Secure Provisioning of Cloud Services
based on SLA Management

SPECS Project - Deliverable 5.1.2

Description of the validation scenarios and identification of common elements

Version no. 1.0
31 October 2015

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

Deliverable information

Deliverable no.:	D5.1.2
Deliverable title:	Description of the validation scenarios and identification of common elements
Deliverable nature:	Report
Dissemination level:	Public
Contractual delivery:	31 October 2015
Actual delivery date:	31 October 2015
Author(s):	Stefano Marrone (CeRICT)
Contributors:	Silvio La Porta (EMC), Alessandra de Benedictis (CeRICT), Marina Bregou (CSA), Giancarlo Capone (CeRICT)
Reviewers:	Madalina Erascu (IeAT), Ruben Trapero (TUDA)
Total number of pages:	77

Executive summary

This deliverable contains part of the results of the second year of activity of T5.1. In particular, this document deals with the methodological, planning and coverage aspects related to the user-oriented system level testing of the SPECS software, products and artifacts.

After this premise, this deliverable describes the advancements during the second year of work carried on in this task. These advancements can be mainly described as:

- The testing activity has been framed into the entire project validation and testing process also concerning integration testing (focusing more on the correct communication among the software components, see T1.5) and unit testing (focusing more on the correct implementation of the software components, see T2.3, T3.4 and T4.5);
- The set of the Validation Scenarios (VSs), defined during the first year, has been improved. New scenarios have been added while the existing ones have been refined according to the actual design and implementation changes, given by the feedbacks of stakeholders and market analysis, significantly enhanced during Y2 on the basis on the availability of prototypes as described in detail in WP6;
- Six Validation Applications (VAs), which support the execution of the VSs, have been defined by User Stories and solution portfolio;
- The coverage level of the five Key Concerns has been improved, and the coverage measurement has also been enriched by the percentage of the executed VSs and tested requirements.

This deliverable reports that the level of coverage of the five Key Concerns satisfies all the planned goals for Y2 (see 7.2.2). The average coverage of the Key Concerns reached by VS specifications is 89% while the VAs available at Y2 accomplish by their executions an average percentage of coverage of 53% among all the Key Concerns.

As stated in this deliverable, the testing process of the SPECS platform and modules does not end with this deliverable since some activities will progress in the last six months of the project, and their results are fundamentals for the verification of the correctness and the quality of the developed products.

Table of contents

Index of figures

Index of tables

1. Introduction

1.1. Objective of the task

This deliverable describes the definition of validation and usage scenarios for the SPECS architecture and services (the main goal of Task 5.1).

The results of this activity depend on both SPECS architecture and service requirements. A proper methodology and validation strategy takes into account the SPECS architecture while the requirements for SPECS Services (mainly Core Services) are necessary to define Validation Scenarios (VSs). The overall picture of the activities and the process followed in Task 5.1 is depicted in Figure 1.

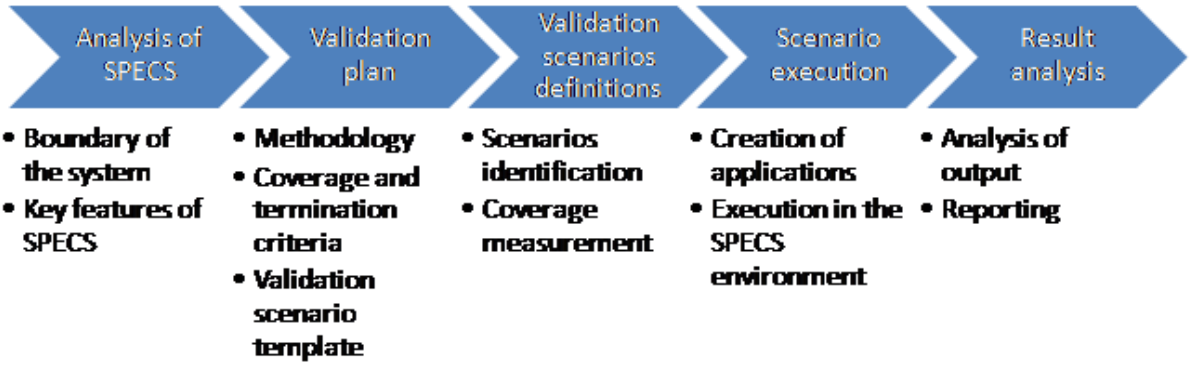


Figure 1. SPECS validation overall process

We accomplished the main aim of the task through the following steps:

1) Analysis of SPECS:
    a. definition of the boundary[1] of the SPECS system under test;
    b. definition of the key concerns of the platform and services.
2) Validation plan:
    a. selection of the level at which the SPECS Platform will be tested;
    b. identification of the significant SPECS features under test;
    c. definition of the coverage and termination criteria;
    d. definition of the *Validation Scenario template* and guidelines for the specification of VSs.
3) Validation Scenarios definition:
    a. identification of the single scenarios involved in the validation of SPECS;
    b. a detailed description of each VS;
    c. detection of the covered items (by each scenario).
4) Scenarios execution:
    a. preparation of the Validation Applications (VAs) running the VSs;
    b. execution of the VAs.
5) Results Analysis:
    a. analysis of the outputs of the execution;
    b. creation of a report summarizing the outcomes of the validation campaign.

[1] We intend for boundary of a system the limit separating what is under test from what is needed to test the system (e.g., CPSs are not in the boundary of the system under test).

This process has been customized starting from classical software testing processes. All the proposed steps are distributed over the 18 months in which the task is active and are collected in three deliverables according to the SPECS DoW. Figure 2 depicts how the steps are distributed over the three deliverables of this task: M12 and M24 indicate the project months at which the deliverables should be released; according to the SPECS project timeline, they respectively stand for October 2014 and October 2015.

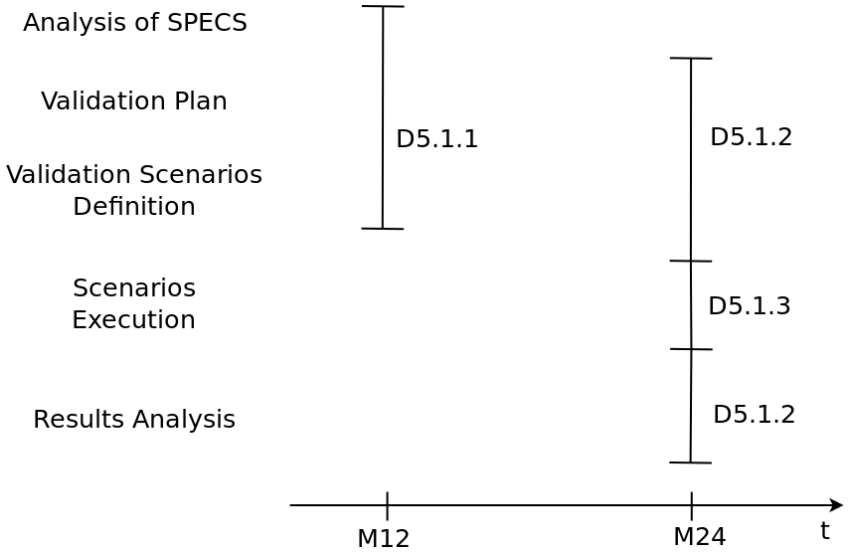


Figure 2. Task 5.1 timeline

Hence, according to the Figure 2, this deliverable contains the improvements of the Validation Plan methodology, the refinements of the VSs and the analysis of the Key Concern coverage.

1.2. Changes with respect D5.1.1

This section summarises the changes between this deliverable and its previous version D5.1.1. These advancements are:

- The testing activity has been framed into the entire project validation and testing process also concerning integration testing (focusing more on the correct communication among the software components, see T1.5) and unit testing (focusing more on the correct implementation of the software components, see T2.3, T3.4 and T4.5);
- The set of the Validation Scenarios (VSs), defined during the first year, have been improved. New scenarios have been added while the existing ones have been refined according to the actual design and implementation changes, given by the feedbacks of stakeholders and market analysis, significantly enhanced during Y2 on the basis of the availability of prototypes as described in detail in WP6;
- Six Validation Applications (VAs), which support the execution of the VSs, have been defined by User Stories and solution portfolio;
- The coverage level of the five Key Concerns has been improved and the coverage measurement has also been enriched by the percentage of the executed VSs and tested requirements.

To provide the reader a finer grained map of the changes, Table 1 reports the list of main updates with respect to D5.1.1.

Section	Subsection	Note
1	1.1	Quite unchanged with respected D5.1.1
	1.2	Added with respect D5.1.1
	1.3	New content with respect D5.1.1
2	-	New content with respect D5.1.1
3	3.1	Added with respect D5.1.1
	3.2	Background information: summary of what described in D5.1.1
	3.3	Added with respect D5.1.1
4	4.1	Additon of new VSs and refinement of D5.1.1 VSs in compliance with the latest version of module interactions
	4.2	Refinement of VSs reported in D5.1.1 in compliance with the latest version of module interactions. Removal of some VSs
	4.3	Removal of VSs of D5.1.1.
	4.4	Refinement (update to version 2.0) and removal of VSs reported in D5.1.1. One VS has also been added.
	4.5	Refinement (update to version 2.0) of VSs reported in D5.1.1. Addition of three new VSs.
5	-	Added with respect D5.1.1
6	6.x	Added with respect D5.1.1
7	7.x	New content with respect D5.1.1
8	-	New content with respect D5.1.1

Table 1. Changes with respect D5.1.1

1.3. Deliverable organization

This deliverable is structured as follows. Section 2 relates this deliverable with other deliverables and documents of the project. Section 3 reminds the main validation concepts of D5.1.1 and gives the improvement of the testing methodology added in the second year. Section 4 reports all the refined VSs. Section 5 describes the methods used to estimate the coverage of the requirements. Section 6 defines the VAs used to implement the testing scenarios. Section 7 reports the measurement of the coverage. Section 8 ends the deliverable. Appendix A enumerates the Key Concerns used to evaluate the quality of the testing activity. Appendix B reports the tables mapping the specified Validation Scenarios on the SPECS components and mapping the Components on the implemented Requirements.

2. Relationship with other deliverables

This deliverable relies upon some antecedent and contemporary deliverables. Figure 3 represents the dependency relationships among this deliverable and other deliverables.

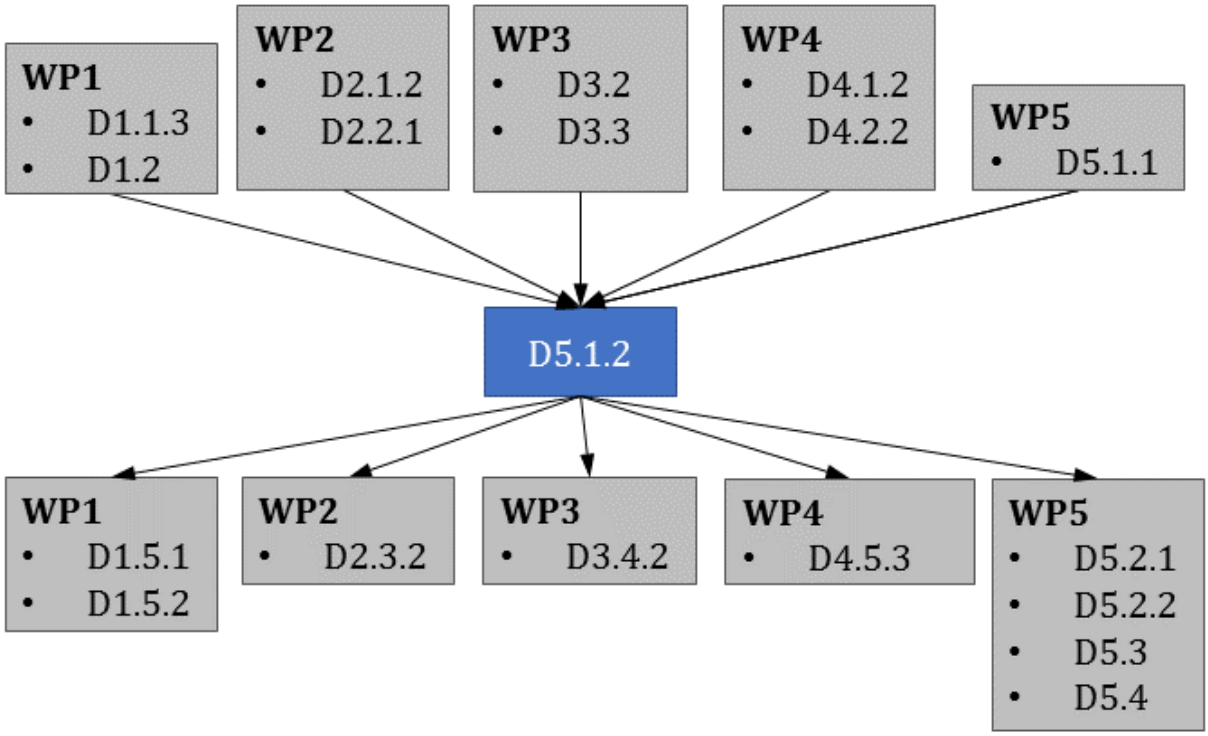


Figure 3. Relationships with other deliverables

More in details, the following deliverables are inputs for D5.1.2:

- D1.1.3 provides the overall architecture of SPECS and defines the greatest part of the Key Concerns used to guide the definition of the VSs. Since the testing approach reported in this deliverable needs Key Concerns, D1.1.3 is needed.
- D1.2 offers a discussion of some usage scenarios, shows the different Interaction Models and states the requirements on the SPECS Platform services. Such scenarios are useful to elicit VSs.
- D2.1.2 provides to this deliverable the set of requirements for the SPECS Negotiation Core services. Since requirements are the most important Key Concern, the document is needed.
- D2.2.1 provides to this deliverable the architecture for the SPECS Negotiation Core services. Since components are used to measure the covered requirements, the document is needed.
- D3.2 provides to this deliverable the set of requirements for the SPECS Monitoring Core services. Since requirements are the most important Key Concern, the document is needed.
- D3.3 provides to this deliverable the architecture for the SPECS Monitoring Core services. Since components are used to measure the covered requirements, the document is needed.
- D4.1.2 provides to this deliverable the set of requirements for the SPECS Enforcement Core services. Since requirements are the most important Key Concern, the document is needed.

- D4.2.2 provides to this deliverable the architecture for the SPECS Enforcement Core services. Since components are used to measure the covered requirements, the document is needed.
- D5.1.1 obviously provides the first version of the testing methodology as well as the first set of the VSs.

The following deliverables use the output of D5.1.2:

- D1.5.1/D1.5.2 will describe the testing approach conducted at integration level. The two deliverables have respectively reported and prototype nature and will be released at M30. The VSs defined in D5.1.2 could be used as the base for defining the proper test cases for the integration testing activities.
- D2.3.2 will report the details of the implementation (and unit testing) of the negotiation module. The VSs defined in D5.1.2 could be used as the base for defining the proper test cases for the unit testing activities.
- D3.4.2 will report the details of the implementation (and unit testing) of the monitoring module. The VSs defined in D5.1.2 could be used as the base for defining the proper test cases for the unit testing activities.
- D4.5.2 will report the details on unit testing of the enforcement module. The VSs defined in D5.1.2 could be used as the base of defining the proper test cases for the unit testing activities.
- D5.2.1, D5.2.2, D5.3 and D5.4 will describe real and industrial applications of SPECS also with the aim of validating the entire approach. These deliverables would benefit from the testing approach described in D5.1.2 as well as the VS set.

3. SPECS validation plan

This section recalls the methodological elements introduced in D5.1.1. The second important aspect is the description of how to implement the VSs using some Validation Applications that could be valuable regarding real world usage: to accomplish this objective, SPECS solution portfolio is used to find such kinds of usage. Before dealing with these topics, a wider approach describing all the testing activities in SPECS, as they are described in the SPECS DoW, are tied together.

3.1. SPECS Testing Levels

The complexity of the SPECS platform, modules and applications requires a structured approach for their validation. Different levels of testing are necessary and distributed among the tasks of the project: without giving full details of such activities that are responsibilities of the concerned tasks, here a bird-eye overview is given.

Three different levels are described:

- *User-Oriented Testing (UOT)* – this level of testing is oriented to the SPECS End-User and the related test cases have been defined from the usage scenarios of SPECS. This level of testing is approached in task T5.1, and it is fully described in D5.1.x deliverables.
- *Integration Testing (IT)* – this level of testing focuses on the interaction between all the SPECS related software components (platform, modules and applications). The test cases are oriented to demonstrate that the APIs provided by each component are correctly invoked by the requesting components. T1.5 focuses on this level of testing showing the results in the related deliverables.
- *Unit Testing (UT)* – this level is devoted to demonstrating that each software component has been built correctly, and its behaviour fulfils the requirements assigned to that component without introducing undesired/dangerous behaviours. Such activities are performed in other tasks in charge of the diverse SPECS modules: negotiation in T2.3; monitoring in T3.4 and enforcement in T4.5.

These levels are not separated: the approaches the relations between them are both present during the testing plan and in the discussion of results.

First, during the test plan phase, the VSs defined in this task (UOT-level) are used as the basis of definition for the scenarios used in IT: during this passage, a different focus is set on the tests passing from a user perspective of the UOT to an interface and API perspective of the IT. Moreover, the scenarios defined at the IT level can be used as the first driver in investigating about functional testing during the UT level.

At the contrary, the results of the testing campaign at UT level can be used as IT level, and those at IT level used at UOT level to maximize the results of the testing campaign keeping the testing effort limited and reusing as much as possible test scripts.

3.2. User-oriented Testing Methodology

This section describes the validation methodology as well as the definition of the approach for the measurement of the quality of the followed testing process.

3.2.1. Domain Model

Figure 4 introduces the concept of *Validation Scenario* as a way to describe an expected behaviour of SPECS. A VS is implemented by one and only one *SPECS validation application*, which is a specialisation of a *SPECS application* for validation purposes. Each SPECS validation

application can be executed on SPECS producing an *execution output*, which is a collection of monitored events. Finally, a SPECS validation application may be included by a SPECS application.

