardening is implemented by the cloud provider. For example intrusion prevention and detection or dos prevention. Also security updates. Application Designer and Application Administrator want to focus on the Application and only to take care of security issues on the own Application.
Rationale	Cloud resources could be very different. In the NURO case study a “worker module” just has PHP 5.x as fixed requirement. The operating system or web server to use is not a fixed requirement. It would overstrain the Application Administrator to harden an unknown mix of systems.
Supporting materials	

Field	Description
Unique ID	Assumption5
Short name	No need for data synchronization
Type	Domain Assumption
Description	We assume that the applications created and deployed with SeaClouds do not require any data synchronization when (part of) their components are migrated from one cloud to another.
Rationale	Data synchronization is in general a complex task and it is outside the boundaries of the project.
Supporting materials	

4.1 SeaClouds Domain Assumptions Summary

Table 5 SeaClouds Domain Assumptions Summary

Unique ID	Short name
Assumption1	Specification Of Orchestration
Assumption2	Specification of QoS requirements and properties
Assumption3	Availability of Tools For Monitoring
Assumption4	Security delegated to Cloud Providers
Assumption5	No need for data synchronization

5. SeaClouds requirements

Field	Description
Unique ID	Requirement1
Short name	QoS dependent resource definition
Type	Requirement
Business goal this requirement refers to	Orchestration Planning
Description	SeaClouds Platform must give the Application Designer the ability to describe the need and resources consumption depending on QoS rules. There should be a minimum and a maximum for automatic consumption and automatic release of resources. Also a definition of notifications, warnings and alerts depending on triggers. Examples for triggers are fixed threshold values or extreme delta values.
Rationale	This is needed for Goal4
Involved Stakeholder (only for business goals and requirements)	Application Designer and Application Administrator
Supporting materials	
Priority of accomplishment	Must have: The system must implement this

(only for business goals and requirements)	goal/requirement to be accepted.
Tentative scheduling (only for business goals and requirements)	To be defined

Field	**Description**
Unique ID	Requirement2
Short name	Service Level Agreement Definition
Type	Requirement
Business goal this requirement refers to	Orchestration Planning
Description	A Service Level Agreement (SLA) is the foundation of the Cloud consumer's (Application Designers and Administrators) trust in the Cloud provider. SeaClouds has to support the definition of that agreement that describe the relationship between cloud providers and cloud consumers. SLA contains Service Level Objectives (SLOs) that define objectively measurable conditions for the service. In the Orchestration Planning, after the technological matchmaking, the planner must weigh the terms of the SLA and its SLOs (of the suitable Clouds) against the goals of its business application to select a best-fit cloud provider where the application (part of it) will run. SLA must be assessed at runtime (see the requirement “ATOSRequirements5”)
Rationale	The SLA is an important element of the agreement between service providers and consumers. As such, it has to be kept under control and recovery actions have to be taken in case it cannot be fulfilled.

Involved Stakeholder (only for business goals and requirements)	Application Administrator
Supporting materials	Task 3.2 Specification of properties and requirements (M6-M22) to allow Application Designer specifying which are the Application Modules to be deployed on multiple clouds and the desired QoS properties for the SLA of the whole application.
Priority of accomplishment (only for business goals and requirements)	Must have: The system must implement this goal/requirement to be accepted.
Tentative scheduling (only for business goals and requirements)	To be defined

Field	**Description**
Unique ID	Requirement3
Short name	SLA Assessing and Violation Management
Type	Requirement
Business goal this requirement refers to	Monitoring and analysis, reconfiguration
Description	If the Monitoring system identifies an issue related to the violation of the SLA, procedures for enforcing the SLA or for managing the violation through some reconfiguration need to be actuated.
Rationale	The SLA is an important element of the agreement between service providers and consumers. As such, it has to be kept under control and recovery actions have to be taken in case it cannot be fulfilled.
Involved Stakeholder (only for business goals and requirements)	Application Administrator
Supporting materials	Task 3.2 Specification of properties and requirements (M6-M22) to allow Application Designer specifying which are the Application Modules to be deployed on

	multiple clouds and the desired QoS properties for the SLA of the whole application. On the other hand the WP4, Task 4.1. Monitoring, verification and traceability of QoS violations (M6-M22) provides monitoring mechanisms to assess SLA Agreements agreed in the design time. SLA negotiation
Priority of accomplishment (only for business goals and requirements)	Must have: The system must implement this goal/requirement to be accepted.
Tentative scheduling (only for business goals and requirements)	To be defined