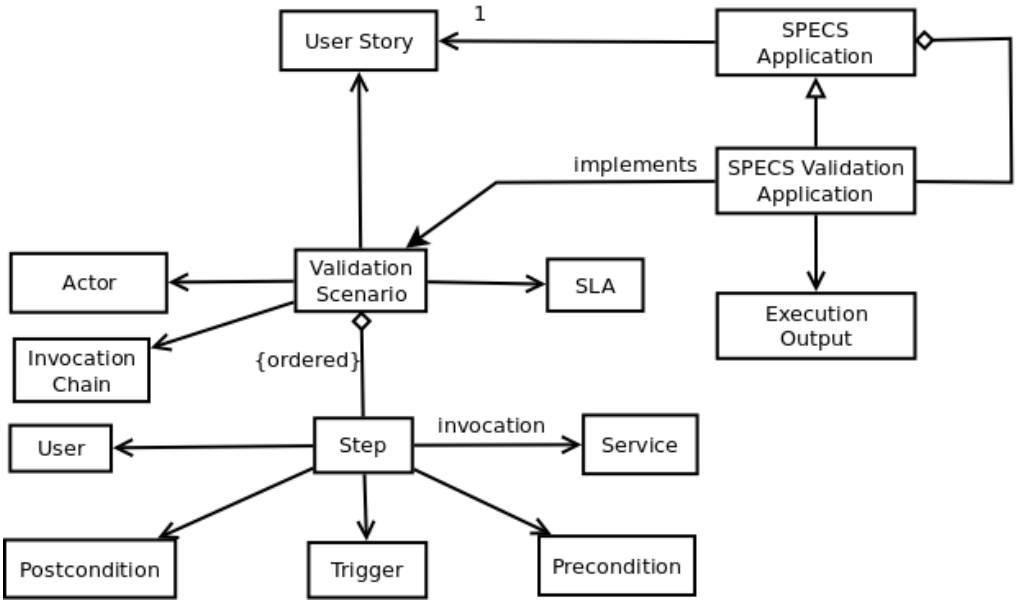


Figure 4. SPECS validation domain model (D5.1.1)

A VS refers to a *User Story* that defines the real world context in which the scenario takes place: a user story is told from the perspective of the cloud users. A validation scenario also refers to an *Invocation Chain* where a specific sequence of interactions among the stakeholders is defined (see D1.2). The third element that constitutes a validation scenario specification is the list of the possible *SLA*s (also regarding security the SLOs) which are dealt with SPECS. The core of a validation scenario specification is represented by the ordered list of *validation step*s, in which the scenario is organized. Each validation step is then described by a list of *preconditions* (i.e., conditions to be satisfied in order to process the step), a *trigger* (i.e., an external event starting the step), an action (i.e., an invocation of a *service*), an *actor* (that may be either an external user or a SPECS component involved in this invocation) and a list of *postconditions* (i.e., conditions to be verified after the execution of the step).

The reader should mind the relation between a SPECS application and the user story. Each user story is implemented by one and only one SPECS application: the latter should contain only SPECS validation applications implementing the validation scenarios related to the user story. This deliverable does not repeat User Stories: see D5.1.1 for further information.

In Y2, we focused on the definition of such SPECS application first by updating the domain model as in Figure 5. In red it is possible to see the two added concepts: the *Solution Portfolio* and the *Solution Portfolio Application.* The Solution Portfolio concept represents a specialisation of the User Story: in fact, among all the possible user stories we captured these that have found industrial application. As the Solution Portfolio represents a specialisation of the User Story concept, the Solution Portfolio Application (SPA) represents a specialisation of

the SPECS Validation Application. Each SPA is hence related to one and only one Solution Portfolio.

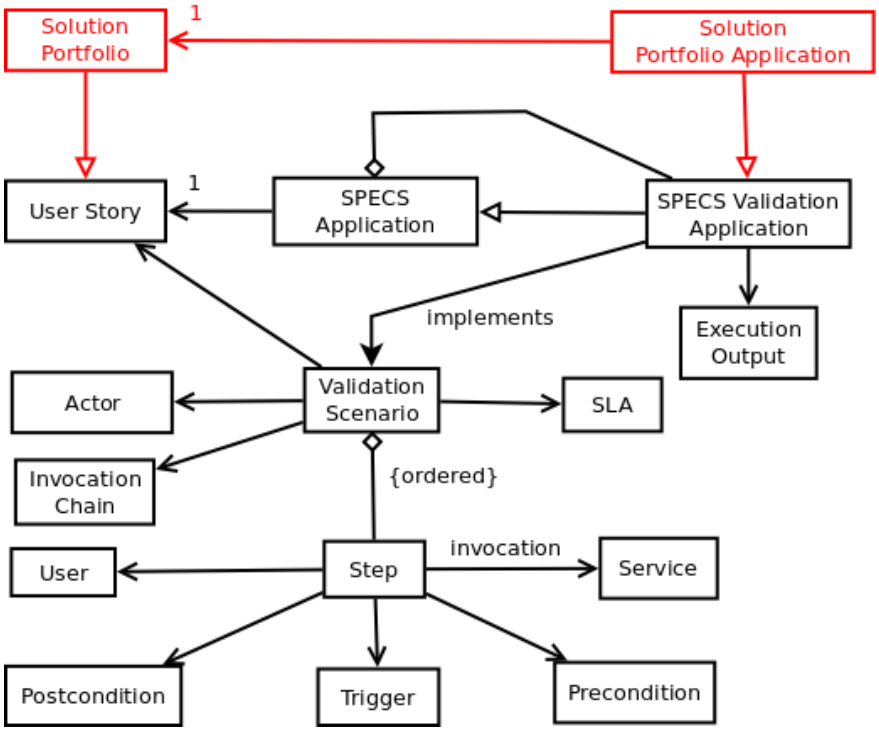


Figure 5. SPECS validation domain model (updated version)

3.2.2. Steps of the User-oriented testing process

Focusing on the UOT level, here we show the sequence of the activities followed to accomplish this task:

- Y1_1. definition of the Key Concern coverage approach to measure the quality of the testing process;
- Y1_2. description of the User Stories;
- Y1_3. elicitation of the VSs from the User Stories and VSs specification using a defined template;
- Y1_4. measurement of the obtained coverage onto the Key Concerns by the VS set;
- Y2_1. refinement of the VSs;
- Y2_2. definition of VAs;
- Y2_3. mapping of these VAs on the VS set;
- Y2_4. refinement of the Key Concerns coverage measurement.

While the activities Y1_x are related to the first year of T5.1, the Y2_x ones span over the second year. While in the remaining part of this section, a recall of the coverage concepts introduced in D5.1.1 are reported, the rest of the document focuses on Y2_x activities. This deliverable focuses on Key Concern Coverage and does not contain any analysis of the results of the testing campaign. This analysis is in D5.1.3, T5.2.2, T5.3 and T5.4,

3.2.3. Coverage

D5.1.1 has defined five key concerns which can be thought as five different dimensions generating all the possible VSs. As depicted in Figure 6, we consider five Key Concerns of the SPECS approach (highlighted in red), namely: *Users, Invocation chains, Target services, SPECS services* and *SLAs*.

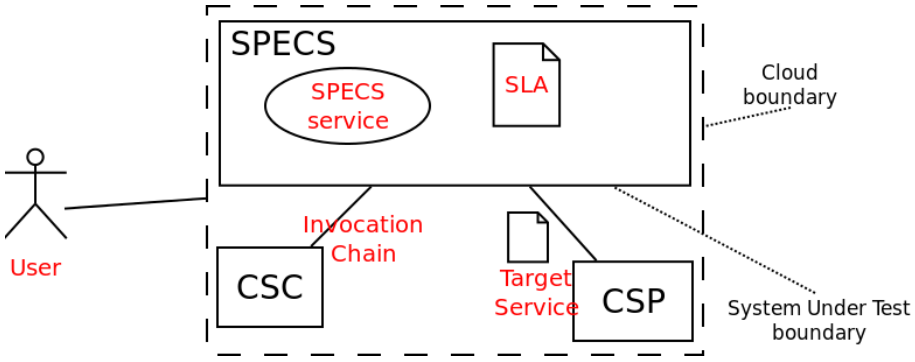


Figure 6. SPECS validation Key Concerns (see D5.1.1)

Such features are used to distinguish between interesting and non-interesting characteristics, leading to the definition of the scenarios. The criterion to distinguish interesting and non-interesting features is the involvement of the EU in that specific aspect. As example, the coverage of all the different techniques protecting the system against a Denial-of-Service attack may be not interesting to a non-expert user. Hence, two VS that cover the same features are considered similar and one of them can be deleted. Table 2 defines the key concern items related to each key concern used to measure the coverage level.

Key concern	Key concern items
Users	Each activity of roles and sub-roles should be involved in at least one VS
Invocation chains	Each Interaction Model / Invocation chain should be covered by at least one VS
Target services	Each kind of Target Service should be referred at least in one VS
SLA lifecycle	Each transition of the state-machine describing the SLA lifecycle state-machine should be covered by at least one VS
SPECS services	Each requirement of the SPECS services should be verified by at least one VS

Table 2. Key Concerns and Key Concern Items

The complete list of the Key Concern Items is available in D5.1.1.

The ultimate goal of every testing campaign is to obtain 100% coverage, this goal is often not reached, due to many technical difficulties and costs-benefits trade-off. Hence, a limited amount of uncovered items could be tolerated if properly justified.

The case of uncovered items could be if, an example, a VS is unfeasible (technically speaking), unreasonable (violating common sense and/or stakeholders aim) or meaningless (not supported by a real world user story). In these cases, uncovering an item could be justified.

3.3. From Validation Scenarios to Validation Applications

To verify the consistency of the expected behaviours at the basis of the specified VSs, they need to be executed using a SPECS Application. One of the main results of the Y2 in T5.1 is the definition of such executable applications we call Validation Applications (VAs). The relationship between VAs and VSs is the same of the relationships between of test scripts and the test cases in traditional software engineering processes. Expressing this last concept regarding a mathematical proportion:

$$VA : VS = \text{test script} : \text{test case}.$$

To define VAs, User Stories have been refined and mapped on specified VSs. The VAs have also been compared to the solution portfolio. There is a major benefit in this comparison: since each application in the solution portfolio is defined by a SPECS partner also by surveying its stakeholders (e.g., customers, business partners, investors), there is a further check on the consistency of the SPECS requirements and the VSs against the stakeholders needs and best practices.

This deliverable defines six VAs: Web Container, Metric Catalogue, Secure Storage, AAA-as-a-Service and Next-Generation Data Center (ngDC) and Security Reasoner. Table 3 reports the mapping of VAs to User Stories and to the solution portfolio.

Validation Applications	**User Stories**	**solution portfolio**
Web Container	Web Container	Secure Web Container
Secure Storage	Secure Storage	End-to-end Encryption
AAA-as-a-Service	Secure Storage	SPECS+ViPR
Metric Catalogue	-	-
ngDC	ngDC	SPECS+ViPR
Security Reasoner	Security Oriented Dashboard	STAR Watch

Table 3. Mapping among VAs, solution portfolio and User Stories

Once the VAs have been defined, each single VA is mapped on the related VSs with the objective to cover as much VSs as possible. Section 5 reports this mapping.

4. SPECS validation scenarios

This section contains the refined version of the VSs published in D5.1.1. The rationale at the basis of the refinement is to add further details and/or to make proper corrections that concern the interaction between the End-user and SPECS. Technological details are sometimes added, but they are not at the centre of this deliverable.

4.1. Secure Storage

4.1.1. Secure_Storage_Selection

General Information			
ID		*SST.1 - Secure_Storage_Selection*	
Version		*2.0*	
User Story		*STO*	*Secure Storage*
Invocation Chain		*IM1-P, IM3*	*Interaction Model 1- SPECS acting the role of Partner*
Scenario Steps			
General Description		*The End-user aims at acquiring a secure storage service from a cloud provider, which fulfils specific security-related requirements. To achieve this service, the End-user negotiates the desired features with SPECS. In this validation scenario, the desired features are entirely implemented by an external CSP, while SPECS only provides the End-user with the functionalities to search, rank and select a service which is compliant to her/his requirements. Moreover, in this scenario, SPECS supports the End-user in signing an SLA with the selected provider.*	
Steps			
1	Phase	*SLA Negotiation*	
	Actor	*End-user, SPECS application, SPECS Negotiation module*	
	Preconditions	*The End-user has a very basic security knowledge, she/he is able to express qualitatively requirements at a high-level of abstraction.*	
	Trigger		
	Actions	*The End-user accesses the SPECS application interface. The negotiation request is forwarded to the SPECS Negotiation module, which retrieves the list of available SLA templates representing the available security services and the related security capabilities, controls and metrics. The services are returned to the End-user.*	
	Postconditions		
2	Phase	*SLA Negotiation*	
	Actor	*End-user, SPECS application*	
	Preconditions		
	Trigger		
	Actions	*The End-user selects, among the available service offers, the desired one, i.e., the Database and Backup. The End-user specifies the desired security features by selecting the capabilities she/he is interested in and specifying the related security controls, and by specifying the desired metrics and setting related SLOs.*	
	Postconditions	*A supply chain compliant to the End-user requirements is built.*	
3	Phase	*SLA Negotiation*	
	Actor	*SPECS application, SPECS Negotiation module, SPECS Enforcement module*	
	Preconditions	*A secure storage service which fulfils the specific security requirements is known to SPECS.*	
	Trigger		

	Actions	*The End-user's choices are forwarded by the SPECS application to the SPECS Negotiation module, which searches for valid supply chains. In particular, the list of supply chains is built with the help of the SPECS Enforcement module.* *For each valid supply chain, a SLA Offer is created. The set of SLA Offers are hence ranked and returned to the SPECS application. The CSPs also add the cost of each service offer.*
	Postconditions	
4	Phase	*SLA Negotiation*
	Actor	*End-user, SPECS application, SLA Platform*
	Preconditions	*The End-user shall be logged on SPECS.*
	Trigger	
	Actions	*The SPECS application validates the SLA Offers which are then presented to the End-user. The service offer is associated with an SLA published by an external CSP.* *The End-user either:* *1. accepts and signs the SLA offered by the external CSP;* *2. does not select any SLA Offer from the list and repeats the whole process from step 1 (possibly specifying a different set of requirements);* *3. does not select any SLA Offer from the list and exits the application.*
	Postconditions	*In case 1 - the signed SLA is stored by SPECS. The End-user is enabled to invoke the desired service on the external CSP with the configuration information included in the SLA.*
Graphical Model		*Not reported to avoid replication of information. See D1.3 for detailed interactions between SPECS modules.*
Coverage Information		
Users	*U_1 (CSC:User)*	
Target services	*TS_3 (Data Storage as a Service)*	
SPECS services	*See Appendix B*	
SLA	*SLA_1, SLA_3. SLA_4, SLA_5*	

4.1.2. Secure_Storage_Brokering_with_Client_Crypto

General Information		
ID	*SST.2 - Secure_Storage_Brokering_with_Client_Crypto*	
Version	*2.0*	
User Story	*STO*	*Secure Storage*
Invocation Chain	*IM1-CSP, IM3*	*Interaction Model 1- SPECS acting the role of CSP*
Scenario Steps		
General Description	*The End-user aims at acquiring a secure storage service from a remote cloud provider, which fulfils specific security-related requirements. Specifically, the End-user needs the two capabilities of Database-as-a-Service and End-2-End Encryption in order to detect and prove security-related violations and to locally encrypt her/his data.* *To achieve this service, the End-user negotiates the desired features with SPECS and signs an SLA including all service terms and guarantees.* *SPECS acquires the Database-as-a-Service on behalf of the End-user (registered on SPECS) and provides her/him with end-2-end encryption security mechanism. In this scenario, SPECS also provides monitoring functionalities.*	
Steps		

1	Phase	*SLA Negotiation*
	Actor	*End-user, SPECS application, SPECS Negotiation module*
	Preconditions	*The End-user has a very basic security knowledge, she/he is able to express qualitatively requirements at a high-level of abstraction.*
	Trigger	
	Actions	*The End-user accesses the SPECS application interface. The negotiation request is forwarded to the SPECS Negotiation module, which retrieves the list of available SLA templates representing the available security services and the related security capabilities, controls and metrics. The services are returned to the End-user.*
	Postconditions	
2	Phase	*SLA Negotiation*
	Actor	*End-user, SPECS application*
	Preconditions	
	Trigger	
	Actions	*The End-user selects, among the available service offers, the desired one, i.e., the Database and Backup with End-2-End Encryption. The End-user specifies the desired security features by selecting the capabilities she/he is interested in and specifying the related security controls, and by specifying the desired metrics and setting related SLOs. Precisely, the End-user specifies, between others, the need of having a client-side encryption mechanism.*
	Postconditions	*A supply chain compliant to the End-user requirements is built.*
3	Phase	*SLA Negotiation*
	Actor	*SPECS application, SPECS Negotiation module, SPECS Enforcement module*
	Preconditions	*A secure storage service which fulfils the specific security requirements is not known to SPECS.* *An external CSP offering the Database-as-a-Service compliant with the related End-user's requirements is known to SPECS, and the end-2-end encryption is offered as SPECS security mechanism.*
	Trigger	
	Actions	*The End-user's choices are forwarded by the SPECS application to the SPECS Negotiation module, which searches for valid supply chains. In particular, the list of supply chains is built with the help of the SPECS Enforcement module. In this step, an external CSP offering the Database-as-a-Service is identified while the Encryption Package, able to support the client-side encryption, is added as a SPECS Enforcement service.* *For each valid supply chain, a SLA Offer is created. The set of SLA Offers are hence ranked and returned to the SPECS application.*
	Postconditions	
4	Phase	*SLA Negotiation*
	Actor	*End-user, SPECS application, SLA Platform*
	Preconditions	*The End-user shall be logged on SPECS.*
	Trigger	
	Actions	*The SPECS application validates the SLA Offers which are then presented to the End-user. The End-user selects the SLA Offer in which the Database-as-a-Service is offered by an external CSP while the client-side encryption is offered as a SPECS security mechanism. The selected SLA Offer is used to update and sign the SLA in the SLA Platform*
	Postconditions	*The SLA, containing all information needed for SLA implementation, has been signed.*
5	Phase	*SLA Implementation*
	Actor	*SPECS application, SPECS Enforcement module, SLA Platform*