Field	**Description**
Unique ID	Requirement4
Short name	Metric-Driven Policy-Based Management
Type	Requirement
Business goal this requirement refers to	Governance
Description	The SeaClouds Platform must allow the Application Administrator to observe real-time operational metrics. In response to these metrics, the Application Administrator — or an automated process such as a management system — can affect changes to the resources managed by the application as a response to a runtime requirement.
Rationale	Policies perform the active management on SeaClouds. They can subscribe to sensors and be triggered by them or they can run periodically to perform calculations, look up other values and invoke effectors, i.e. for autoscaling.
Involved Stakeholder (only for business goals and requirements)	The Application Administrator

Supporting materials	
Priority of accomplishment (only for business goals and requirements)	Must have: The system must implement this goal/requirement to be accepted.
Tentative scheduling (only for business goals and requirements)	To be defined

Field	**Description**
Unique ID	Requirement5
Short name	Comprehensive Graphical User Interface
Type	Requirement
Business goal this requirement refers to	All
Description	SeaClouds must provide a smart GUI where Application Designer, Application Administrator and cloud provider can: ● Access and view the cloud capabilities listed in a service catalogue. ● Search and browse for the best-fit capabilities (IaaS, PaaS) based on application requirements (functional and non-functional) ● Add information about cloud capabilities on the service catalogue, in particular: 1. Technical features offered by the cloud capabilities. 2. Non-Functional characteristic like QoS, Location, Security, etc. ● Load Application Modules belonging to the application. ● Access capabilities to monitor and govern the application and violations of QoS policies.
Rationale	Without a proper GUI the SeaClouds Platform will not be well accepted by its potential users.
Involved Stakeholder (only for business goals and requirements)	Application Designer, Application Administrator, Cloud Provider

Supporting materials	See WP5 description in the DoW
Priority of accomplishment (only for business goals and requirements)	Must have: The system must implement this goal/requirement to be accepted.
Tentative scheduling (only for business goals and requirements)	To be defined

Field	**Description**
Unique ID	Requirement6
Short name	The SeaClouds platform must rely on standard APIs and languages
Type	Requirement
Business goal this requirements refers to	SeaClouds API
Description	SeaClouds will produce a deployment plan that will be compatible with the CAMP API. Moreover, it will be able to receive as input a TOSCA or a CAMP compatible topology specification.
Rationale	This is to fulfill the SeaClouds goal to be compatible with related standards.
Involved Stakeholder (only for business goals and requirements)	Application Designer and Administrator
Supporting materials	DoW
Priority of accomplishment (only for business goals and requirements)	Must have: The system must implement this goal/requirement to be accepted.
Tentative scheduling (only for business goals and requirements)	To be defined.

Field	**Description**
Unique ID	Requirement7
Short name	Application Migration
Type	Requirement

Business goal this requirement refers to	Reconfiguration
Description	SeaClouds has to support Software Application portability (its Application Modules) and its data between cloud platforms that use the same underlying technology.
Rationale	Migrating to a different cloud zone or even to a different cloud is a way to handle a need for reconfiguration.
Involved Stakeholder (only for business goals and requirements)	Application Administrator
Supporting materials	None for the moment
Priority of accomplishment (only for business goals and requirements)	Must have: The system must implement this goal/requirement to be accepted.
Tentative scheduling (only for business goals and requirements)	To be defined

Field	**Description**
Unique ID	Requirement8
Short name	Orchestration Specification reuse to build different plans
Type	Requirement
Business goal this requirements refers to	Orchestration Planning, Governance
Description	An Orchestration Specification can be reused to build different plans and executed more than one time and can, therefore, lead to many different instances, possibly coexisting at the same time. In the Nuro case study, for instance, if a new Game with the same engine (a Reskin) is launched, a new Deployment Plan for the Orchestration Specification should be created.
Rationale	A Reskin is just an exchange of assets (images, texts, etc) and other values. The system design

	(i.e., the structure of the orchestration) is untouched.
Involved Stakeholder (only for business goals and requirements)	Application Administrator
Supporting materials	WP6. (D6.1, section 4.2)
Priority of accomplishment (only for business goals and requirements)	Should have: The system should implement this goal/requirement: some deviation from the goal/requirement as stated may be acceptable.
Tentative scheduling (only for business goals and requirements)	To be defined.