	Preconditions	*A valid signed SLA containing all service terms and service guarantees is available in the SLA Platform*
	Trigger	
	Actions	*The SPECS application invokes the SPECS Enforcement module which retrieves the SLA to implement from the SLA Platform and prepares a plan to implement the signed SLA: it analyses the SLA, deduces alert thresholds, chooses the security and monitoring mechanisms to activate and determines all related software to install and their configurations.*
	Postconditions	
6	Phase	*SLA Implementation*
	Actor	*SPECS Enforcement module*
	Preconditions	*A plan has been built to implement a signed SLA.*
	Trigger	
	Actions	*The SPECS Enforcement module implements the plan, by configuring and deploying all the components in order to respect the features granted in the SLA. The SPECS Enforcement module deploys and configures monitoring agents and activates all the components and services.*
	Postconditions	
7	Phase	*SLA Implementation*
	Actor	*SPECS Enforcement module, SPECS Monitoring Module*
	Preconditions	*All components and services needed for SLA implementation have been correctly configured and activated.*
	Trigger	
	Actions	*The SPECS Enforcement module configures the Monitoring module with a monitoring policy by setting proper alert/violation thresholds for specific metrics.*
	Postconditions	
8	Phase	*SLA Monitoring*
	Actor	*SPECS Monitoring module*
	Preconditions	
	Trigger	
	Actions	*SPECS keeps collecting information about the provided service and evaluates them against the current monitoring policy.*
	Postconditions	
Graphical Model		*Not reported to avoid replication of information. See D1.3 for detailed interactions between SPECS modules.*
Coverage Information		
Users	*U_1 (CSC:User)*	
Target services	*TS_3 (Data Storage as a Service), TS_7 (Software as a Service)*	
SPECS services	*See Appendix B*	
SLA	*SLA_1, SLA_3, SLA_6, SLA_7*	

4.1.3. Secure_Storage_with_Defined_CSP

General Information		
ID	*SST.3 - Secure_Storage_with_Defined_CSP*	
Version	*2.0*	
User Story	*STO*	*Secure Storage*

Invocation Chain		*IM1-CSP, IM3*	*Interaction Model 1- SPECS acting the role of CSP*
Scenario Steps			
General Description		*The End-user aims at storing encrypted data on a known remote cloud provider which offers a Database-as-a-service. The End-user asks SPECS for End-2-End Encryption capability, needed to locally encrypt her/his data.* *To achieve this service, the End-user also gives SPECS her/his credentials on the chosen provider; SPECS manages these credentials and uses them to log into the chosen provider and store User's data.* *In this scenario, SPECS also provides monitoring functionalities.*	
Steps			
1	Phase	*SLA Negotiation*	
	Actor	*End-user, SPECS application, SPECS Negotiation module*	
	Preconditions	*The End-user has a very basic security knowledge, she/he is able to express qualitatively requirements at a high-level of abstraction.*	
	Trigger		
	Actions	*The End-user accesses the SPECS application interface. The negotiation request is forwarded to the SPECS Negotiation module, which retrieves the list of available SLA templates representing the available security services and the related security capabilities, controls and metrics. The services are returned to the End-user.*	
	Postconditions		
2	Phase	*SLA Negotiation*	
	Actor	*End-user, SPECS application*	
	Preconditions	*The external CSP offering the Database-as-a-Service chosen by the End-user is known to SPECS, and the end-2-end encryption is offered as SPECS security mechanism.*	
	Trigger		
	Actions	*The End-user selects, among the available service offers, the desired one, i.e., the Database and Backup with End-2-End Encryption. The End-user specifies the desired security features by selecting the capabilities she/he is interested in and specifying the related security controls, and by specifying the desired metrics and setting related SLOs. Precisely, the End-user specifies, between others, the needs of using a specific CSP as Database-as-a-Service provider and having a client-side encryption mechanism.*	
	Postconditions	*A supply chain compliant to the End-user requirements is built.*	
3	Phase	*SLA Negotiation*	
	Actor	*SPECS application, SPECS Negotiation module, SPECS Enforcement module*	
	Preconditions		
	Trigger		
	Actions	*The End-user's choices are forwarded by the SPECS application to the SPECS Negotiation module, which searches for valid supply chains. In particular, the list of supply chains is built with the help of the SPECS Enforcement module. In this step, the specific CSP defined by the End-user is identified while the Encryption Package, able to support the client-side encryption, is added as a SPECS Enforcement service.* *For each valid supply chain, a SLA Offer is created. The set of SLA Offers are hence ranked and returned to the SPECS application.*	
	Postconditions		
4	Phase	*SLA Negotiation*	
	Actor	*End-user, SPECS application, SLA Platform*	
	Preconditions	*The End-user shall be logged on SPECS.*	

	Trigger	
	Actions	*The SPECS application validates the SLA Offers which are then presented to the End-user. The End-user selects the SLA Offer in which the Database-as-a-Service is offered by an external CSP while the client-side encryption is offered as a SPECS security mechanism. The selected SLA Offer is used to update and sign the SLA in the SLA Platform*
	Postconditions	*The SLA, containing all information needed for SLA implementation, has been signed.*
5	Phase	*SLA Implementation*
	Actor	*SPECS application, SPECS Enforcement module, SLA Platform*
	Preconditions	*A valid signed SLA containing all service terms and service guarantees is available in the SLA Platform.*
	Trigger	
	Actions	*The SPECS application invokes the SPECS Enforcement module which retrieves the SLA to implement from the SLA Platform and prepares a plan to implement the signed SLA: it analyses the SLA, deduces alert thresholds, chooses the security and monitoring mechanisms to activate and determines all related software to install and their configurations.*
6	Phase	*SLA Implementation*
	Actor	*SPECS Enforcement module*
	Preconditions	*A plan has been built to implement a signed SLA.* *The credentials of the End-user on the external CSP have been acquired.*
	Trigger	
	Actions	*The SPECS Enforcement module implements the plan, by configuring and deploying all the components in order to respect the features granted in the SLA. The SPECS Enforcement module acquires the storage service with the credentials of the End-user on the external CSP and deploys and configures monitoring agents. The SPECS Enforcement module activates all the components and services.*
	Postconditions	
7	Phase	*SLA Implementation*
	Actor	*SPECS Enforcement module, SPECS Monitoring Module*
	Preconditions	*All components and services needed for SLA implementation have been correctly configured and activated.*
	Trigger	
	Actions	*The SPECS Enforcement module configures the Monitoring module with a monitoring policy by setting proper alert/violation thresholds for specific metrics.*
	Postconditions	
8	Phase	*SLA Monitoring*
	Actor	*SPECS Monitoring module*
	Preconditions	
	Trigger	
	Actions	*SPECS keeps collecting information about the provided service and evaluates them against the current monitoring policy.*
	Postconditions	
Graphical Model		*Not reported to avoid replication of information. See D1.3 for detailed interactions between SPECS modules.*
Coverage Information		
Users	*U_1 (CSC:User)*	
Target services	*TS_3 (Data Storage as a Service), TS_7 (Software as a Service)*	

SPECS services	*See Appendix B*
SLA	*SLA_1, SLA_3, SLA_6, SLA_7*

4.1.4. Secure_Storage_Brokering_with_Client_Crypto_alert

General Information			
ID		*SST.4 - Secure_Storage_Brokering_with_Client_Crypto_alert*	
Version		*2.0*	
User Story		*STO*	*Secure Storage*
Invocation Chain		*IM1-CSP, IM3*	*Interaction Model 1- SPECS acting the role of CSP*
Scenario Steps			
General Description		*The End-user aims at acquiring a secure storage service from a remote cloud provider, which fulfils specific security-related requirements. Specifically, the End-user needs the two capabilities of Database-as-a-Service and End-2-End Encryption in order to detect and prove security-related violations and to locally encrypt her/his data.* *To achieve this service, the End-user negotiates the desired features with SPECS and signs an SLA including all service terms and guarantees.* *SPECS acquires the Database-as-a-Service on behalf of the End-user (registered on SPECS) and provides her/him with end-2-end encryption security mechanism. In this scenario, SPECS also provides monitoring functionalities.* *In this scenario, an alert is raised since the Encryption Server component is detected to be down and, since no data are sent from the End-user during the down time, no violation occurs.*	
Steps			
1	Phase	*SLA Negotiation*	
	Actor	*End-user, SPECS application, SPECS Negotiation module*	
	Preconditions	*The End-user has a very basic security knowledge, she/he is able to express qualitatively requirements at a high-level of abstraction.*	
	Trigger		
	Actions	*The End-user accesses the SPECS application interface. The negotiation request is forwarded to the SPECS Negotiation module, which retrieves the list of available SLA templates representing the available security services and the related security capabilities, controls and metrics. The services are returned to the End-user.*	
	Postconditions		
2	Phase	*SLA Negotiation*	
	Actor	*End-user, SPECS application*	
	Preconditions		
	Trigger		
	Actions	*The End-user selects, among the available service offers, the desired one, i.e., the Database and Backup with End-2-End Encryption. The End-user specifies the desired security features by selecting the capabilities she/he is interested in and specifying the related security controls, and by specifying the desired metrics and setting related SLOs. Precisely, the End-user specifies, between others, the need of having a client-side encryption mechanism.*	
	Postconditions	*A supply chain compliant to the End-user requirements is built.*	
3	Phase	*SLA Negotiation*	
	Actor	*SPECS application, SPECS Negotiation module, SPECS Enforcement module*	

	Preconditions	*A secure storage service which fulfils the specific security requirements is not known to SPECS.* *An external CSP offering the Database-as-a-Service compliant with the related End-user's requirements is known to SPECS, and the end-2-end encryption is offered as SPECS security mechanism.*
	Trigger	
	Actions	*The End-user's choices are forwarded by the SPECS application to the SPECS Negotiation module, which searches for valid supply chains. In particular, the list of supply chains is built with the help of the SPECS Enforcement module. In this step, an external CSP offering the Database-as-a-Service is identified while the Encryption Package, able to support the client-side encryption, is added as a SPECS Enforcement service.* *For each valid supply chain, a SLA Offer is created. The set of SLA Offers are hence ranked and returned to the SPECS application.*
	Postconditions	
4	Phase	*SLA Negotiation*
	Actor	*End-user, SPECS application, SLA Platform*
	Preconditions	*The End-user shall be logged on SPECS.*
	Trigger	
	Actions	*The SPECS application validates the SLA Offers which are then presented to the End-user. The End-user selects the SLA Offer in which the Database-as-a-Service is offered by an external CSP while the client-side encryption is offered as a SPECS security mechanism. The selected SLA Offer is used to update and sign the SLA in the SLA Platform*
	Postconditions	*The SLA, containing all information needed for SLA implementation, has been signed.*
5	Phase	*SLA Implementation*
	Actor	*SPECS application, SPECS Enforcement module, SLA Platform*
	Preconditions	*A valid signed SLA containing all service terms and service guarantees is available in the SLA Platform*
	Trigger	
	Actions	*The SPECS application invokes the SPECS Enforcement module which retrieves the SLA to implement from the SLA Platform and prepares a plan to implement the signed SLA: it analyses the SLA, deduces alert thresholds, chooses the security and monitoring mechanisms to activate and determines all related software to install and their configurations.*
	Postconditions	
6	Phase	*SLA Implementation*
	Actor	*SPECS Enforcement module*
	Preconditions	*A plan has been built to implement a signed SLA.*
	Trigger	
	Actions	*The SPECS Enforcement module implements the plan, by configuring and deploying all the components in order to respect the features granted in the SLA. The SPECS Enforcement module deploys and configures monitoring agents and activates all the components and services.*
	Postconditions	
7	Phase	*SLA Implementation*
	Actor	*SPECS Enforcement module, SPECS Monitoring Module*
	Preconditions	*All components and services needed for SLA implementation have been correctly configured and activated.*
	Trigger	
	Actions	*The SPECS Enforcement module configures the Monitoring module with a monitoring policy by setting proper alert/violation thresholds for specific metrics.*

	Postconditions	
8	Phase	*SLA Monitoring*
	Actor	*SPECS Monitoring module*
	Preconditions	
	Trigger	
	Actions	*SPECS keeps collecting information about the provided service and evaluates them against the current monitoring policy.*
	Postconditions	
9	Phase	*SLA Remediation*
	Actor	*SPECS Monitoring module, SPECS Enforcement module*
	Preconditions	
	Trigger	*The SPECS Monitoring module generates monitoring events due to the deviation of some metrics from set thresholds (since the the Encryption Server component is down).*
	Actions	*The SPECS Enforcement module analyses monitoring events and classifies it as an alert. The root cause of the monitoring event is determined (the Encryption server component is detected to be down, but no data has been sent from the End-user during the down time; thus no violation occurs).*
	Postconditions	*A report on the alert and on the root cause of the monitoring event is created.*
10	Phase	*SLA Remediation*
	Actor	*SPECS Enforcement module*
	Preconditions	
	Trigger	
	Actions	*The SPECS Enforcement module reacts by restarting the component before any encrypted files are sent to the server.*
	Postconditions	*The alert is solved.*
Graphical Model		*Not reported to avoid replication of information. See D1.3 for detailed interactions between SPECS modules.*
Coverage Information		
Users	*U_1 (CSC:User)*	
Target services	*TS_3 (Data Storage as a Service), TS_7 (Software as a Service)*	
SPECS services	*See Appendix B*	
SLA	*SLA_1, SLA_3, SLA_6, SLA_7, SLA_9, SLA_10, SLA_11*	

4.1.5. Secure_Storage_Brokering_with_Client_Crypto_violation

General Information		
ID	*SST.5 - Secure_Storage_Brokering_with_Client_Crypto_violation*	
Version	*2.0*	
User Story	*STO*	*Secure Storage*
Invocation Chain	*IM1-CSP, IM3*	*Interaction Model 1- SPECS acting the role of CSP*
Scenario Steps		

General Description		*The End-user aims at acquiring a secure storage service from a remote cloud provider, which fulfils specific security-related requirements. Specifically, the End-user needs the two capabilities of Database-as-a-Service and End-2-End Encryption in order to detect and prove security-related violations and to locally encrypt her/his data.* *To achieve this service, the End-user negotiates the desired features with SPECS and signs an SLA including all service terms and guarantees.* *SPECS acquires the Database-as-a-Service on behalf of the End-user (registered on SPECS) and provides her/him with end-2-end encryption security mechanism. In this scenario, SPECS also provides monitoring functionalities.* *In this scenario, a violation is detected since the Encryption Server component is detected to be down.*
Steps		
1	Phase	*SLA Negotiation*
	Actor	*End-user, SPECS application, SPECS Negotiation module*
	Preconditions	*The End-user has a very basic security knowledge, she/he is able to express qualitatively requirements at a high-level of abstraction.*
	Trigger	
	Actions	*The End-user accesses the SPECS application interface. The negotiation request is forwarded to the SPECS Negotiation module, which retrieves the list of available SLA templates representing the available security services and the related security capabilities, controls and metrics. The services are returned to the End-user.*
	Postconditions	
2	Phase	*SLA Negotiation*
	Actor	*End-user, SPECS application*
	Preconditions	
	Trigger	
	Actions	*The End-user selects, among the available service offers, the desired one, i.e., the Database and Backup with End-2-End Encryption. The End-user specifies the desired security features by selecting the capabilities she/he is interested in and specifying the related security controls, and by specifying the desired metrics and setting related SLOs. Precisely, the End-user specifies, between others, the need of having a client-side encryption mechanism.*
	Postconditions	*A supply chain compliant to the End-user requirements is built.*
3	Phase	*SLA Negotiation*
	Actor	*SPECS application, SPECS Negotiation module, SPECS Enforcement module*
	Preconditions	*A secure storage service which fulfils the specific security requirements is not known to SPECS.* *An external CSP offering the Database-as-a-Service compliant with the related End-user's requirements is known to SPECS, and the end-2-end encryption is offered as SPECS security mechanism.*
	Trigger	
	Actions	*The End-user's choices are forwarded by the SPECS application to the SPECS Negotiation module, which searches for valid supply chains. In particular, the list of supply chains is built with the help of the SPECS Enforcement module. In this step, an external CSP offering the Database-as-a-Service is identified while the Encryption Package, able to support the client-side encryption, is added as a SPECS Enforcement service.* *For each valid supply chain, a SLA Offer is created. The set of SLA Offers are hence ranked and returned to the SPECS application.*
	Postconditions	
4	Phase	*SLA Negotiation*
	Actor	*End-user, SPECS application, SLA Platform*

	Preconditions	*The End-user shall be logged on SPECS.*
	Trigger	
	Actions	*The SPECS application validates the SLA Offers which are then presented to the End-user. The End-user selects the SLA Offer in which the Database-as-a-Service is offered by an external CSP while the client-side encryption is offered as a SPECS security mechanism. The selected SLA Offer is used to update and sign the SLA in the SLA Platform*
	Postconditions	*The SLA, containing all information needed for SLA implementation, has been signed.*
5	Phase	*SLA Implementation*
	Actor	*SPECS application, SPECS Enforcement module, SLA Platform*
	Preconditions	*A valid signed SLA containing all service terms and service guarantees is available in the SLA Platform*
	Trigger	
	Actions	*The SPECS application invokes the SPECS Enforcement module which retrieves the SLA to implement from the SLA Platform and prepares a plan to implement the signed SLA: it analyses the SLA, deduces alert thresholds, chooses the security and monitoring mechanisms to activate and determines all related software to install and their configurations.*
	Postconditions	
6	Phase	*SLA Implementation*
	Actor	*SPECS Enforcement module*
	Preconditions	*A plan has been built to implement a signed SLA.*
	Trigger	
	Actions	*The SPECS Enforcement module implements the plan, by configuring and deploying all the components in order to respect the features granted in the SLA. The SPECS Enforcement module deploys and configures monitoring agents and activates all the components and services.*
	Postconditions	
7	Phase	*SLA Implementation*
	Actor	*SPECS Enforcement module, SPECS Monitoring Module*
	Preconditions	*All components and services needed for SLA implementation have been correctly configured and activated.*
	Trigger	
	Actions	*The SPECS Enforcement module configures the Monitoring module with a monitoring policy by setting proper alert/violation thresholds for specific metrics.*
	Postconditions	
8	Phase	*SLA Monitoring*
	Actor	*SPECS Monitoring module*
	Preconditions	
	Trigger	
	Actions	*SPECS keeps collecting information about the provided service and evaluates them against the current monitoring policy.*
	Postconditions	
9	Phase	*SLA Remediation*
	Actor	*End-user, SPECS Monitoring module, SPECS Enforcement module*
	Preconditions	*The End-user has sent files to encrypt to the server while it is down*
	Trigger	*The SPECS Monitoring module generates monitoring events due to the deviation of some metrics from set thresholds (since the Encryption Server component is down).*

	Actions	*The SPECS Enforcement module analyses monitoring events and detects a violation. The root cause analysis of the monitoring event is determined (the Encryption Server component is detected to be down).*
	Postconditions	*A report on the violation and on the root cause of the monitoring event is created.*
10	Phase	*SLA Remediation*
	Actor	*SPECS Enforcement module*
	Preconditions	
	Trigger	
	Actions	*SPECS notifies the violation to the End-User through the SPECS Application. The SPECS Enforcement module searches for alternatives for the End-user by building new services.*
	Postconditions	*The SLA is no more fulfilled.*
Graphical Model		*Not reported to avoid replication of information. See D1.3 for detailed interactions between SPECS modules.*
Coverage Information		
Users	*U_1 (CSC:User)*	
Target services	*TS_3 (Data Storage as a Service), TS_7 (Software as a Service)*	
SPECS services	*See Appendix B*	
SLA	*SLA_1, SLA_3, SLA_6, SLA_7, SLA_9, SLA_12*	

4.2. Secure Web Container

4.2.1. Secure_Web_Container_Selection

General Information		
ID	*SWC.1 - Secure_Web_Container_Selection*	
Version	*2.0*	
User Story	*WEB*	*Secure Web Container*
Invocation Chain	*IM1-P*	*Interaction Model 1- SPECS acting the role of Partner*
Scenario Steps		
General Description	*The End-user aims at acquiring a web container from an Infrastructure-as-a-Service CSP, represented by a VM hosting the Web Server, which fulfils specific security requirements. To achieve this service, the End-User negotiates the desired features with SPECS.* *In this validation scenario, the desired features are entirely implemented by an Infrastructure-as-a-Service CSP. SPECS only returns to the End-user the reference to such provider.*	
Steps		
1	Phase	*SLA Negotiation*
	Actor	*End-user, SPECS application, SPECS Negotiation module*
	Preconditions	*The End-user is an expert customer since she/he is able to evaluate each individual metric with respect to her/him own security requirements.*
	Trigger	
	Actions	*The End-user accesses the SPECS application interface using the expert interface, in order to enter/specify in a specific way her/his security requirements. The negotiation request is forwarded to the SPECS Negotiation module, which retrieves the list of available SLA templates representing the available security services and the related security capabilities, controls and metrics. The services are returned to the End-user.*
	Postconditions	

2	Phase	*SLA Negotiation*
	Actor	*End-user, SPECS application*
	Preconditions	
	Trigger	
	Actions	*The End-user selects, among the available service offers, the desired one, i.e., the Secure Web Container. The End-user specifies the desired security features by selecting the capabilities she/he is interested in and specifying the related security controls, and by specifying the desired metrics and setting related SLOs.*
	Postconditions	*A supply chain compliant to the End-user requirements is built.*
3	Phase	*SLA Negotiation*
	Actor	*SPECS application, SPECS Negotiation module, SPECS Enforcement module*
	Preconditions	*A web container, which fulfils the specific security requirements, is offered by at least one external CSP, known to SPECS.*
	Trigger	
	Actions	*The End-user's choices are forwarded by the SPECS application to the SPECS Negotiation module, which searches for valid supply chains. In particular, the list of supply chains is built with the help of the SPECS Enforcement module. In this step, an external CSP offering the Secure Web Container is identified.* *For each valid supply chain, a SLA Offer is created. The set of SLA Offers are hence ranked and returned to the SPECS application. The CSPs also add the cost of each service offer.*
	Postconditions	
4	Phase	*SLA Negotiation*
	Actor	*End-user, SPECS application, SLA Platform*
	Preconditions	*The End-user shall be logged on SPECS.*
	Trigger	
	Actions	*The SPECS application validates the SLA Offers which are then presented to the End-user. The service offer is associated with an SLA published by an external CSP.* *The End-user either:* *1. accepts and signs the SLA offered by the external CSP;* *2. does not select any SLA Offer from the list and repeats the whole process from step 1 (possibly specifying a different set of requirements);* *3. does not select any SLA Offer from the list and exits the application.*
	Postconditions	*In case 1 - the signed SLA is stored by SPECS. The End-user is enabled to invoke the desired service on the external CSP with the configuration information included in the SLA.*
Graphical Model		*Not reported to avoid replication of information. See D1.3 for detailed interactions between SPECS modules.*
Coverage Information		
Users	*U_1 (CSC:User)*	
Target services	*TS_4 (Infrastructure as a Service)*	
SPECS services	*See Appendix B*	
SLA	*SLA_1, SLA_3. SLA_4, SLA_5*	

4.2.2. Secure_Web_Container_Brokering

General Information	
ID	*SWC.2 - Secure_Web_Container_Brokering*
Version	*2.0*

User Story		*WEB*	*Secure Web Container*
Invocation Chain		*IM1-CSP*	*Interaction Model 1- SPECS acting the role of CSP*
Scenario Steps			
General Description		*The End-user aims at acquiring a web container from an Infrastructure-as-a-Service CSP, represented by a VM hosting the Web Server, which fulfils specific security-related requirements. To achieve this service, the End-User negotiates the desired security features with SPECS.* *In this validation scenario, the desired features are entirely implemented by an Infrastructure-as-a-Service CSP. SPECS acquires the resources on behalf of the End-user (registered on SPECS) and sets up some monitoring functionalities in order to monitor the SLA achievement.*	
Steps			
1	Phase	*SLA Negotiation*	
	Actor	*End-user, SPECS application, SPECS Negotiation module*	
	Preconditions	*The End-user has a very basic security knowledge, she/he is able to express qualitatively requirements at a high-level of abstraction.*	
	Trigger		
	Actions	*The End-user accesses the SPECS application interface. The negotiation request is forwarded to the SPECS Negotiation module, which retrieves the list of available SLA templates representing the available security services and the related security capabilities, controls and metrics. The services are returned to the End-user.*	
	Postconditions		
2	Phase	*SLA Negotiation*	
	Actor	*End-user, SPECS application*	
	Preconditions		
	Trigger		
	Actions	*The End-user selects, among the available service offers, the desired one, i.e., the Secure Web Container. The End-user specifies the desired security features by selecting the capabilities she/he is interested in and specifying the related security controls, and by specifying the desired metrics and setting related SLOs.* *The End-user accesses the Security Metric Catalogue in order to have additional and detailed information about the specific chosen metrics.*	
	Postconditions	*A supply chain compliant to the End-user requirements is built.*	
3	Phase	*SLA Negotiation*	
	Actor	*SPECS application, SPECS Negotiation module, SPECS Enforcement module*	
	Preconditions	*A web container, which fulfils the specific security requirements, is offered by at least one external CSP, known to SPECS.*	
	Trigger		
	Actions	*The End-user's choices are forwarded by the SPECS application to the SPECS Negotiation module, which searches for valid supply chains. In particular, the list of supply chains is built with the help of the SPECS Enforcement module. In this step, an external CSP offering the Secure Web Container is identified.* *For each valid supply chain, a SLA Offer is created. The set of SLA Offers are hence ranked and returned to the SPECS application.*	
	Postconditions		
4	Phase	*SLA Negotiation*	
	Actor	*End-user, SPECS application, SLA Platform*	
	Preconditions	*The End-user shall be logged on SPECS.*	
	Trigger		

	Actions	*The SPECS application validates the SLA Offers which are then presented to the End-user. The End-user selects the SLA Offer in which the Secure Web Container is offered by an external CSP. The selected SLA Offer is used to update and sign the SLA in the SLA Platform*
	Postconditions	*The SLA, containing all information needed for SLA implementation, has been signed.*
5	Phase	*SLA Implementation*
	Actor	*SPECS application, SPECS Enforcement module, SLA Platform*
	Preconditions	*A valid signed SLA containing all service terms and service guarantees is available in the SLA Platform*
	Trigger	
	Actions	*The SPECS application invokes the SPECS Enforcement module which retrieves the SLA to implement from the SLA Platform and prepares a plan to implement the signed SLA: it analyses the SLA, deduces alert thresholds, chooses the security and monitoring mechanisms to activate and determines all related software to install and their configurations.*
	Postconditions	
6	Phase	*SLA Implementation*
	Actor	*SPECS Enforcement module*
	Preconditions	*A plan has been built to implement a signed SLA.*
	Trigger	
	Actions	*The SPECS Enforcement module implements the plan, by configuring and deploying all the components in order to respect the features granted in the SLA. The SPECS Enforcement module deploys and configures monitoring agents and activates all the components and services.*
	Postconditions	
7	Phase	*SLA Implementation*
	Actor	*SPECS Enforcement module, SPECS Monitoring Module*
	Preconditions	*All components and services needed for SLA implementation have been correctly configured and activated.*
	Trigger	
	Actions	*The SPECS Enforcement module configures the Monitoring module with a monitoring policy by setting proper alert/violation thresholds for specific metrics.*
	Postconditions	
8	Phase	*SLA Monitoring*
	Actor	*SPECS Monitoring module*
	Preconditions	
	Trigger	
	Actions	*SPECS keeps collecting information about the provided service and evaluates them against the current monitoring policy.*
	Postconditions	
Graphical Model		*Not reported to avoid replication of information. See D1.3 for detailed interactions between SPECS modules.*
Coverage Information		
Users	*U_1 (CSC:User)*	
Target services	*TS_4 (Infrastructure as a Service)*	
SPECS services	*See Appendix B*	
SLA	*SLA_1, SLA_3, SLA_6, SLA_7*	

4.2.3. Secure_Web_Container_TLS_enhanced

General Information			
ID	*SWC.3 - Secure_Web_Container_TLS_enhanced*		
Version	*2.0*		
User Story	*WEB*	*Secure Web Container*	
Invocation Chain	*IM1-CSP*	*Interaction Model 1- SPECS acting the role of CSP*	
Scenario Steps			
General Description	*The End-user aims at acquiring a web container from an Infrastructure-as-a-Service CSP, represented by a VM hosting the Web Server, which fulfils specific security-related requirements. In particular, the End-user requires the adoption of Transport Layer Security (TLS) protocol to protect the Web Server communications, DoS detection and mitigation mechanisms. To achieve this service, the End-user negotiates the desired features with SPECS.* *In this validation scenario, the VM (without TLS) is provided by an Infrastructure-as-a-Service CSP while the TLS protocol and the DoS detection and mitigation mechanisms are provided by SPECS. SPECS acquires the resources on behalf of the End-user (registered on SPECS), adds the TLS protocol, and sets up some monitoring functionalities in order to monitor the TLS communication. In this scenario, an alert regarding a DoS attack is detected, and SPECS reacts by activating proper mitigation strategies. The scenario ends without any other alert.*		
Steps			
1	Phase	*SLA Negotiation*	
	Actor	*End-user, SPECS application, SPECS Negotiation module*	
	Preconditions	*The End-user has a very basic security knowledge, she/he is able to express qualitatively requirements at a high-level of abstraction.*	
	Trigger		
	Actions	*The End-user accesses the SPECS application interface. The negotiation request is forwarded to the SPECS Negotiation module, which retrieves the list of available SLA templates representing the available security services and the related security capabilities, controls and metrics. The services are returned to the End-user.*	
	Postconditions		
2	Phase	*SLA Negotiation*	
	Actor	*End-user, SPECS application*	
	Preconditions		
	Trigger		
	Actions	*The End-user selects, among the available service offers, the desired one, i.e., the Secure Web Container. The End-user specifies the desired security features by selecting the capabilities she/he is interested in and specifying the related security controls, and by specifying the desired metrics and setting related SLOs.*	
	Postconditions	*A supply chain compliant to the End-user requirements is built.*	
3	Phase	*SLA Negotiation*	
	Actor	*SPECS application, SPECS Negotiation module, SPECS Enforcement module*	
	Preconditions	*A web container, which fulfils the specific security requirements, is not known to SPECS.* *An Infrastructure-as-a-Service provider that offers plain VMs is known to SPECS, and the TLS and DoS detection and mitigation tools are offered as SPECS security mechanisms.*	
	Trigger		

	Actions	*The End-user's choices are forwarded by the SPECS application to the SPECS Negotiation module, which searches for valid supply chains. In particular, the list of supply chains is built with the help of the SPECS Enforcement module. In this step, an external CSP offering the Secure Web Container is identified. TLS, DoS detection and DoS mitigation components are identified among SPECS Enforcement security components.* *For each valid supply chain, a SLA Offer is created. The set of SLA Offers are hence ranked and returned to the SPECS application.*
	Postconditions	
4	Phase	*SLA Negotiation*
	Actor	*End-user, SPECS application, SLA Platform*
	Preconditions	*The End-user shall be logged on SPECS.*
	Trigger	
	Actions	*The SPECS application validates the SLA Offers which are then presented to the End-user. The End-user selects the SLA Offer in which the Secure Web Container is offered by an external CSP while the TLS, DoS detection and DoS mitigation are offered as SPECS security mechanisms. The selected SLA Offer is used to update and sign the SLA in the SLA Platform*
	Postconditions	*The SLA, containing all information needed for SLA implementation, has been signed.*
5	Phase	*SLA Implementation*
	Actor	*SPECS application, SPECS Enforcement module, SLA Platform*
	Preconditions	*A valid signed SLA containing all service terms and service guarantees is available in the SLA Platform*
	Trigger	
	Actions	*The SPECS application invokes the SPECS Enforcement module which retrieves the SLA to implement from the SLA Platform and prepares a plan to implement the signed SLA: it analyses the SLA, deduces alert thresholds, chooses the security and monitoring mechanisms to activate and determines all related software to install and their configurations.*
	Postconditions	
6	Phase	*SLA Implementation*
	Actor	*SPECS Enforcement module*
	Preconditions	*A plan has been built to implement a signed SLA.*
	Trigger	
	Actions	*The SPECS Enforcement module implements the plan, by configuring and deploying all the components in order to respect the features granted in the SLA. The SPECS Enforcement module deploys and configures monitoring agents and activates all the components and services.*
	Postconditions	
7	Phase	*SLA Implementation*
	Actor	*SPECS Enforcement module, SPECS Monitoring Module*
	Preconditions	*All components and services needed for SLA implementation have been correctly configured and activated.*
	Trigger	
	Actions	*The SPECS Enforcement module configures the Monitoring module with a monitoring policy by setting proper alert/violation thresholds for specific metrics.*
	Postconditions	
8	Phase	*SLA Monitoring*
	Actor	*SPECS Monitoring module*
	Preconditions	

	Trigger	
	Actions	*SPECS keeps collecting information about the provided service and evaluates them against the current monitoring policy.*
	Postconditions	
9	Phase	*SLA Remediation*
	Actor	*SPECS Monitoring module, SPECS Enforcement module*
	Preconditions	
	Trigger	*The SPECS Monitoring module generates monitoring events related to detection of DoS attack by the DoS Monitoring component.*
	Actions	*The SPECS Enforcement module analyses monitoring events and, relying upon the attack classification functionalities provided by the SPECS DoS Mitigation component, classifies it as an alert.*
	Postconditions	
10	Phase	*SLA Remediation*
	Actor	*SPECS Enforcement module*
	Preconditions	*Some mitigation strategies are available.*
	Trigger	*An alert has been detected.*
	Actions	*The SPECS Enforcement module reacts by activating proper mitigation strategies, defined by the SPECS DoS Mitigation component.*
	Postconditions	*The alert is solved and the SLA is completed since neither other alerts or violations occur.*
Graphical Model		*Not reported to avoid replication of information. See D1.3 for detailed interactions between SPECS modules.*
Coverage Information		
Users	*U_1 (CSC:User)*	
Target services	*TS_4 (Infrastructure as a Service)*	
SPECS services	*See Appendix B*	
SLA	*SLA_1, SLA_3, SLA_6, SLA_7, SLA_9, SLA_10, SLA_11, SLA_8*	

4.2.4. Secure_Web_Container_SVA_enhanced_alert

General Information		
ID	*SWC.4 - Secure_Web_Container_SVA_enhanced_alert*	
Version	*2.0*	
User Story	*WEB*	*Secure Web Container*
Invocation Chain	*IM1-CSP*	*Interaction Model 1- SPECS acting the role of CSP*
Scenario Steps		
General Description	*The End-user aims at acquiring a web container from an Infrastructure-as-a-Service CSP, represented by a VM hosting the Web Server, which fulfils specific security-related requirements. In particular, the End-user requires the adoption of Software Vulnerability Assessment (SVA) tools to protect the Web Server environment. To achieve this service, the End-user negotiates the desired features with SPECS.* *In this validation scenario, the VM (without SVA) is provided by an Infrastructure-as-a-Service CSP while the SVA agent is installed by SPECS. SPECS acquires the resources on behalf of the End-user (registered on SPECS), adds the SVA agents, and sets up some monitoring functionalities. This scenario includes the raising of an alert due to a deviation of some metrics; SPECS reacts by updating the software (redressing). The scenario ends without any other alerts.*	

Steps		
1	Phase	*SLA Negotiation*
	Actor	*End-user, SPECS application, SPECS Negotiation module*
	Preconditions	*The End-user has a very basic security knowledge, she/he is able to express qualitatively requirements at a high-level of abstraction.*
	Trigger	
	Actions	*The End-user accesses the SPECS application interface. The negotiation request is forwarded to the SPECS Negotiation module, which retrieves the list of available SLA templates representing the available security services and the related security capabilities, controls and metrics. The services are returned to the End-user.*
	Postconditions	
2	Phase	*SLA Negotiation*
	Actor	*End-user, SPECS application*
	Preconditions	
	Trigger	
	Actions	*The End-user selects, among the available service offers, the desired one, i.e., the Secure Web Container. The End-user specifies the desired security features by selecting the capabilities she/he is interested in and specifying the related security controls, and by specifying the desired metrics and setting related SLOs.*
	Postconditions	*A supply chain compliant to the End-user requirements is built.*
3	Phase	*SLA Negotiation*
	Actor	*SPECS application, SPECS Negotiation module, SPECS Enforcement module*
	Preconditions	*A web container, which fulfils the specific security requirements, is not known to SPECS.* *An Infrastructure-as-a-Service provider that offers plain VMs is known to SPECS, and SVA agents are offered as SPECS security mechanisms.*
	Trigger	
	Actions	*The End-user's choices are forwarded by the SPECS application to the SPECS Negotiation module, which searches for valid supply chains. In particular, the list of supply chains is built with the help of the SPECS Enforcement module. In this step, an external CSP offering the Secure Web Container is identified. SVA agents are identified among SPECS Enforcement security components.* *For each valid supply chain, a SLA Offer is created. The set of SLA Offers are hence ranked and returned to the SPECS application.*
	Postconditions	
4	Phase	*SLA Negotiation*
	Actor	*End-user, SPECS application, SLA Platform*
	Preconditions	*The End-user shall be logged on SPECS.*
	Trigger	
	Actions	*The SPECS application validates the SLA Offers which are then presented to the End-user. The End-user selects the SLA Offer in which the Secure Web Container is offered by an external CSP while the SVA agents are offered as SPECS security mechanisms. The selected SLA Offer is used to update and sign the SLA in the SLA Platform*
	Postconditions	*The SLA, containing all information needed for SLA implementation, has been signed.*
5	Phase	*SLA Implementation*
	Actor	*SPECS application, SPECS Enforcement module, SLA Platform*
	Preconditions	*A valid signed SLA containing all service terms and service guarantees is available in the SLA Platform*