Field	**Description**
Unique ID	Requirement9
Short name	Application updates
Type	Requirement
Business goal this requirement refers to	Governance
Description	SeaClouds Platform shall support the Application Administrator to deploy application updates without enforced downtime. Application updates could be due to software updates, database definition or data updates. We assume that the update is correct and that, if data synchronization is needed, this is handled by the person in charge of the update. SeaClouds is not able to check for this correctness. Also, SeaClouds does not support zero downtime updates, but we will try to keep the update time low.
Rationale	After deployment software is often under development. Bugfixes, database and data changes or new functionality make system updates a must have.
Involved Stakeholder (only for	The Application Administrator

business goals and requirements)	
Supporting materials	WP6. (D6.1, section 4.2)
Priority of accomplishment (only for business goals and requirements)	Should have: Application updates with minimal downtime is a must, application updates with zero downtime is a nice to have requirement.
Tentative scheduling (only for business goals and requirements)	To be defined

5.1 SeaClouds Requirements Summary

Table 6 SeaClouds Requirements Summary

Unique ID	Short name	Business goal Referred to	Stakeholder	Priority	Tentative scheduling
Requirement1	QoS dependent resource definition	Orchestration Planning	Application Designer and Application Administrator	Must have	To be defined
Requirement2	Service Level Agreement Definition	Orchestration Planning	Application Administrator	Must have	To be defined
Requirement3	SLA Assessing and Violation Management	Monitoring and analysis, reconfiguration	Application Administrator	Must have	To be defined
Requirement4	Metric-Driven Policy-Based Management	Governance	The Application Administrator	Must have	To be defined
Requirement5	Comprehensive Graphical User Interface	All	Application Designer, Application Administrator, Cloud Provider	Must have	To be defined
Requirement6	The SeaClouds	SeaClouds API	Application Designer and	Must have	To be defined

	platform must rely on standard APIs and languages		Application Administrator		
Requirement7	Application Migration	Reconfiguration	Application Administrator	Must have	To be defined
Requirement8	Orchestration Specification reuse to build different plans	Orchestration Planning, Governance	Application Administrator	Should have	To be defined.
Requirement9	Application updates	Governance	Application Administrator	Should have	To be defined

6. Use cases for the SeaClouds Platform

Field	Description
Unique ID	UseCase1
Short name	Create a Deployment Plan
Participating actors	Application Designer
Flow of events	1. The Application Designer starts the SeaClouds Planner 2. The Application Designer inputs the service composition plus Non-functional requirements and constraints 3. The SeaClouds Planner answers with a possible association of cloud capabilities to the composition 4. The Application Designer reviews the proposed solution 5. If needed, the Application Designer removes some of the proposed associations and adds further requirements and constraints 6. The interaction starts again from step 3 7. SeaClouds planner produce the Deployment plan as result.
Exceptions	None
Non-functional Requirements	The SeaClouds Planner should provide an answer within 10 minutes

Field	Description
Unique ID	UseCase2
Short name	Define Service Level Agreement
Participating actors	Application Designer, SeaClouds Planner
Flow of events	1. The Application Designer starts the SeaClouds Planner 2. The Application Designer inputs the service composition plus functional and Non-functional requirements and constraints - As part of the Non-functional requirements the Application Designer

	detail the Global Application QoS and the Application Modules QoS to be enforced. 3. The SeaClouds Planner answers with a possible Deployment Plan that matches with the QoS requirements specified in the point2. 4. The Application Designer reviews the proposed alternatives and chooses the best-fit (for his) topology 5. If needed, the Application Designer removes some of the proposed associations and adds further requirements and constraints 6. The interaction starts again from step 3 7. When the Application Designer finalizes the process, the Deployment Plan is enriched with a Service Level Agreement that will be assessed at runtime.
Exceptions	
Non-functional Requirements	

Field	**Description**
Unique ID	UseCase3
Short name	Manage Service Level Agreement
Participating actors	Application Administrator, SeaClouds Platform
Flow of events	1. While the application runs the SeaClouds Platform assesses the SLA 2. When a violation of the SLA occurs, the SeaClouds Platform looks for the best recovery action already agreed at design time. 3. The SeaClouds Platform enforces the recovery action a. Some recovery actions could be enforced automatically (scale, replication etc). Some others should be just notified waiting for the human intervention (migration, undeployment) 4. The SeaClouds Platform stores the violation and any useful information about it.
Exceptions	

Non-functional Requirements	

Field	**Description**
Unique ID	UseCase4
Short name	Monitor periodically
Participating actors	The “Time” on behalf of the Application Administrator (the real actor of the system)
Flow of events	This use case is executed periodically. The period is configurable at run time. 1. The monitoring period expires. 2. The system starts taking measures of the magnitudes involved in the QoS properties. 3. The system analyzes the application behaviour from the collected samples. 4. The system stores the collected samples for further (post-portem) analysis. 5. No violation detected. The use case finishes.
Exceptions	5.a A violation is detected. 5.a.1 The system reports that a QoS property violation has been found. 5.a.2 The use case finishes.
Non-functional Requirements	Monitoring period. TBD. Response Time for reconfiguration suggestions creation?

Field	**Description**
Unique ID	UseCase5
Short name	Monitor on events
Participating actors	The “Agent, Robot, Spider or Observer” on the behalf of the Application Administrator
Flow of events	Applications are “instrumented” for every action to be monitored to include an event generator that reports the system. 1. An “agent inserted as observer” detects an event to be monitored.

	2. The “Agent” reports the event to be monitored to the SeaClouds controller. 3. The controller starts taking measures. 4. The controller analyzes the application behaviour from the collected samples. 5. The controller stores the collected samples for further (post-portem) analysis.
Exceptions	6.a A violation is detected. 6.a.1 The system reports that a QoS property violation has been found. 6.a.2 The use case finishes.
Non-functional Requirements	Response Time for reconfiguration suggestions creation?

Field	**Description**
Unique ID	UseCase6
Short name	Initialize Application Deployment
Participating actors	The Application Administrator, The SeaClouds Platform
Flow of events	Requirement: “UseCase1: Create a Deployment Plan” 1. The Application Administrator adds the Application Software Package 2. The Application Administrator prepares configuration 3. The Application Administrator prepares and adds initialisation scripts 4. Start up 4.1. The Application Administrator initiates the start up 4.2. SeaClouds Platform starts resources in the cloud 4.3. SeaClouds Platform deploys the Application 5. The Application Administrator initiates the initialization 6. The Application is running well. The use case finishes.
Exceptions	4.a Startup Fails on composition failure

	4.a 1 Continue with UseCase1 4.b Startup Fails on configuration error 4.b 1 Continue with 2 5.a Initialisation Fails 5.a 1 The Application Administrator stops the resources 5.a 1 The Application Administrator analyzes the issue 5.a 2 Continue with 3.
Non-functional Requirements	The Application Administrator needs access to the Application Software and Asset Repositories

Field	**Description**
Unique ID	UseCase7
Short name	Update Deployed Application
Participating actors	The Application Administrator, SeaClouds Platform/Deployer
Flow of events	Requirement: Successful “UseCase6: Initialize Application Deployment " 1. The Application Administrator updates the Application Software Package 2. The Application Administrator updates the configuration (if needed) 3. The Application Administrator prepares and adds update scripts (if needed) 4. Update 4.1. The Application Administrator initiates the update 4.2. The SeaClouds Platform deploys the updates 4.3. The SeaClouds Platform initiates the scripts (if needed) 5. The SeaClouds Platform restarts resources (if needed) 6. The Application is running well. The use case finishes.

Exceptions	4.a The Update Fails on configuration error 4.a 1 Continue with 2 5.a Restart Fails 5.a 1 The Application Administrator analyzes the issue 5.a 2 Continue with 1, 2 or 3.
Non-functional Requirements	The Application Administrator needs access to the Application Software and Asset Repositories

Field	**Description**
Unique ID	UseCase8
Short name	Application Administrator reconfigures the application deployed on multiple clouds.
Participating actors	Application Administrator
Flow of events	1. The SeaClouds Monitor (WP4) will notify the Analyzer (WP4) in case of violations of any QoS and Technology Requirements. 2. The Analyzer (WP4) will generate reconfiguration suggestions for the Planner (WP3). These suggestions will inform the Planner regarding cloud functionalities that need to be replaced to satisfy (technology and QoS) requirements. 3. The Planner, supervised by the Application Administrator, will take reconfiguration suggestions as input along with previously available user input. 4. Once the decision has been taken, the Planner generates a new Deployment Plan for the Application Modules.
Exceptions	1a. No violation is detected by Monitor (WP4). In this case, Analyzer will not be notified and no reconfiguration suggestion will be generated. 2a. If the Analyser does not generate any reconfiguration suggestion (viz., it decides to take no action for the moment), the Application Administrator may be allowed to provide them so as to trigger the reconfiguration

	process.
Reference	Description of WP4/Task 4.2 in the DoW.
Internal note	Please note that the planner will use the inputs already provided by the Application Administrator i.e., 1. Application Modules that compose the application, together with the set of inter-module relationships. 2. For each (group of) Application Modules, a set of (hard/soft) technical and QoS requirements. 3. QoS requirements for the whole application.