	Trigger	
	Actions	*The SPECS application invokes the SPECS Enforcement module which retrieves the SLA to implement from the SLA Platform and prepares a plan to implement the signed SLA: it analyses the SLA, deduces alert thresholds, chooses the security and monitoring mechanisms to activate and determines all related software to install and their configurations.*
	Postconditions	
6	Phase	*SLA Implementation*
	Actor	*SPECS Enforcement module*
	Preconditions	*A plan has been built to implement a signed SLA.*
	Trigger	
	Actions	*The SPECS Enforcement module implements the plan, by configuring and deploying all the components in order to respect the features granted in the SLA (including the installation of SVA agents on the plain VM). The SPECS Enforcement module deploys and configures monitoring agents and activates all the components and services.*
	Postconditions	
7	Phase	*SLA Implementation*
	Actor	*SPECS Enforcement module, SPECS Monitoring Module*
	Preconditions	*All components and services needed for SLA implementation have been correctly configured and activated.*
	Trigger	
	Actions	*The SPECS Enforcement module configures the Monitoring module with a monitoring policy by setting proper alert/violation thresholds for specific metrics.*
	Postconditions	
8	Phase	*SLA Monitoring*
	Actor	*SPECS Monitoring module*
	Preconditions	
	Trigger	
	Actions	*SPECS keeps collecting information about the provided service and evaluates them against the current monitoring policy.*
	Postconditions	
9	Phase	*SLA Remediation*
	Actor	*SPECS Monitoring module, SPECS Enforcement module*
	Preconditions	
	Trigger	*The SPECS Monitoring module generates monitoring events related to the deviation of some metrics from set thresholds (e.g., number of exposed vulnerabilities).*
	Actions	*The SPECS Enforcement module makes an analysis of monitoring events and classifies them as an alert.*
	Postconditions	
10	Phase	*SLA Remediation*
	Actor	*SPECS Enforcement module*
	Preconditions	*The new version of the vulnerable software is available.*
	Trigger	*An alert regarding a vulnerability threat has been detected*
	Actions	*The SPECS Enforcement module reacts by activating the available redressing technique (it checks the presence of new versions, and updates the vulnerable software).*
	Postconditions	*The alert is solved.*

Graphical Model	*Not reported to avoid replication of information. See D1.3 for detailed interactions between SPECS modules.*
Coverage Information	
Users	*U_1 (CSC:User)*
Target services	*TS_4 (Infrastructure as a Service)*
SPECS services	*See Appendix B*
SLA	*SLA_1, SLA_3, SLA_6, SLA_7, SLA_9, SLA_10, SLA_11*

4.2.5. Secure_Web_Container_TLS_SVA_enhanced_violation

General Information			
ID		*SWC.5 - Secure_Web_Container_TLS_SVA_enhanced_violation*	
Version		*2.0*	
User Story		*WEB*	*Secure Web Container*
Invocation Chain		*IM1-CSP*	*Interaction Model 1- SPECS acting the role of CSP*
Scenario Steps			
General Description		*The End-user aims at acquiring a web container from an Infrastructure-as-a-Service CSP, represented by a VM hosting the Web Server, which fulfils specific security-related requirements. In particular, the End-user requires the adoption of Software Vulnerability Assessment (SVA) tools to protect the Web Server environment. To achieve this service, the End-user negotiates the desired features with the SPECS.* *In this validation scenario, the VM (without SVA) is provided by an Infrastructure-as-a-Service CSP while the SVA agents are installed by SPECS. SPECS acquires the resources on behalf of the End-user (registered on SPECS), adds the SVA agents, and sets up some monitoring functionalities in order to detect the presence of exposed vulnerabilities. This scenario includes the raising of an alert regarding a vulnerability threat which corresponds to a violation of the agreed SLA. SPECS reacts by renegotiating the SLA; the End-user asks for the adoption of Transport Layer Security (TLS) protocol to protect the Web Server communications. The renegotiated SLA is hence signed and properly monitored by SPECS.*	
Steps			
1	Phase	*SLA Negotiation*	
	Actor	*End-user, SPECS application, SPECS Negotiation module*	
	Preconditions	*The End-user has a very basic security knowledge, she/he is able to express qualitatively requirements at a high-level of abstraction.*	
	Trigger		
	Actions	*The End-user accesses the SPECS application interface. The negotiation request is forwarded to the SPECS Negotiation module, which retrieves the list of available SLA templates representing the available security services and the related security capabilities, controls and metrics. The services are returned to the End-user.*	
	Postconditions		
2	Phase	*SLA Negotiation*	
	Actor	*End-user, SPECS application*	
	Preconditions		
	Trigger		

	Actions	*The End-user selects, among the available service offers, the desired one, i.e., the Secure Web Container. The End-user specifies the desired security features by selecting the capabilities she/he is interested in and specifying the related security controls, and by specifying the desired metrics and setting related SLOs.*
	Postconditions	*A supply chain compliant to the End-user requirements is built.*
3	Phase	*SLA Negotiation*
	Actor	*SPECS application, SPECS Negotiation module, SPECS Enforcement module*
	Preconditions	*A web container, which fulfils the specific security requirements, is not known to SPECS.* *An Infrastructure-as-a-Service provider that offers plain VMs is known to SPECS, and SVA agents are offered as SPECS security mechanisms.*
	Trigger	
	Actions	*The End-user's choices are forwarded by the SPECS application to the SPECS Negotiation module, which searches for valid supply chains. In particular, the list of supply chains is built with the help of the SPECS Enforcement module. In this step, an external CSP offering the Secure Web Container is identified. SVA agents are identified among SPECS Enforcement security components.* *For each valid supply chain, a SLA Offer is created. The set of SLA Offers are hence ranked and returned to the SPECS application.*
	Postconditions	
4	Phase	*SLA Negotiation*
	Actor	*End-user, SPECS application, SLA Platform*
	Preconditions	*The End-user shall be logged on SPECS.*
	Trigger	
	Actions	*The SPECS application validates the SLA Offers which are then presented to the End-user. The End-user selects the SLA Offer in which the Secure Web Container is offered by an external CSP while the SVA agents are offered as SPECS security mechanisms. The selected SLA Offer is used to update and sign the SLA in the SLA Platform*
	Postconditions	*The SLA, containing all information needed for SLA implementation, has been signed.*
5	Phase	*SLA Implementation*
	Actor	*SPECS application, SPECS Enforcement module, SLA Platform*
	Preconditions	*A valid signed SLA containing all service terms and service guarantees is available in the SLA Platform*
	Trigger	
	Actions	*The SPECS application invokes the SPECS Enforcement module which retrieves the SLA to implement from the SLA Platform and prepares a plan to implement the signed SLA: it analyses the SLA, deduces alert thresholds, chooses the security and monitoring mechanisms to activate and determines all related software to install and their configurations.*
	Postconditions	
6	Phase	*SLA Implementation*
	Actor	*SPECS Enforcement module*
	Preconditions	*A plan has been built to implement a signed SLA.*
	Trigger	
	Actions	*The SPECS Enforcement module implements the plan, by configuring and deploying all the components in order to respect the features granted in the SLA (including the installation of SVA agents on the plain VM). The SPECS Enforcement module deploys and configures monitoring agents and activates all the components and services.*
	Postconditions	

7	Phase	*SLA Implementation*
	Actor	*SPECS Enforcement module, SPECS Monitoring Module*
	Preconditions	*All components and services needed for SLA implementation have been correctly configured and activated.*
	Trigger	
	Actions	*The SPECS Enforcement module configures the Monitoring module with a monitoring policy by setting proper alert/violation thresholds for specific metrics.*
	Postconditions	
8	Phase	*SLA Monitoring*
	Actor	*SPECS Monitoring module*
	Preconditions	
	Trigger	
	Actions	*SPECS keeps collecting information about the provided service and evaluates them against the current monitoring policy.*
	Postconditions	
9	Phase	*SLA Remediation*
	Actor	*SPECS Monitoring module, SPECS Enforcement module*
	Preconditions	
	Trigger	*The SPECS Monitoring module generates monitoring events related to the deviation of some metrics from set thresholds (e.g., number of exposed vulnerabilities).*
	Actions	*The SPECS Enforcement module makes an analysis of monitoring events and classifies them as a violation.*
	Postconditions	
10	Phase	*SLA Remediation*
	Actor	*SPECS Application, SPECS Enforcement module*
	Preconditions	*No remedies can be applied by SPECS; renegotiation is needed.*
	Trigger	*A violation of the signed SLA has been detected.*
	Actions	*SPECS notifies the violation to the End-User through the SPECS Application. The SPECS Enforcement module searches for alternatives for the End-user by building new services.*
	Postconditions	*The SLA is no more fulfilled*
11	Phase	*Renegotiation*
	Actor	*End-user, SPECS Application, SPECS Negotiation module*
	Preconditions	
	Trigger	
	Actions	*The End-user asks for the adoption of Transport Layer Security (TLS) protocol to protect the Web Server communications. The renegotiation follows the same activities described in the steps from 1 to 4.*
	Postconditions	*The renegotiated SLA is signed.*
12	Phase	*SLA Implementation*
	Actor	*SPECS Enforcement module, SPECS Monitoring Module*
	Preconditions	*A valid signed SLA containing all service terms and service guarantees is available.*
	Trigger	
	Actions	*The implementation of the SLA follows the same activities described in steps from 5 to 7.*
	Postconditions	
13	Phase	*SLA Monitoring*

	Actor	*SPECS Monitoring module*
	Preconditions	*The monitoring policy has been updated to include thresholds related to the SLA.*
	Trigger	
	Actions	*SPECS keeps collecting information about the provided service and evaluates them against the current monitoring policy.*
	Postconditions	
Graphical Model		*Not reported to avoid replication of information. See D1.3 for detailed interactions between SPECS modules.*
Coverage Information		
Users	*U_1 (CSC:User)*	
Target services	*TS_4 (Infrastructure as a Service)*	
SPECS services	*See Appendix B*	
SLA	SLA_1, SLA_3, SLA_6, SLA_7, SLA_13, SLA_14, SLA_17, SLA_19	

4.2.6. Secure_Web_Container_TLS_multitenancy

General Information		
ID	*SWC.6 - Secure_Web_Container_TLS_multitenancy*	
Version	*2.0*	
User Story	*WEB*	*Secure Web Container*
Invocation Chain	*IM1-CSP*	*Interaction Model 1- SPECS acting the role of CSP*
Scenario Steps		
General Description	*Two End-users aim at acquiring different web containers Infrastructure-as-a-Service CSPs, represented by VMs hosting the Web Servers, which fulfil different security requirements. In addition, both End-users require the adoption of Transport Layer Security (TLS) protocol to protect the communications of Web Servers. To achieve this service, the first End-user negotiates the desired features with SPECS. The VM (without TLS) is provided by an Infrastructure-as-a-Service CSP while the TLS protocol is added by SPECS setting up proper resources (e.g., reverse proxy).* *The second End-user negotiates the desired features with the SPECS framework. A different VM (without TLS) is provided by an Infrastructure-as-a-Service CSP (either the same or a different one) while the TLS protocol is added by SPECS reusing, for scalability purposes, the same resources adopted for the first End-user.* *This validation scenario considers the multi-tenancy in the usage of shared resources between End-users.*	
Steps		
1	Phase	*SLA Negotiation*
	Actor	*End-user (first), SPECS application, SPECS Negotiation module*
	Preconditions	*The End-user has a very basic security knowledge, she/he is able to express qualitatively requirements at a high-level of abstraction.*
	Trigger	
	Actions	*The first End-user accesses the SPECS application interface. The negotiation request is forwarded to the SPECS Negotiation module, which retrieves the list of available SLA templates representing the available security services and the related security capabilities, controls and metrics. The services are returned to the End-user.*
	Postconditions	
2	Phase	*SLA Negotiation*

	Actor	*End-user (first), SPECS application*
	Preconditions	
	Trigger	
	Actions	*The first End-user selects, among the available service offers, the desired one, i.e., the Secure Web Container. The End-user specifies the desired security features by selecting the capabilities she/he is interested in and specifying the related security controls, and by specifying the desired metrics and setting related SLOs.*
	Postconditions	*A supply chain compliant to the End-user requirements is built.*
3	Phase	*SLA Negotiation*
	Actor	*SPECS application, SPECS Negotiation module, SPECS Enforcement module*
	Preconditions	*A web container, which fulfils the specific security requirements, is not known to SPECS.* *An Infrastructure-as-a-Service provider that offers plain VMs is known to SPECS, and the TLS and DoS detection and mitigation tools are offered as SPECS security mechanisms.*
	Trigger	
	Actions	*The End-user's choices are forwarded by the SPECS application to the SPECS Negotiation module, which searches for valid supply chains. In particular, the list of supply chains is built with the help of the SPECS Enforcement module. In this step, an external CSP offering the Secure Web Container is identified. TLS, DoS detection and DoS mitigation components are identified among SPECS Enforcement security components.* *For each valid supply chain, a SLA Offer is created. The set of SLA Offers are hence ranked and returned to the SPECS application.*
	Postconditions	
4	Phase	*SLA Negotiation*
	Actor	*End-user (first), SPECS application, SLA Platform*
	Preconditions	*The End-user shall be logged on SPECS.*
	Trigger	
	Actions	*The SPECS application validates the SLA Offers which are then presented to the End-user. The End-user selects the SLA Offer in which the Secure Web Container is offered by an external CSP while the TLS, DoS detection and DoS mitigation are offered as SPECS security mechanisms. The selected SLA Offer is used to update and sign the SLA in the SLA Platform*
	Postconditions	*The SLA, containing all information needed for SLA implementation, has been signed.*
5	Phase	*SLA Implementation*
	Actor	*SPECS application, SPECS Enforcement module, SLA Platform*
	Preconditions	*A valid signed SLA containing all service terms and service guarantees is available in the SLA Platform*
	Trigger	
	Actions	*The SPECS application invokes the SPECS Enforcement module which retrieves the SLA to implement from the SLA Platform and prepares a plan to implement the signed SLA: it analyses the SLA, deduces alert thresholds, chooses the security and monitoring mechanisms to activate and determines all related software to install and their configurations.*
	Postconditions	
6	Phase	*SLA Implementation*
	Actor	*SPECS Enforcement module*
	Preconditions	*A plan has been built to implement a signed SLA.*
	Trigger	

	Actions	*The SPECS Enforcement module implements the plan, by configuring and deploying all the components in order to respect the features granted in the SLA. The SPECS Enforcement module deploys and configures monitoring agents and activates all the components and services.*
	Postconditions	
7	Phase	*SLA Implementation*
	Actor	*SPECS Enforcement module, SPECS Monitoring Module*
	Preconditions	*All components and services needed for SLA implementation have been correctly configured and activated.*
	Trigger	
	Actions	*The SPECS Enforcement module configures the Monitoring module with a monitoring policy by setting proper alert/violation thresholds for specific metrics.*
	Postconditions	
8	Phase	*SLA Monitoring*
	Actor	*SPECS Monitoring module*
	Preconditions	
	Trigger	
	Actions	*SPECS keeps collecting information about the provided service and evaluates them against the current monitoring policy.*
	Postconditions	
9	Phase	*SLA Negotiation*
	Actor	*End-user (second), SPECS application, SPECS Negotiation module, SPECS Enforcement module, SLA Platform*
	Preconditions	*The End-user has a very basic security knowledge, she/he is able to express qualitatively requirements at a high-level of abstraction.* *The End-user shall be logged on SPECS.*
	Trigger	
	Actions	*The second End-user accesses the SPECS application interface, asking for a secure web container which fulfils the specific security requirements.* *The negotiation follows the same activities described in the steps from 1 to 4.*
	Postconditions	*The SLA, containing all information needed for SLA implementation, has been signed.*
10	Phase	*SLA Implementation*
	Actor	*SPECS application, SPECS Enforcement module, SLA Platform, SPECS Monitoring Module*
	Preconditions	*A valid signed SLA containing all service terms and service guarantees is available in the SLA Platform*
	Trigger	
	Actions	*The SPECS application invokes the SPECS Enforcement module which prepares and implements the plan which implement the signed SLA. It configures the Monitoring module with a monitoring policy by setting proper alert/violation thresholds for specific metrics.* *The implementation of the SLA follows the same activities described in steps from 5 to 7 but the TLS protocol is added by reusing, for scalability purposes, the same resources adopted for the first End-user.*
	Postconditions	
11	Phase	*SLA Monitoring*
	Actor	*SPECS Monitoring module*
	Preconditions	
	Trigger	
	Actions	*SPECS keeps collecting information about the provided service and evaluates the monitoring policies.*

	Postconditions	*The signed SLA is fulfilled since neither alerts nor violations occur.*
Graphical Model		*Not reported to avoid replication of information. See D1.3 for detailed interactions between SPECS modules.*
Coverage Information		
Users	*U_1 (CSC:User)*	
Target services	*TS_4 (Infrastructure as a Service)*	
SPECS services	*See Appendix B*	
SLA	SLA_1, SLA_3, SLA_6, SLA_7	

4.2.7. Secure_Web_Container_Web_Pool_Replication_enhanced_alert

General Information		
ID	*SWC.7 - Secure_Web_Container_Web_Pool_Replication_enhanced_alert*	
Version	*2.0*	
User Story	*WEB*	*Secure Web Container*
Invocation Chain	*IM1-CSP*	*Interaction Model 1- SPECS acting the role of CSP*
Scenario Steps		
General Description	*The End-user aims at acquiring a set of web containers from an Infrastructure-as-a-Service CSP, each of them represented by a VM hosting the Web Server, which fulfil specific security-related requirements. In particular, the End-user requires a specific level of redundancy and session persistence among web container replicas. To achieve this service, the End-user negotiates the desired features with SPECS.* *In this validation scenario, the VMs are provided by an Infrastructure-as-a-Service CSP while session persistence among replicas is implemented by the SPECS web pool mechanism. SPECS acquires the resources on behalf of the End-user (registered on SPECS), adds the web pool components, and sets up proper resources to handle HTTP request through proxying functionality in order to forward the requests to one of the available the web container. In this scenario, the proxy functionality is added, by SPECS, on a dedicated VM.* *This scenario includes the rising of an alert regarding a vulnerability threat on a specific web container; SPECS reacts by updating the implemented forwarding policy (redressing) and removes the affected web container from the pool of available web containers. The scenario ends without any other alerts.*	
Steps		
1	Phase	*SLA Negotiation*
	Actor	*End-user, SPECS application, SPECS Negotiation module*
	Preconditions	*The End-user has a very basic security knowledge, she/he is able to express qualitatively requirements at a high-level of abstraction.*
	Trigger	
	Actions	*The End-user accesses the SPECS application interface. The negotiation request is forwarded to the SPECS Negotiation module, which retrieves the list of available SLA templates representing the available security services and the related security capabilities, controls and metrics. The services are returned to the End-user.*
	Postconditions	
2	Phase	*SLA Negotiation*
	Actor	*End-user, SPECS application*
	Preconditions	
	Trigger	

	Actions	*The End-user selects, among the available service offers, the desired one, i.e., the Secure Web Container. The End-user specifies the desired security features by selecting the capabilities she/he is interested in and specifying the related security controls, and by specifying the desired metrics and setting related SLOs.* *In particular, the End-user requires the adoption of a web pool mechanism to ensure session persistence among web container replicas*
	Postconditions	*A supply chain compliant to the End-user requirements is built.*
3	Phase	*SLA Negotiation*
	Actor	*SPECS application, SPECS Negotiation module, SPECS Enforcement module*
	Preconditions	*An Infrastructure-as-a-Service provider that offers VMs which fulfil the specific requirements is known to SPECS. The web pool mechanism is offered as a SPECS security mechanism.*
	Trigger	
	Actions	*The End-user's choices are forwarded by the SPECS application to the SPECS Negotiation module, which searches for valid supply chains. In particular, the list of supply chains is built with the help of the SPECS Enforcement module. In this step, an external CSP offering the Secure Web Container is identified; the web pool mechanism is identified among SPECS security mechanisms.* *For each valid supply chain, a SLA Offer is created. The set of SLA Offers are hence ranked and returned to the SPECS application.*
	Postconditions	
4	Phase	*SLA Negotiation*
	Actor	*End-user, SPECS application, SLA Platform*
	Preconditions	*The End-user shall be logged on SPECS.*
	Trigger	
	Actions	*The SPECS application validates the SLA Offers which are then presented to the End-user. The End-user selects the SLA Offer in which the Secure Web Container is offered by an external CSP while the web pool mechanism is offered as a SPECS security mechanism. The selected SLA Offer is used to update and sign the SLA in the SLA Platform*
	Postconditions	*The SLA, containing all information needed for SLA implementation, has been signed.*
5	Phase	*SLA Implementation*
	Actor	*SPECS application, SPECS Enforcement module, SLA Platform, SPECS Monitoring Module*
	Preconditions	
	Trigger	
	Actions	*SPECS acquires the VMs on behalf of the End-user on the external CSP and adds the web pool components, and sets up proper resources to handle HTTP request through proxying functionality in order to forward the requests to one of the available the web container. SPECS launches the related monitoring services.*
	Postconditions	
6	Phase	*SLA Monitoring*
	Actor	*SPECS Monitoring module*
	Preconditions	
	Trigger	
	Actions	*SPECS keeps collecting information about the provided service and evaluates the monitoring policies.*
	Postconditions	
7	Phase	*SLA Remediation*

	Actor	*SPECS Monitoring module, SPECS Enforcement module*
	Preconditions	*A redressing technique can be adopted according to the signed SLA, and is available as SPECS security mechanisms.*
	Trigger	*An alert regarding a vulnerability threat on a web container is raised by the enforcement diagnosis, after the notification of a monitoring event by the SPECS Monitoring module.*
	Actions	*SPECS updates the implemented forwarding policy (redressing technique) and removes the affected web container from the pool of available web containers*
	Postconditions	*The discovered vulnerabilities are solved and no more alerts are generated.*
Graphical Model		*Not reported to avoid replication of information. See D1.3 for detailed interactions between SPECS modules.*
Coverage Information		
Users	*U_1 (CSC:User)*	
Target services	*TS_4 (Infrastructure as a Service)*	
SPECS services	*See Appendix B*	
SLA	*SLA_1, SLA_3, SLA_6, SLA_7, SLA_8, SLA_9, SLA_10, SLA_11*	

4.2.8. Secure_Web_Container_Web_Pool_Replication_enhanced_violation

General Information		
ID	*SWC.8 - Secure_Web_Container_Web_Pool_Replication_enhanced_violation*	
Version	*2.0*	
User Story	*WEB*	*Secure Web Container*
Invocation Chain	*IM1-CSP*	*Interaction Model 1- SPECS acting the role of CSP*
Scenario Steps		
General Description	*The End-user aims at acquiring a precise number of web containers from an Infrastructure-as-a-Service CSP, each of them represented by a VM hosting the Web Server, which fulfil specific security requirements. In particular, the End-user requires a specific level of redundancy and session persistence among web container replicas. To achieve this service, the End-user negotiates the desired features with SPECS.* *In this validation scenario, the VMs are provided by an Infrastructure-as-a-Service CSP while the session persistence among replicas is implemented through the SPECS web pool mechanism by SPECS. SPECS acquires the resources on behalf of the End-user (registered on SPECS), adds the web pool components, and sets up proper resources to handle HTTP request through proxying functionality in order to forward the requests to one of the available the web container. In this scenario, the proxy functionality is added, by SPECS, on a dedicated VM.* *This scenario includes the rising of an alert regarding a vulnerability threat on a specific web container; SPECS reacts by removing the affected web container from the pool of available web containers. The signed SLA is hence violated since the number of available VMs is not sufficient to fulfil the SLA.*	
Steps		
1	Phase	*SLA Negotiation*
	Actor	*End-user, SPECS application, SPECS Negotiation module*
	Preconditions	*The End-user has a very basic security knowledge, she/he is able to express qualitatively requirements at a high-level of abstraction.*
	Trigger	

	Actions	*The End-user accesses the SPECS application interface. The negotiation request is forwarded to the SPECS Negotiation module, which retrieves the list of available SLA templates representing the available security services and the related security capabilities, controls and metrics. The services are returned to the End-user.*
	Postconditions	
2	Phase	*SLA Negotiation*
	Actor	*End-user, SPECS application*
	Preconditions	
	Trigger	
	Actions	*The End-user selects, among the available service offers, the desired one, i.e., the Secure Web Container. The End-user specifies the desired security features (in particular, the End-user requires the adoption of a web pool mechanism to ensure session persistence among web container replicas) by selecting the capabilities she/he is interested in and specifying the related security controls, and by specifying the desired metrics and setting related SLOs.*
	Postconditions	*A supply chain compliant to the End-user requirements is built.*
3	Phase	*SLA Negotiation*
	Actor	*SPECS application, SPECS Negotiation module, SPECS Enforcement module*
	Preconditions	*An Infrastructure-as-a-Service provider that offers VMs which fulfil the specific requirements is known to SPECS. The web pool mechanism is offered as a SPECS security mechanism.*
	Trigger	
	Actions	*The End-user's choices are forwarded by the SPECS application to the SPECS Negotiation module, which searches for valid supply chains. In particular, the list of supply chains is built with the help of the SPECS Enforcement module. In this step, an external CSP offering the Secure Web Container is identified; the web pool mechanism is identified among SPECS security mechanisms.* *For each valid supply chain, a SLA Offer is created. The set of SLA Offers are hence ranked and returned to the SPECS application.*
	Postconditions	
4	Phase	*SLA Negotiation*
	Actor	*End-user, SPECS application, SLA Platform*
	Preconditions	*The End-user shall be logged on SPECS.*
	Trigger	
	Actions	*The SPECS application validates the SLA Offers which are then presented to the End-user. The End-user selects the SLA Offer in which the Secure Web Container is offered by an external CSP while the web pool mechanism is offered as a SPECS security mechanism. The selected SLA Offer is used to update and sign the SLA in the SLA Platform*
	Postconditions	*The SLA, containing all information needed for SLA implementation, has been signed.*
5	Phase	*SLA Implementation*
	Actor	*SPECS application, SPECS Enforcement module, SLA Platform, SPECS Monitoring Module*
	Preconditions	
	Trigger	
	Actions	*SPECS acquires the VMs on behalf of the End-user on the external CSP and adds the web pool components, and sets up proper resources to handle HTTP request through proxying functionality in order to forward the requests to one of the available the web container. SPECS launches the related monitoring services.*

	Postconditions	
6	Phase	*SLA Monitoring*
	Actor	*SPECS Monitoring module*
	Preconditions	
	Trigger	
	Actions	*SPECS keeps collecting information about the provided service and evaluates the monitoring policies.*
	Postconditions	
7	Phase	*SLA Remediation*
	Actor	*SPECS Monitoring module, SPECS Enforcement module*
	Preconditions	
	Trigger	*An alert regarding a vulnerability threat on a web container is raised by the enforcement diagnosis, after the notification of a monitoring event by the SPECS Monitoring module.*
	Actions	*SPECS removes the affected web container from the pool of available web containers.*
	Postconditions	
8	Phase	*SLA Remediation*
	Actor	*SPECS Enforcement module*
	Preconditions	
	Trigger	*A violation of the signed SLA is detected by the enforcement diagnosis.*
	Actions	*SPECS notifies the violation to the End-user.*
	Postconditions	*The SLA is no more fulfilled*
Graphical Model		*Not reported to avoid replication of information. See D1.3 for detailed interactions between SPECS modules.*
Coverage Information		
Users	*U_1 (CSC:User)*	
Target services	*TS_4 (Infrastructure as a Service)*	
SPECS services	*See Appendix B*	
SLA	*SLA_1, SLA_3, SLA_6, SLA_7, SLA_9, SLA_12*	

4.2.9. Secure_Web_Container_ClientEncryption_Replication

This VS has been removed since, during Y2, the End-2-end encryption mechanism has been offered as an enhancement of the Database and Backup mechanism, which provides storage and assures business continuity through backup.

4.2.10. Secure_Web_Container_ClientEncryption_Replication_alert

This VS has been removed since, during Y2, the End-2-end encryption mechanism has been offered as an enhancement of the Database and Backup mechanism, which provides storage and assures business continuity through backup. Furthermore, the following scenario has been added: *Secure_Storage_Brokering_with_Client_Crypto_alert*.

4.2.11. Secure_Web_Container_ClientEncryption_Replication_violation

This VS has been removed since, during Y2, the End-2-end encryption mechanism has been offered as an enhancement of the Database and Backup mechanism, which provides storage and assures business continuity through backup. Furthermore, the following scenario has been added: *Secure_Storage_Brokering_with_Client_Crypto_violation*.

4.3. Usage of a Security-Oriented Dashboard

4.3.1. DM_Dashboard_Security_CSP_NonExpert

A deeper analysis has highlighted that this VS is similar to the following ones from the End-user point of view:

- *Secure_Storage_Selection*;
- *Secure_Web_Container_Selection;*
- *DM_Dashboard_Security_CSP_Expert*.

For this reason, this scenario has been deleted during Y2 while the *Secure_Storage_Selection* scenario has been revised and updated. The obtained coverage level is the same.

4.3.2. DM_Dashboard_Security_CSP_Expert

A deeper analysis has highlighted that this VS is similar to the following ones from the End-user point of view:

- *Secure_Storage_Selection*;
- *Secure_Web_Container_Selection;*
- *DM_Dashboard_Security_CSP_NonExpert*.

For this reason, this scenario has been deleted during Y2 while the *Secure_Web_Container_Selection* scenario has been revised and updated. The obtained coverage level is the same.

4.4. Next-Generation Data Centers

4.4.1. Data_Center_Bursting_for_Storage_Resources

General Information		
ID	*NGDC.1 - Data_Center_Bursting_for_Storage_Resources*	
Version	*1.1*	
User Story	*ngDC*	*Next Generation Data Center*
Invocation Chain	*IM2-CSP*	*Interaction Model 2- SPECS acting in the role of CSP*
Scenario Steps		
General Description	*A CSP hosting its own ngDC acting within a CSC role aims at using the SPECS framework to perform Cloud bursting in order to extend its Secure Storage as a Service (SStaaS) capabilities during a period of increased storage demand beyond its own ngDC storage capabilities by its CSCs and/or End-users.* *The CPS considers its storage as first class storage due the capability to tune all the security parameters. The CSP will allocate the first class storage to the End-User that don't need high-security capability. Otherwise it will allocate storage to an external provider throw SPECS. All that process is transparent to the End-user.* *Note while the CSP acquiring external CSP storage resources is typically considered an End-user, it is not in the context of a SPECS defined End-user. That is, the CSP intends to resell its acquired external storage resources and so is considered a CSC (in the context of SPECS). For ease of exposition 'customer' is used as a syntactic sugar to refer to either a CSC or End-user of the CSP hosting the ngDC.*	
Steps		
1	Phase	*Negotiation*

	Actor	*CSC (CSP is acting within a CSC role)*
	Preconditions	*The CSC monitors the current state of its ngDC in terms of its on-premise storage resources.*
	Trigger	*Capacity threshold reached.*
	Actions	*The CSC asks its locally hosted SPECS for an external CSP offering SStaaS, which fulfils its specific security requirements. These security requirements might be based on either or both the CSC's own security requirements or that of the CSC's own customers. Examples of security requirements are the data geo-location, the Drive type, RAID level, etc.*
	Postconditions	
2	Phase	*Negotiation*
	Actor	*SPECS Negotiation module*
	Preconditions	*An external CSP that fulfils the specific secure storage requirements must already be present within the locally hosted CSC's SPECS SLA Repository.*
	Trigger	
	Actions	*SPECS searches for possible supply chains compliant with the specified secure storage requirements, evaluates if the external CSP fulfils the End-User requirements SPECS will allocate directly the resource, otherwise it will allocate resource on the local storage platform.*
	Postconditions	
3	Phase	*Negotiation*
	Actor	*CSC*
	Preconditions	
	Trigger	
	Actions	*The CSC selects one supply chain from the retrieved list and signs the SLA with the external CSPs that form part of the SPECS supply chain.*
	Postconditions	
Graphical Model		
Coverage Information		
Users	*U_1(CSC:user)*	
Target services	*TS_3(Data Storage as a Service)*	
SPECS services	*See Appendix B*	
SLA Lifecycle	*SLA_1, SLA_3. SLA_4, SLA_5, SLA_6*	

4.4.2. Data_Center_Bursting_Backup_and_Archive_Resources

A deeper analysis has highlighted that this VS is just a specialisation of the previous one: moreover, its presence does not add more coverage (it does not dominate any other VS). Hence, it has been deleted.

4.4.3. Data_Center_Storage_Selection

General Information		
ID	*NGDC.3 – Data_Center_Storage_Selection*	
Version	*1.0*	
User Story	*NgDC*	*Next Generation Data Center*
Invocation Chain	*IM2-CSP*	*Interaction Model 2- SPECS acting the role of CSP*
Scenario Steps		

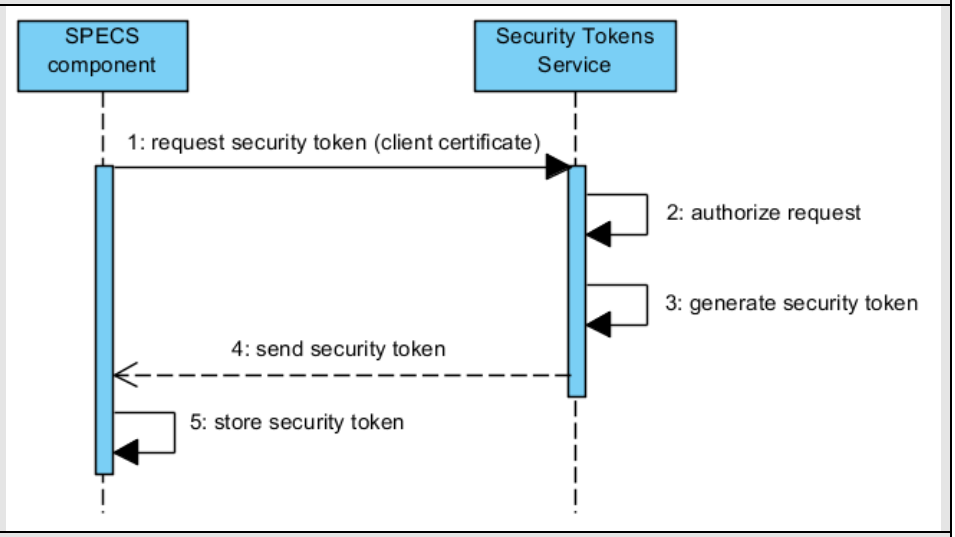
General Description		*A CSP owning SPECS and hosting its own ngDC acting within a CSC role aims at using the SPECS framework to perform Cloud bursting in order to extend its Secure Storage as a Service (SStaaS) capabilities during a period of increased storage demand beyond its own ngDC storage capabilities by its CSCs and/or End-users.* *In this validation scenario, the desired features are entirely implemented by an external CSP, while SPECS only aids the End-user with the functionalities to search, rank and select a service which is compliant to her/his requirements. Moreover, in this scenario, SPECS supports the End-user in signing an SLA with the selected provider.*
Steps		
1	Phase	*SLA Negotiation*
	Actor	*CSC (CSP is acting within a CSC role)*
	Preconditions	*The End-user has a good security knowledge, she/he is able to express qualitatively requirements at a high-level of abstraction.*
	Trigger	
	Actions	*The CSC asks its locally hosted SPECS for an external CSP offering SStaaS, which fulfils its specific security requirements. These security requirements might be based on either or both the CSC's own security requirements or that of the CSC's own customers. Examples of security requirements are the data geo-location, the Drive type, RAID level, etc.* *The End-user accesses the SPECS application interface. The negotiation request is forwarded to the SPECS Negotiation module, which retrieves the list of available SLA templates representing the available security services and the related security capabilities, controls and metrics. The services are returned to the End-user.*
	Postconditions	
2	Phase	*SLA Negotiation*
	Actor	*CSC, SPECS application*
	Preconditions	
	Trigger	
	Actions	*The End-user selects, among the available service offers, the desired one, i.e., the Database and Backup. The End-user specifies the desired security features by selecting the capabilities she/he is interested in and specifying the related security controls, and by specifying the desired metrics and setting related SLOs.*
	Postconditions	*A supply chain compliant to the End-user requirements is built.*
3	Phase	*SLA Negotiation*
	Actor	*SPECS application, SPECS Negotiation module, SPECS Enforcement module*
	Preconditions	*A secure storage service which fulfils the specific security requirements is known to SPECS.*
	Trigger	
	Actions	*The End-user's choices are forwarded by the SPECS application to the SPECS Negotiation module, which searches for valid supply chains. In particular, the list of supply chains is built with the help of the SPECS Enforcement module.* *For each valid supply chain, a SLA Offer is created. The set of SLA Offers are hence ranked and returned to the SPECS application. The CSPs also add the cost of each service offer.*
	Postconditions	
4	Phase	*SLA Negotiation*
	Actor	*End-user, SPECS application, SLA Platform*
	Preconditions	*The End-user shall be logged on SPECS.*
	Trigger	