The following Use Case Diagram summarize the described use cases

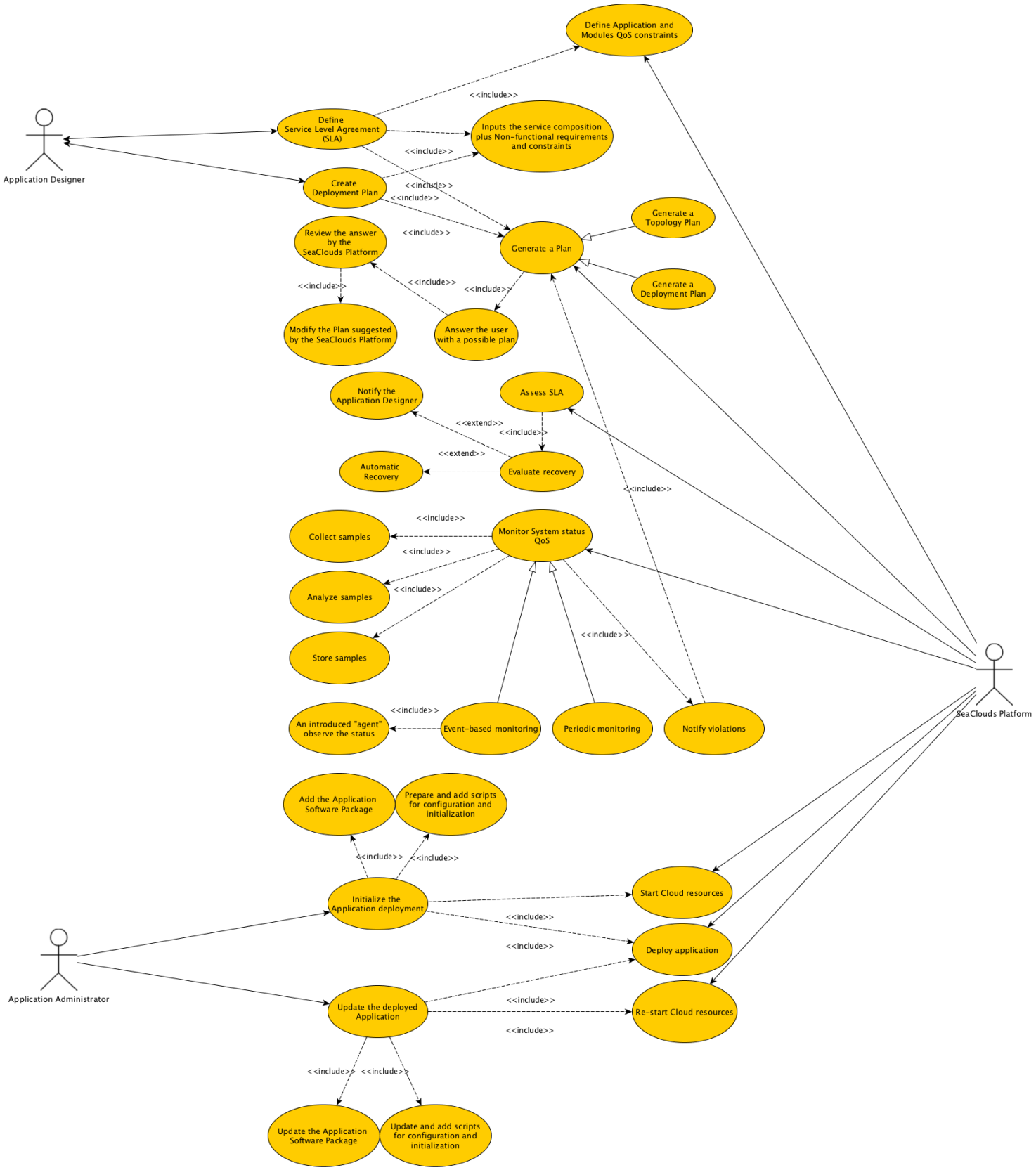

Figure 2 Use Case Diagram

6.1 SeaClouds Use Cases Summary

Table 7 SeaClouds Use Cases Summary

Unique ID	Short name	Actors	Non-functional Requirements
UseCase1	Create a Deployment Plan	Application Designer	The SeaClouds Planner should provide an answer within 10 minutes
UseCase2	Define Service Level Agreement	Application Designer, SeaClouds Planner	N/A
UseCase3	Manage Service Level Agreement	Application Designer, SeaClouds Platform	N/A
UseCase4	Monitor periodically	The "Time" on behalf of the Application Administrator (the real actor of the system)	Monitoring period. TBD. Response Time for reconfiguration suggestions creation?
UseCase5	Monitor on events	The "Agent, Robot, Spider or Observer" on the behalf of the Application Administrator	Response Time for reconfiguration suggestions creation?
UseCase6	Initialize Application Deployment	The Application Administrator, The SeaClouds Platform	The Application Administrator needs access to the Application Software and Asset Repositories
UseCase7	Update Deployed Application	The Application Administrator, SeaClouds Platform/Deployer	The Application Administrator needs access to the Application Software and Asset Repositories
UseCase8	Application Administrator reconfigures the application deployed on multiple clouds.	Application Administrator	N/A

7. SeaClouds Domain description

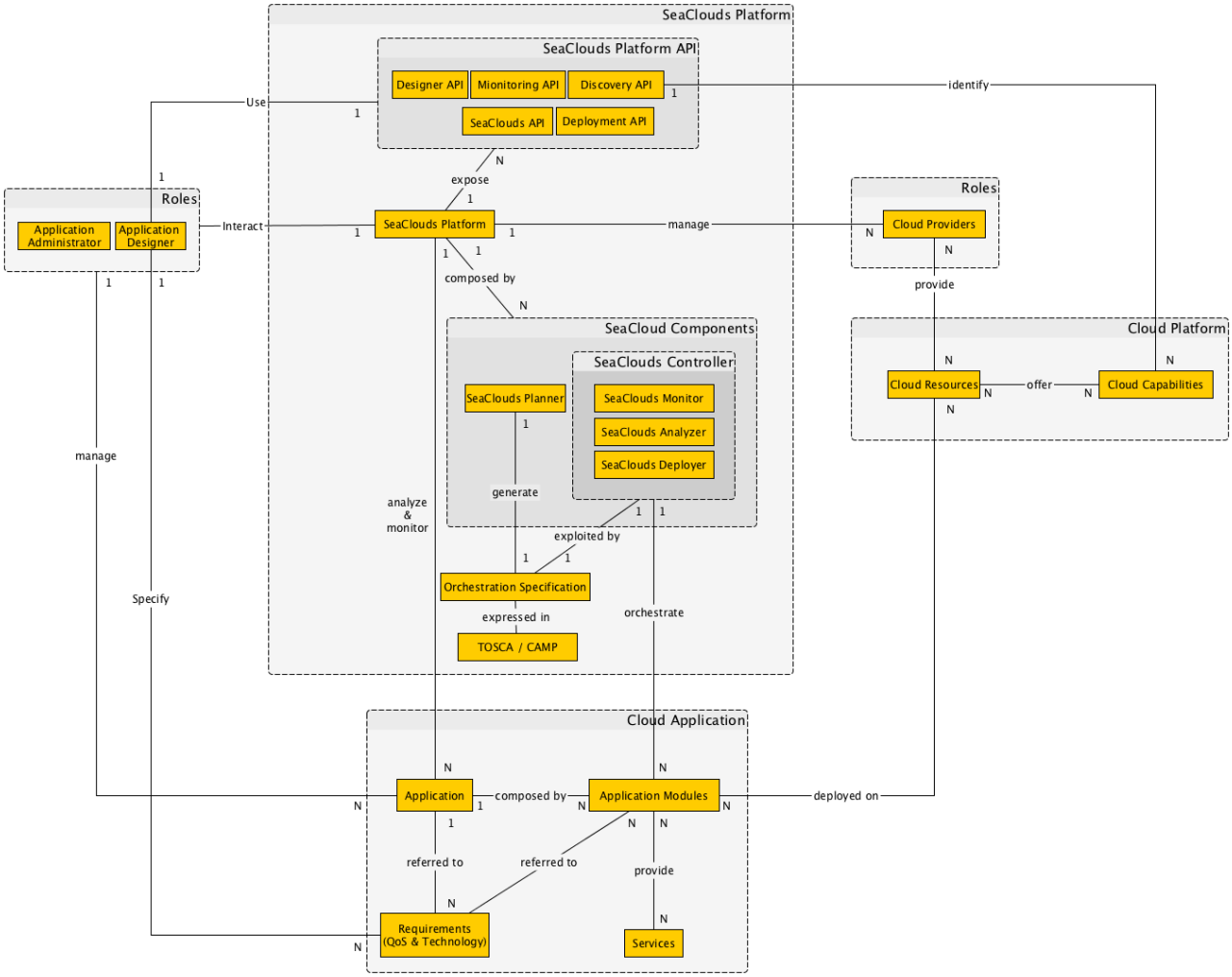

Figure 3 Domain Description

The diagram shows the Application Domain of the SeaClouds Platform.

- **Roles**: human actors of the SeaClouds Platform. The Application Administrator oversees the correct execution of the application. The Application Designer design the Application as an orchestration of services and interact with the SeaClouds Platform through the exposed SeaClouds API. He interact mainly with the SeaClouds Planner in order to plan the deployment of services and the distribution of Application Modules among multiple available Cloud Resources. Cloud Providers provide the Cloud Resources (which offer some Cloud Capabilities) where the Application Modules are deployed. Cloud Providers have tools to monitor several kind of properties and information about the status of Cloud Resources (CPU, latency, memory, etc.).

- **Cloud Application**: the Application is expressed as an orchestration of services. The Application is composed by some Application Modules, which are deployed on the available Cloud Resources. The Application as well as the Application

Modules must satisfy some QoS and Technology Requirements defined by the Application Designer

- **Cloud Platform**: The Cloud Platform is the set of all the Cloud Resources provided by different Cloud Providers. The Cloud Resources offer some Cloud Capabilites which are exploited by the SeaClouds Platform.