	Actions	*The SPECS application validates the SLA Offers which are then presented to the End-user. The service offer is associated with an SLA published by an external CSP.* *The End-user either:* *1. accepts and signs the SLA offered by the external CSP;* *2. does not select any SLA Offer from the list and repeats the whole process from step 1 (possibly specifying a different set of requirements);* *3. does not select any SLA Offer from the list and exits the application.*
	Postconditions	*In case 1 - the signed SLA is stored by SPECS. The End-user is enabled to invoke the desired service on the external CSP with the configuration information included in the SLA.*
Graphical Model		*Not reported to avoid replication of information. See D1.3 for detailed interactions between SPECS modules.*
Coverage Information		
Users	*U_1 (CSC:User)*	
Target services	*TS_3 (Data Storage as a Service)*	
SPECS services	*See Appendix B*	
SLA	*SLA_1, SLA_3. SLA_4, SLA_5*	

4.5. Cross-cutting validation scenarios

4.5.1. Security_Tokens_Acquisition

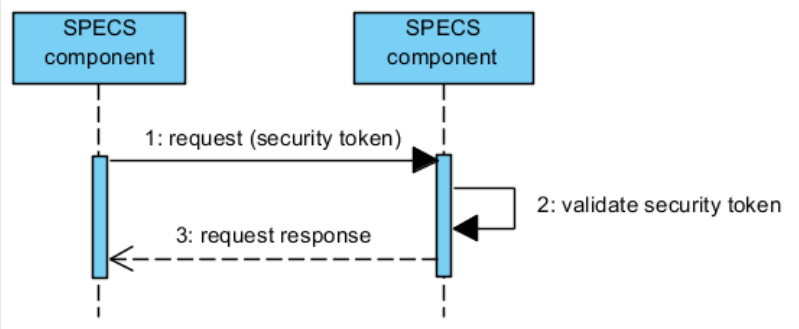
General Information		
ID	*CRO.1 - Security_Tokens_Acquisition*	
Version	*2.0*	
User Story	*n.d.*	
Invocation Chain	*n.d.*	
Scenario Steps		
General Description	*Each invocation of a SPECS component API must be authenticated and authorized through a proper mechanism based on security tokens.* *In this validation scenario, the acquisition of a security token is shown.*	
Steps		
1	Phase	*Token Acquisition*
	Actor	*SPECS component*
	Preconditions	*The component has a valid client certificate.*
	Trigger	*The SPECS component would like to call some SPECS service.*
	Actions	*The SPECS component sends a request to the Security Tokens Service and asks for a security token. It authenticates with its client certificate.*
	Postconditions	
2	Phase	*Token Acquisition*
	Actor	*Security Tokens Service*
	Preconditions	*The SPECS component authenticated with valid a client certificate.*
	Trigger	
	Actions	*The Security Tokens Service authorizes the request for a security token.*
	Postconditions	
3	Phase	*Token Acquisition*
	Actor	*Security Tokens Service*

	Preconditions	*The client is authorized to request a security token.*
	Trigger	
	Actions	*The Security Tokens Service generates a security token containing the subject and list of services the token is eligible to access and returns it to the client.*
	Postconditions	
4	Phase	*Token Acquisition*
	Actor	*SPECS component*
	Preconditions	*The request for a security token was granted.*
	Trigger	
	Actions	*The SPECS component stores the security token to the token vault noting the token's expiration time.*
	Postconditions	
4	Phase	*Token Acquisition*
	Actor	*SPECS component*
	Preconditions	*The SPECS component has a valid security token*
	Trigger	
	Actions	*The SPECS component calls some SPECS service, attaching the security token to the request. When making REST API calls, the security token is put in the HTTP header named X-AUTH-TOKEN. All communication among components is encrypted by using secure HTTPS connection.*
	Phase	
Graphical Model		

Coverage Information	
Users	*U_1 (CSC:User)*
Target services	*Not Applicable.*
SPECS services	*See Appendix B*
SLA	*Not Applicable.*

4.5.2. Security_Tokens_Validation

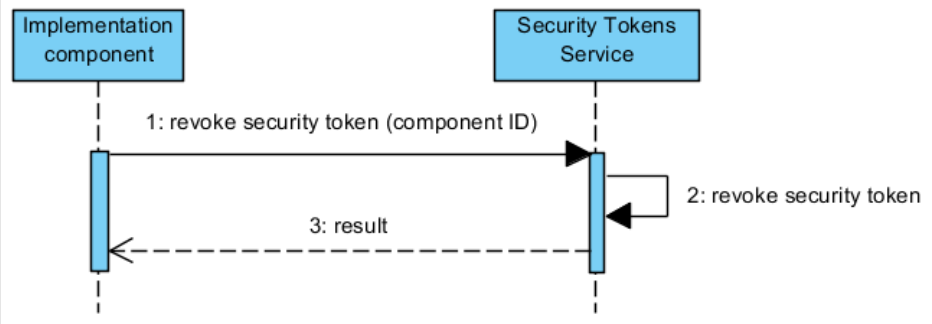
General Information		
ID	*CRO.2 - Security_Tokens_Validation*	
Version	*2.0*	
User Story	*n.d.*	
Invocation Chain	*n.d.*	

Scenario Steps		
General Description		*Each invocation of a SPECS component API must be authenticated and authorized through a proper mechanism based on security tokens. In this validation scenario, the validation of a security token is shown.*
Steps		
1	Phase	*Token Validation*
	Actor	*SPECS component*
	Preconditions	*The SPECS component has a valid security token.*
	Trigger	
	Actions	*The SPECS component calls another SPECS component, attaching the security token to the request.*
	Postconditions	
2	Phase	*Token Validation*
	Actor	*SPECS component*
	Preconditions	
	Trigger	
	Actions	*The SPECS component uses the security-tokens-client library to validate and decode the token.*
	Postconditions	
3	Phase	*Token Validation*
	Actor	*SPECS component*
	Preconditions	*The security token is valid.*
	Trigger	
	Actions	*The SPECS component authorizes the request based on the information in the security token using XACML authorization engine.*
	Postconditions	
Graphical Model		

Coverage Information	
Users	*U_1 (CSC:User)*
Target services	*Not Applicable.*
SPECS services	*See Appendix B*
SLA	*Not Applicable.*

4.5.3. Security_Tokens_Revocation

General Information		
ID	*CRO.3 - Security_Tokens_Revocation*	
Version	*2.0*	
User Story	*n.d.*	
Invocation Chain	*n.d.*	

Scenario Steps		
General Description		*In this validation scenario, the revocation of a security token is shown.*
Steps		
1	Phase	*Token Revocation*
	Actor	*Implementation component*
	Preconditions	
	Trigger	*The SLA is terminated.*
	Actions	*The Implementation component sends request to the Security Tokens Service to revoke the security tokens issued to a specific SPECS component. The Implementation component authenticates with its certificate.*
	Postconditions	
2	Phase	*Token Revocation*
	Actor	*Security Tokens Service*
	Preconditions	*The revoke request is authenticated and authorized.*
	Trigger	
	Actions	*The Security Tokens Service finds the tokens issued to the specified SPECS component, marks them as revoked and adds them to the token revocation list.*
	Postconditions	
3	Phase	*Token Revocation*
	Actor	*All SPECS components*
	Preconditions	
	Trigger	*Periodical update of the token revocation list*
	Actions	*SPECS components periodically pull delta token revocation list and update local token revocation list cache. The revoked tokens are propagated to the local token revocation list caches.*
	Postconditions	
4	Phase	*Token Revocation*
	Actor	*Blocked component*
	Preconditions	*The revoked tokens were propagated to local token revocation list caches.*
	Trigger	
	Actions	*The blocked component calls some other SPECS component with security token attached. The target component validates the token, finds out the token is on the revocation list and denies the request.*
	Postconditions	
Graphical Model		

Coverage Information	
Users	*U_1 (CSC:User)*
Target services	*Not Applicable.*

SPECS services	*See Appendix B*
SLA	*Not Applicable.*

4.5.4. Credential_Management

General Information			
ID		*CRO.4 - Credential_Management*	
Version		*2.0*	
User Story		*n.d.*	
Invocation Chain		*n.d.*	
Scenario Steps			
General Description		*The SPECS Credential Management service handles the authentication/authorization requests coming from non-human clients on behalf of End-users and targeted to a CSP.* *In this scenario, the interactions between the Secure Provisioning component and the Credential Management component are illustrated, with respect to the authentication of SPECS with the CSP through authentication tokens.* *In details, the Credential Management component stores SPECS credentials for the CSP and performs the authentication by returning authentication tokens, used for the request.*	
Steps			
1	Phase	*Authentication Information Acquisition*	
	Actor	*SPECS Secure Provisioning component, SPECS Credential Management component*	
	Preconditions		
	Trigger		
	Actions	*The SPECS Secure Provisioning component requests an authentication token to the Credential Management component related to a specific CSP.*	
	Postconditions		
2	Phase	*Authentication*	
	Actor	*SPECS Credential Management component, SPECS Secure Provisioning component, CSP*	
	Preconditions		
	Trigger		
	Actions	*The SPECS Credential Management component retrieves SPECS credentials for the CSP and performs authentication at the CSP. A token is retrieved and passed to the SPECS Secure Provisioning component.*	
	Postconditions		
3	Phase	*Service Invocation*	
	Actor	*SPECS Secure Provisioning component, CSP*	
	Preconditions		
	Trigger		
	Actions	*The SPECS Secure Provisioning component sends a request to the CSP along with the authentication token.*	
	Postconditions	*The resources is acquired.*	
Graphical Model		*Not reported to avoid replication of information. See D1.3 for detailed interactions between SPECS modules.*	
Coverage Information			
Users		*n.d.*	
Target services		*Not Applicable.*	

SPECS services	*Se See Appendix B*
SLA	*Not Applicable.*

4.5.5. User_Direct_Registration

General Information			
ID		*CRO.5 - User_Direct_Registration*	
Version		*2.0*	
User Story		*WEB*	*Secure Web Container*
Invocation Chain		*IM1-CSP*	*Interaction Model 1- SPECS acting the role of CSP*
Scenario Steps			
General Description		*Some SPECS services are offered to registered End-users. In this validation scenario, the registration is performed manually by the End-user, by inserting her/his personal information through the compilation of proper forms. The process ends with SPECS adding the registered user to the user list.*	
Steps			
1	Phase	*Registration*	
	Actor	*End-user*	
	Preconditions		
	Trigger		
	Actions	*The End-user fills the registration form with her/his personal information and submits it.*	
	Postconditions		
2	Phase	*Registration*	
	Actor	*SPECS AAA component*	
	Preconditions	*The End-user is not registered yet on SPECS*	
	Trigger		
	Actions	*The SPECS user repository is updated by adding a new entry with the information of the End-user.*	
	Postconditions	*The End-user's information is stored in the SPECS user repository.*	
Graphical Model		*Not reported to avoid replication of information. See D1.3 for detailed interactions between SPECS modules.*	
Coverage Information			
Users	*U_1 (CSC:User)*		
Target services	*Not Applicable*		
SPECS services	*See Appendix B*		
SLA	*Not Applicable*		

4.5.6. User_Registration_External_Account

General Information		
ID	*CRO.6 - User_Registration_External_Account*	
Version	*2.0*	
User Story	*WEB*	*Secure Web Container*
Invocation Chain	*IM1-CSP*	*Interaction Model 1- SPECS acting the role of CSP*
Scenario Steps		

General Description		*Some SPECS services are offered to registered End-users. In this validation scenario, the registration is performed by using a pre-existing external account (e.g., from an account of a social network or from an LDAP entry).* *The process ends with SPECS adding the registered user to the user list and linking it with the external account.*
Steps		
1	Phase	*Registration*
	Actor	*End-user*
	Preconditions	
	Trigger	
	Actions	*The End-user submits an authentication request (through, for example, a SAML request) to the SPECS AAA component.*
	Postconditions	
2	Phase	*Registration*
	Actor	*End-user*
	Preconditions	*The End-user has a valid account on the selected external authentication source.*
	Trigger	
	Actions	*The End-user selects the external authentication source and performs the login with the credentials of the external account, retrieving her/his personal information.*
	Postconditions	
3.1	Phase	*Registration*
	Actor	*SPECS AAA component*
	Preconditions	*The End-user is not registered yet on SPECS.*
	Trigger	
	Actions	*The SPECS user repository is updated by adding a new entry with the information of the End-User from the external authentication source. A link to the external account is also created.*
	Postconditions	*The End-user's information, along with the link to the external account, is stored in the SPECS user repository.*
3.2	Phase	*Registration*
	Actor	*SPECS AAA component*
	Preconditions	*The End-user is already registered on SPECS, and the link with the external account has not been yet specified.*
	Trigger	
	Actions	*The link with the external account is created for the user entry.*
	Postconditions	*The link to the external account is stored in the SPECS user repository.*
Graphical Model		*Not reported to avoid replication of information. See D1.3 for detailed interactions between SPECS modules.*

Coverage Information	
Users	*U_1 (CSC:User)*
Target services	*Not Applicable*
SPECS services	*See Appendix B*
SLA	*Not Applicable*

4.5.7. User_Authentication_External_Account

General Information

ID		*CRO.7 - User_Authentication_External_Account*	
Version		*2.0*	
User Story		*WEB*	*Secure Web Container*
Invocation Chain		*IM1-CSP*	*Interaction Model 1- SPECS acting the role of CSP*
Scenario Steps			
General Description		*Some SPECS services are offered to authenticated End-users. In this validation scenario, the authentication is performed by using a pre-existing external account (e.g., social accounts as Facebook, Twitter, or from an LDAP entry).* *When the user chooses to authenticate through an external source, SPECS checks that the external account is associated with a valid SPECS account. In this case, the user is authenticated. Otherwise SPECS asks if she/he wants to associate the external account to her/his existing SPECS account. In this latter case, the End-user must be preliminary authenticated on SPECS in order to prove the ownership of the SPECS account.*	
Steps			
1	Phase	*Authentication*	
	Actor	*End-user*	
	Preconditions		
	Trigger		
	Actions	*The End-user submits an authentication request (through, for example, an SAML request) to the SPECS AAA component.*	
	Postconditions		
2	Phase	*Authentication*	
	Actor	*End-user*	
	Preconditions	*The End-user has a valid account on the selected external authentication source.*	
	Trigger		
	Actions	*The End-user selects the external authentication source and performs the login with the credentials of the external account, retrieving her/his personal information.*	
	Postconditions	*The End-user is authenticated on the external authentication source.*	
3.1	Phase	*Authentication*	
	Actor	*SPECS AAA component*	
	Preconditions	*A SPECS account exists for the End-user. The SPECS account is already linked to the external account.*	
	Trigger		
	Actions	*SPECS authenticates the End-user.*	
	Postconditions	*The End-user is authenticated on SPECS.*	
3.2	Phase	*Authentication*	
	Actor	*SPECS AAA component*	
	Preconditions	*A SPECS account exists for the End-user. The SPECS account is not yet linked to the external account.*	
	Trigger		
	Actions	*SPECS asks the End-user to associate the external account to her/his existing SPECS account.*	
	Postconditions		
4.2	Phase	*Authentication*	
	Actor	*End-user*	

	Preconditions	
	Trigger	
	Actions	*The End-user logs into SPECS with the credentials of the SPECS account.*
	Postconditions	*The End-user is authenticated on the external authentication source.*
5.2	Phase	*Authentication*
	Actor	*SPECS AAA component*
	Preconditions	
	Trigger	
	Actions	*The link with the external account is created for the user entry and SPECS authenticates the End-user.*
	Postconditions	*The link to the external account is stored in the SPECS user repository, and the End-user is authenticated on SPECS.*
Graphical Model		*Not reported to avoid replication of information. See D1.3 for detailed interactions between SPECS modules.*
Coverage Information		
Users	*U_1 (CSC:User)*	
Target services	*Not Applicable*	
SPECS services	*See Appendix B*	
SLA	*Not Applicable*	

4.5.8. Metric_Definition

General Information			
ID		*CRO.8 - Metric_Definition*	
Version		*1.0*	
User Story		*n.d.*	
Invocation Chain		*n.d.*	
Scenario Steps			
General Description		*A SPECS user can manage easily a catalogue of security metrics and can also define her/his own security metric. In this scenario, the definition of a new security metric is shown.*	
Steps			
1	Phase	*Retrieve Metric*	
	Actor	*End-user, Security Metric Catalogue*	
	Preconditions	*The End-user shall be logged on SPECS*	
	Trigger		
	Actions	*The End-user accesses the section of SPECS in which the metric catalogue is stored. She/he finds the set of stored metrics and retrieves needed information in a structured way.*	
	Postconditions		
2	Phase	*Store Metric*	
	Actor	*End-user, Security Metric Catalogue*	
	Preconditions		
	Trigger		

	Actions	*The End-user compiles a form to define a new metric. Specifically, she/he chooses the type of the metric and compile the appropriate fields. The End-user asks for the storing of the defined metrics.*
	Postconditions	*The defined metric is added in the Metric Catalogue*
3	Phase	*Store Metric*
	Actor	*End-user, Security Metric Catalogue*
	Preconditions	
	Trigger	
	Actions	*The End-user decides to update an already defined metric, by selecting the specific metric she/he wants to update. The chosen metric is shown in a structured way by the Security Metric Catalogue and the End-user can update easily the appropriate fields. The End-user asks for the storing of the updates.*
	Postconditions	*The metric is updated in the Metric Catalogue*
Graphical Model		*Not reported to avoid replication of information. See D1.3 for detailed interactions between SPECS modules.*
Coverage Information		
Users	*U_1 (CSC:User)*	
Target services	*Not Applicable*	
SPECS services	*See Appendix B*	
SLA	*Not Applicable*	

4.5.9. Security_Mechanism_Development

General Information		
ID	*CRO.9 - Security_Mechanism_Development*	
Version	*1.0*	
User Story	*n.d.*	
Invocation Chain	*n.d.*	
Scenario Steps		
General Description	*A SPECS developer aims at developing a new SPECS security mechanism and integrating it into the SPECS framework. In this scenario, the development of a new security mechanisms and its integration into the SPECS framework is shown. Commercial-off-the-Shelf components are used.*	
Steps		
1	Phase	*Define Services*
	Actor	*SPECS developer*
	Preconditions	
	Trigger	
	Actions	*The SPECS developer defines the security properties that the security mechanism she/he want to develop is able to grant and the types of services to which the mechanism can be applied. Specifically, she/he identifies the security capabilities enforced by the mechanism and the associated security grants. She/he also defines the remediation process associated with the developed security mechanism.*
	Postconditions	
2	Phase	*Define Mechanism Architecture*
	Actor	*SPECS developer*
	Preconditions	
	Trigger	