- **SeaClouds Platform**: SeaClouds Platform interact with the Application Administrator and the Application Designer providing information about the status of the system as well as tools to orchestrate the deployment of Application Modules into the available Cloud Resources. SeaClouds Platform exposes SeaClouds API (Designer API, Discovery API, etc.) to support the Application Designer in the analyses of Cloud Capabilities offered by the available Cloud Resources, and in the creation of an effective Orchestration Specification. The Orchestration Specification is expressed in TOSCA or CAMP and is generated by the SeaClouds Planner starting from the specifications provided by the Application Designer. The Orchestration Specification is exploited by the SeaClouds Controller (composed by the SeaClouds Monitor, SeaClouds Analyzer and the SeaClouds Deployer) to orchestrate the deployment of Application Modules to the available Cloud Resources. SeaClouds Platform monitors and analyzes the status of the Application to check the violation of QoS constraints, and support the process of migrating Application Modules distributed in heterogeneous Cloud Platforms. The SeaClouds Platform is able to manage Cloud Resources depending on QoS Requirements and other limits.

8. Ideal Scenario - How the requirements will be mapped in a real scenario

The following scenario builds on the innovations that SeaClouds will deliver in Cloud Software Engineering and exemplifies how this will benefit the European IT ecosystem. SocialCareSP is a Spanish software development company with a main product line specialized in IT e-health and social care for Public Administration. SocialCareSP takes care of the development of customized Application Modules to accommodate new functional requirements.

This customization involves application integration with existing databases (i.e. Oracle Database), as well as existing applications or legacy business systems at the customer's site. These customers require assured Quality of Service (QoS), such as high-availability for the provided services during the working hours, or Low Response Time, scalability and low operational costs for batch analytic workloads running after-hours. In addition

since the nature of the Application the databases SocialCareSP has to integrate in the application have specific security requirements.

SocialCareSP fulfils these quality requirements with a capacity management consultancy team following the application life-cycle at the customer's site.
However, SocialCareSP customer requirements have evolved rapidly in the last few years due to competitive advancements and growth of the Cloud market (reduced spending on technology infrastructure, improved accessibility, improved flexibility and QoS).
To remain competitive, SocialCareSP's solution must evolve to address these changing requirements. To do so, the SocialCareSP needs to apply advanced software engineering methodologies revising both its software development process and life-cycle management services:

1. The delivered product must support integration, orchestration and adaptation of the existing application software with a broad spectrum of customer PaaS/IaaS, and possibly, SaaS solutions in order to exploit customer data locality.
2. It needs to be deployed/replicated/distributed on several Multi-Cloud infrastructures providing quality assurance to avoid that availability or performance outages of a single Cloud provider would turn into a disaster for SocialCareSP's business and customers.
3. It has to provide a cloud-agnostic application having a flexible architecture that could be adapted to new Cloud offerings emerging in the upcoming years to adapt to changes of context and requirements.
4. The life-cycle management team, rather than providing support at the customer's site, must now support a system deployed on multi-Clouds infrastructures.

SeaClouds project supports SocialCareSP in the implementation of the described requirements.

Using the SeaClouds Graphical User Interface, SocialCareSP can design a complex cloud-based application combining existing components into new software modules that operate directly on multiple and heterogeneous Clouds. Indeed, SeaClouds only requires that the software modules are deployable on the cloud, so the application does not need to be rewritten to work (UseCase6; UseCase1 ; UseCase7).

Through the SeaClouds GUI, SocialCareSP also matches the functional and Non-functional requirements of each one the modules belong to the application finding the best-fit topology with the desired Service Level Agreement (UseCase2).

SeaClouds runs workloads on different clouds depending on the customer preference and the application functional as well as Non-functional requirements and hence enjoy the benefits of the multiple-cloud deployment (UseCase1; UseCase8).

SeaClouds benefits SocialCareSP by providing the capability of deciding the best-fit Cloud to adopt, comparing cloud services offered by each supplier, analysing their characteristics (i.e. QoS) and improving trust in Cloud solutions.

Finally, run-time operation is easily integrated inside SocialCareSP's software development process through SeaClouds's feedback system that automatically provides recommendation or raise Service Level Agreement violation about possible abnormal behavior of the Complex Application (UseCase3; UseCase4; UseCase5).

Thus, thanks to SeaClouds, SocialCareSP can remain a leader in its sector and neutralise competitive advantages of emerging North American and Asian start-ups that have developed their solutions directly on the Cloud.

9. Conclusion

This deliverable contains the requirement-related information that has been produced by the SeaClouds consortium in the first project phase. All assumptions, requirements and use cases identified in this deliverable will be further analysed in the next phases of the project, will be associated to specific architectural components, and then implemented within the SeaClouds platform.

Annex 1: Details on important concepts

- QoS Properties
  - Availability and performance of apps and systems, appropriateness of apps to their daily job
  - Monitor and measure technical indicators: throughput level, jitter, delay, response time, new features, scalability requirements, new app introductions, manageability, security and even cost

- Technology Requirements
  - Programming language support, type of databases, execution models, runtime environments, data management and networking constraints.

- SLA (ranging from specifications to monitoring, management and enforcement)
  - Agreement between service provider and customer about the required QoS characteristics
  - Defines the QoS and expresses itself generally in percentages of contractual objectives
  - The contract defines methods of evaluation and measure of the quality to avoid imprecision
  - Measuring: definition, provision, availability, performance, measurement, accuracy, security, affordability, customer attention, response time, maintenance, penalty, legal considerations, contract terms
  - Metrics: percentage of time services will be available, number of users served simultaneously, specific performance benchmarks, schedule for notification of network changes, help desk response time for different problems, dial-in access availability, usage statistics provided
  - SLAs for cloud services focus on characteristics of the data center and more recently include characteristics of the network to support end-to-end SLAs

Example of "Technology and QoS Requirements" to be included in the Deliverable 2.1 (based on the data for the SparQL request from Cloud Pier & Cloud4SOA):

Application Profile

- Application details
  - Title
  - Version
  - Programming Language
  - Application code
  - License type
  - Compute scaling factor (optional value)

- Web scaling factor (optional value)

- Software component
  - SQL Database (parameters about SQL database)
    - Category (MySQL, Oracle, Postgres, SQL server)
    - Database min capacity (Storage unit: Kb / Mb / Gb / Tb – optional value)
    - Database min cache size (Storage unit: Kb / Mb / Gb / Tb – optinal value)
  - NoSQL Database (parameters about noSQL database)
    - Category (Blobstore, Datastore, Mongo DB, S3)
    - Database min capacity (Storage unit: Kb / Mb / Gb / Tb – optional value)
    - Database min cache size (Storage unit: Kb / Mb / Gb / Tb – optinal value)
  - Generic software component (generic SW components and any other add-ons)
    - Computation category (app server, auth service, memcache, database, monitoring, performance, security, webserver)

- Hardware component
  - Network resource (network parameters needed by the app)
    - Category (network support, communications media – Ethernet / Fibre channel)
    - Max latency (Networking unit: ms / sec / min / hours – optional value)
    - Min bandwidth (Storage unit: Kb / Mb / Gb / Tb – optional value)
  - Compute resource (parameters about the computational power needed by the app)
    - Category (Azure core, EC2, Gear)
    - Architecture
    - Cache (Storage unit: Kb / Mb / Gb / Tb – optional value)
    - Number of cores (optional value)
    - Memory (Storage unit: Kb / Mb / Gb / Tb – optional value)
    - Computational power factor (optional value)
  - Storage resource (parameters about the storage HW needed by the app)
    - Category (FileSytem)
    - Bandwidth (Networking unit: Kb/s / Mb/s / Gb/s / Tb/s – optional value)
    - Capacity (Storage unit: Kb / Mb / Gb / Tb – optional value)

- HTTP request handler (parameters about the HW need to manage the HTTP requests to the app)
  - Category (Azure core, EC2, Gear)
  - Computational power factor (optional value)
  - Number of HTTP requests (optional value)

- QoS / Service Level Agreement (to be completed)
  - Uptime (%)
  - Maximum latency (ms)
  - CPU load (%)
  - Memory load (%)
  - Cloud response time (ms)
  - Container response time (ms)

References

(Mylopoulos et al 92) Mylopoulos, J.; Chung, L.; Nixon, B., "Representing and using nonfunctional requirements: a process-oriented approach," *Software Engineering, IEEE Transactions on* , vol.18, no.6, pp.483,497, Jun 1992

(Jackson 1995) Jackson, M. Software requirements & specifications: a lexicon of practice,
principles and prejudices. New York, NY, USA: ACM Press/Addison-Wesley
Publishing Co. 1995

(NEXOFRA, 2009). NESSI Open Framework Reference
Architecture (NEXOFRA): Scenarios and Requirements for Open Framework Construction. 2009.

(S-Cube, 2009) Elisabetta Di Nitto, Valentina Mazza, Andrea Mocci, Collection of industrial best practices, scenarios and business cases, CD-IA-2.2.2 S-Cube Deliverable 2009, available from
http://www.s-cube-network.eu/results/deliverables/wp-ia-2.2/CD-IA-2.2.2_Collection_of_industrial_best_practices_scenarios_and_business_cases.pdf/view

Cloud Computing Service Level Agreements - Exploitation of Research Results European Commission http://ec.europa.eu/digital-agenda/en/news/cloud-computing-service-level-agreements-exploitation-research-results

Cloud4Soa Project http://www.cloud4soa.eu/

Cloud Pier Project http://www.opencloudpier.org/