	Actions	*The SPECS developer identifies concretely the technologies and the solutions to be implemented through Chef recipes. Specifically, she/he maps each security metric to one basic measurement with which the system can identify possible violations. Each basic measurement is associated to at least one additional measurement.*
	Postconditions	
3	Phase	*Define Remediation Process, RDS SPECS component*
	Actor	*SPECS developer*
	Preconditions	
	Trigger	
	Actions	*The SPECS developer identifies the set of recipes that RDS SPECS component will use to automate the SLA remediation.*
	Postconditions	
4	Phase	*Prepare Mechanism Metadata*
	Actor	*SPECS developer, SPECS SLA Platform*
	Preconditions	
	Trigger	
	Actions	*The SPECS developer prepares the description of the mechanism behaviours, according to the SPECS security mechanism metadata. The developed description is stored in the SLA Platform in order to automate the SLA life cycle management process.*
	Postconditions	
5	Phase	*Prepare Mechanism Cookbook*
	Actor	*SPECS developer*
	Preconditions	
	Trigger	
	Actions	*The SPECS developer prepares the cookbook which automates the security mechanism's execution. The cookbook is organized according to Chef rules. SPECS Monitoring Adapter must be developed accordingly. The SPECS developer tests the developed security mechanism.*
	Postconditions	
Graphical Model		*Not reported to avoid replication of information. See D1.3 for detailed interactions between SPECS modules.*
Coverage Information		
Users	*U_4, U_6 (CSN:developer)*	
Target services	*Not Applicable*	
SPECS services	*See Appendix B*	
SLA	*Not Applicable*	

4.5.10. SPECS_Application_Development

General Information		
ID	*CRO.10 - SPECS_Application_Development*	
Version	*1.0*	
User Story	*n.d.*	
Invocation Chain	*n.d.*	
Scenario Steps		

General Description		*A SPECS developer aims at developing a new SPECS application. In this scenario, the development of a new SPECS application by using the default SPECS application as template is shown.*
Steps		
1	Phase	*Cloud Service Definition*
	Actor	*SPECS developer*
	Preconditions	
	Trigger	
	Actions	*The SPECS developer defines the types of cloud services to deliver and prepares the related cookbooks. She/he needs to specify the mechanisms able to enforce specific security capabilities and/or to monitor specific metrics, as well as she/he needs to provide proper mechanisms to automatically deploy and configure the target services themselves*
	Postconditions	
2	Phase	*Prepare Security Mechanisms*
	Actor	*SPECS developer*
	Preconditions	
	Trigger	
	Actions	*The SPECS developer selects, among available security mechanisms, those needed to offer the cloud services.*
	Postconditions	
3	Phase	*Prepare SLA Template*
	Actor	*SPECS developer*
	Preconditions	
	Trigger	
	Actions	*The SPECS developer builds a WS-Agreement-compliant SLA template, which summarizes the security capabilities that can be offered and the related guarantees.*
	Postconditions	
4	Phase	*Deploy SLA Templates and Security Mechanisms*
	Actor	*SPECS developer, SLA Platform*
	Preconditions	
	Trigger	
	Actions	*The SPECS developer deploys the security mechanisms in order to make them available to the SPECS application. All the cookbooks must be registered with the Chef Server in order to enable the SPECS Enforcement module to implement the SLA, and the mechanisms' metadata must be registered in the SLA Platform in order to enable the SPECS application to retrieve the information and to implement the SLA.* *The SPECS developer tests the deployed SPECS application.*
	Postconditions	
Graphical Model		*Not reported to avoid replication of information. See D1.3 for detailed interactions between SPECS modules.*
Coverage Information		
Users		*U_4, U_5, U_6 (CSN:developer)*
Target services		*Not Applicable*
SPECS services		*See Appendix B*
SLA		*Not Applicable*

5. Key Concern Coverage Approach

An important objective of this task is to measure the coverage level of the Key Concerns accomplished by the definition and the execution of VSs. As also explained in D5.1.1, the coverage of the five Key Concerns is the key to such measurement; they are Users (i.e., the kinds of the SPECS users), Invocation Chains (i.e., the possible deployment configurations), Target Services (i.e., the kinds of –as-a-service SPECS called to improve security), SLA lifecycle (i.e., the transitions between states of the SLA state machine) and the SPECS Services (i.e., the SPECS requirements). While the coverage level of first four of these concerns is measured directly starting from the VSs, the coverage of the SPECS requirements is done indirectly by means of SPECS components: VSs are mapped onto components by means of the *Validation-Scenario-to-Components* (VS2C) matrix; the *Component-to-Requirements* (C2R) matrix evaluates the percentage of the implemented requirements for each component. The VS2C matrix is reported in Appendix B and the C2R matrix is reported in Table 4.

Component	**Current Requirement coverage percentage**	**Deliverables (design and implementation)**
component:SLA Manager	99%	D1.4.1, D1.4.2
component:Service Manager	100%	D1.4.1, D1.4.2
component:Security Metrics Catalogue	100%	D1.4.1, D1.4.2
component:Interoperability Layer	100%	D1.4.1, D1.4.2
component:Auditing	78%	D4.2.2, D1.4.1, D1.4.2
component:User Management	56%	D4.2.2, D1.4.1, D1.4.2
component: Security token	100%	D4.2.2, D4.4.1, D4.4.2
component: Credential Manager	100%	D4.2.2, D4.4.1, D4.4.2
model: SLA machine readable format	100%	D2.2.2
model: SLA XML framework	100%	D1.4.1
model: SPECS data model	100%	D1.3, D1.4.1
component:Custom OS	~ 70%	D1.1.3, D1.6.1
component:Components Logging	~ 70%	D1.1.3, D1.6.1
component:NodeBootstrapper	~ 70%	D1.1.3, D1.6.1
component:Node Logging	~ 70%	D1.1.3, D1.6.1
component:Node discovery	~ 70%	D1.1.3, D1.6.1
component:Node controller	~ 70%	D1.1.3, D1.6.1
component:Component Discover Sys.	~ 70%	D1.1.3, D1.6.1
component:Components Controller	~ 70%	D1.1.3, D1.6.1
component:Artifact Repository	~ 70%	D1.1.3, D1.6.1
component:Component Operational REST API	~ 70%	D1.1.3, D1.6.1
component:Node Operational REST API	~ 70%	D1.1.3, D1.6.1
component:Cluster Manager	~ 70%	D1.1.3, D1.6.1

component:SLOManager	80%	D2.2.2, D2.3.1
component:SupplyChain	n/a	D2.2.2, D2.3.1
component:SecurityReasoner	80%	D2.2.2, D2.3.1
model:SLAConceptualModel	100%	D2.2.2, D2.3.1
model:SecurityMetricsCatalogue	100%	D2.2.2, D2.3.1
Components: Event Hub	100%	D3.3, D3.4.1
Components: Event Aggregator	~ 40%	D3.3, D3.4.1
Components: Event Archiver	~ 40%	D3.3, D3.4.1
Components: SLOM Exporter	~ 40%	D3.3, D3.4.1
Components: Monitoring Policy Filter	~ 40%	D3.3, D3.4.1
Components: Adapters	100%	D3.3, D3.4.1
model:Monipoli	100%	D3.3, D3.4.1
component:Planning	85%	D4.2.2, D4.3.2
component:Implementation	89%	D4.2.2, D4.3.2
component:Diagnosis	95%	D4.2.2, D4.3.2
component:RDS	100%	D4.2.2, D4.3.2
component:WebPool	80%	D4.2.2, D4.3.2
component:Broker	100%	D4.2.2, D4.3.2
component:DBB	100%	D4.2.2, D4.3.2
component:E2EE	100%	D4.2.2, D4.3.2
component:SVA	75%	D4.2.2, D4.3.2
component:TLS	100%	D4.2.2, D4.3.2

Table 4. C2R matrix

On the other hand, VAs are mapped onto VSs with the *Validation-Application-to-Validation-Scenarios* (VA2VS) matrix will be reported in Section 7. By composing VS2C, C2R and VA2VS matrices, the percentage of the verified requirements can be calculated.
In this context, the coverage analysis can be formalised by the definition of ten KPIs:

- SC_U: the percentage of the covered User Key Concern Items by specified VSs (*Specification-related Coverage of Users*);
- SC_{TS}: the percentage of the covered Target Services Key Concern Items by specified VSs (*Specification-related Coverage Target Services*);
- SC_{IC}: the percentage of the covered Invocation Chain Key Concern Items by specified VSs (*Specification-related Coverage of Invocation Chains*);
- SC_{SS}: the percentage of the covered SPECS Services Key Concern Items by specified VSs (*Specification-related Coverage of SPECS Services*);
- SC_{SLA}: the percentage of the covered SLA lifecycle Key Concern Items by specified VSs (*Specification-related Coverage of SLA lifecycle transitions*);
- EC_U: the percentage of the covered User Key Concern Items by executed VAs (*Execution-related Coverage of Users*);
- EC_{TS}: the percentage of the covered Target Services Key Concern Items by executed VAs (*Execution-related Coverage of Target Services*);

- EC_{IC}: the percentage of the covered Invocation Chain Key Concern Items by executed VAs (*Execution-related Coverage of Invocation Chains*);
- EC_{SS}: the percentage of the covered SPECS Services Key Concern Items by executed VAs (*Execution-related Coverage of SPECS Services*);
- EC_{SLA}: the percentage of the covered SLA lifecycle Key Concern Items by executed VAs (*Execution-related Coverage of SLA lifecycle transitions*).

6. Validation Applications

This section describes each of the VA listed in Section 3.3. For each VA, a brief discussion of its importance regarding covered items is also reported.

6.1. Web Container

In this application, the EU aims at acquiring one or more Virtual Machine to run his/her applications. He/she also wants to improve the security of the application: he/she is aware of some security-oriented mechanisms, but he/she is not an expert of security.

Using the application the EU can (1) select service and existing cloud provider; (2) add security capability to the service; (3) select the security controls and, for each control, select the metric to monitor. After this phases, the SLA is signed and the service is deployed: the monitoring phase starts.

This VA covers the greatest part of the VSs and solicits some components (e.g., TLS, SVA, WebPool) that otherwise will be not solicited.

The details on the architecture of this application, the solicited SPECS components and the VSs this application covers are reported in D5.1.3.

6.2. Metric Catalogue

Through a web interface the EU can manage a database that represents the catalogue of all the metrics available in SPECS. The user is guided by a software wizard among all the functionalities of this application. The functionalities are Create, Read, Update, Delete (CRUD) functionalities. Hence it is possible to add a new Security Metric; get and remove a Security Metric; update a Security Metric. In addition to these basic functionalities, it is also possible to update the entire database.

Even if this application does not find any correspondence neither in the solution portfolio nor in the SPECS User Stories, the Metric Catalogue represents a SPECS application that has been added to the set of the available applications since the services it offers are used from other SPECS applications.

The details on the architecture of this application, the solicited SPECS components and the VSs this application covers are reported in D5.1.3.

6.3. Security Reasoner

A cloud service customer (CSC), representing the EU of this user story, aims at acquiring a cloud service, which fulfils some security requirements. It is reasonable to suppose that the EU is not an expert in security field, but has specific security requirements. He aims at selecting and ranking SLAs according to their declared security controls and his/her security requirements. Without SPECS, the EU should manually compare the questionnaires by each CSP according to his/her own interests and competences. Security Reasoner can define a common and heterogeneous mean to analyse what the CSPs offer in an automatic way and according to the security goals expressed by the EU.

This application is important since it shows how available techniques can used in order to evaluate and rank the different SLA offers.

The details on the architecture of this application, the solicited SPECS components and the VSs this application covers will be reported in D2.3.3.

6.4. Secure Storage

The End-user aims at acquiring a Secure file storage service from the company private cloud system, represented by a Virtual NFS partition, which fulfils specific security requirements. In particular, the End-user requires the adoption of some hardware and software capabilities to protect the Storage service environment. To achieve this service, the End-user negotiates the desired features with SPECS.

The application is built using XLAB software solutions (for more details, please refer to D5.2.1 and D5.2.2).

6.5. ngDC

In this industrial application, SPECS will be used in a private cloud environment (IM2), in which the CSP would preserve its storage space about the end user requirement.

Considering that the CSP hosting SPECS has more control over its internal storage about storage hosted on an external provider if the End-user request could be satisfied through the external provider, SPECS will broker the storage on the external CSP. Otherwise it will choose the best internal storage resource that fits the End-user requirements.

Considering that a CSP has more control over its internal storage resources about storage hosted on an external provider, in a traditional data center, the CSP will attempt to provide resources that offer a "closest fit" solution to the End-user. In the SPECS solution, if a user request could be satisfied through the external provider, SPECS will broker the storage on the external CSP. Otherwise, it will choose the best internal storage resource that fits the End-user requirements.

The application is built using EMC storage hardware solutions and the ViPR software layer (for more detail please refer to D5.3).

6.6. AAA-as-a-Service

The goal of this task is the development of a set of applications, offered "–as-a-service", on the top of the SPECS platform, dedicated to Identity Management and Access Control. Thanks to the SPECS platform a security manager should be able to apply such security mechanisms on their services through simple service invocation, maintaining grants about the offered functionalities.

The application is built using EMC storage hardware solutions and the ViPR software layer (for more detail please refer to D5.4).

7. Coverage Analysis

This section presents the quantitative and qualitative analysis of the KPIs defined in Section 5. Also, the evolution during the project of the values related to such KPIs are reported. The specification-related KPIs (SC_U, SC_{TS}, SC_{IC}, SC_{SS}, and SC_{SLA}) are evaluated at Y1 and Y2. The execution-related KPIs (EC_U, EC_{TS}, EC_{IC}, EC_{SS}, and EC_{SLA}) are evaluated only at Y2 (since no execution data were available at Y1); nevertheless, a prevision of what expected by the end of the project (M30) is reported.

To evaluate the execution related KPIs it is necessary to introduce the VA2VS matrix, which is reported in Table 5.

	Web container	*Metric Catalogue*	*Security Reasoner*	*Secure Storage*	*ngDC*	*AAA-as-a-Service*
Secure_Storage_Selection			X	X		
Secure_Storage_Brokering_with_Client_Crypto				X		
Secure_Storage_with_Defined_CSP				X		
Secure_Storage_Brokering_with_Client_Crypto_alert				X		
Secure_Storage_Brokering_with_Client_Crypto_violation				X		
Secure_Web_Container_Selection	X		X			
Secure_Web_Container_Brokering	X					
Secure_Web_Container_TLS_enhanced	X					
Secure_Web_Container_SVA_enhanced_alert	X					
Secure_Web_Container_TLS_SVA_enhanced_violation	X					
Secure_Web_Container_TLS_multitenancy	X					
Secure_Web_Container_Web_Pool_Replication_enhanced_alert	X					
Secure_Web_Container_Web_Pool_Replication_enhanced_violation	X					
Data_Center_Bursting_for_Storage_Resources					X	
Data_Center_Storage_Selection			X		X	
Security_Tokens_Acquisition						X
Security_Tokens_Validation						X
Security_Tokens_Revocation						X
Credential_Management						X
User_Direct_Registration						X
User_Registration_External_Account						X
User_Authentication_External_Account						X
Metric_Definition		X				
Security_Mechanism_Development						
SPECS_Application_Development						

Table 5. VA2VS Matrix

Each of the following subsections covers one specific Key Concern.

7.1. User

D5.1.1 reported that at Y1 five users were uncovered by VSs: *U_2 – CSC:Integrator, U_3 – CSC:Inter-cloud Provider, U_4 – CSC:Developer, U_5 – CSC:Developer, U_6 – CSC:Developer.* This deliverable adds new VSs that focus on the developer: *Security_Mechanism_Development* and *SPECS_Application_Development* which cover *U_4 – CSC:Developer, U_5 – CSC:Developer* and *U_6 – CSC:Developer.* The values of the KPIs related to the User Key Concern are reported in Table 6.

	Y1	**Y2**	**M30**
SC_U	16.67%	66.67%	-
EC_U	-	16.67%	16.67%

Table 6. Values of KPIs related to the User Key Concern

7.2. Invocation Chain

The values of the KPIs related to the Invocation Chain Key Concern are reported in Table 7. The VSs cover all the Interaction Chains. Moreover, the VAs of Y2 only consider *IM1* while *IM2* and *IM3* will be executed in the context of the applications at M30.

	Y1	**Y2**	**M30**
SC_{IC}	66.67%	100%	-
EC_{IC}	-	33.33%	100%

Table 7. Values of KPIs related to the Invocation Chain Key Concern

7.3. Target Services

The values of the KPIs related to the Target Services Key Concern are reported in Table 8. D5.1.2 has not added any further VSs covering the Target Services not covered in D5.1.1. However, the definition of the web container VA, where an EU can populate his/her VMs with a generic application, allows us to extend the coverage of the VSs related to this VA also to the following Target Services: *TS_1 (Compute as a Service), TS_2 (Communications as a Service), TS_5 (Network as a Service), TS_6 (Platform as a Service) and TS_7 (Software as a Service)*.

For what concerns the execution, *TS_3 (Data Storage as a Service)* is covered by VSs related to the Secure Storage VA and. Hence, it will be covered at M30.

	Y1	**Y2**	**M30**
SC_{TS}	28.5%	100%	-
EC_{TS}	-	85.7%	100%

Table 8. Values of KPIs related to the Target Services Key Concern

7.4. SLA lifecycle

The values of the KPIs related to the SLA lifecycle Key Concern are reported in Table 8. D5.1.2 has not added any further VSs covering the transitions of the SLA lifecycle state machine not covered in D5.1.1. Furthermore, D1.1.3 modifies this model by deleting a transition covered in the D5.1.1 (the *SLA_2* transition).

The transitions that remain not covered are *SLA_15 (from Reacting to Terminating), SLA_16 (from Observed to Terminating), SLA_18 (from Renegotiating to Observed) and SLA_20 (from Renegotiating to Terminating)*. However, the absence of coverage for these transitions is a minor issue because all the states are covered and, then, passing to Terminating and/or Observed states has just been covered by some VSs.

For what concerns the execution, the web container and metric catalogue applications covers VSs specifying almost all the transitions except *SLA_3 (from Negotiating to Terminating)* and *SLA_4 (from Terminating to Terminate SLA)*. These last transitions will be executed at M30.

	Y1	**Y2**	**M30**
SC_{SLA}	80%	78.9%[2]	-
EC_{SLA}	-	73.6%	78.9%

Table 9. Values of KPIs related to the SLA lifecycle Key Concern

7.5. SPECS Services

The number of the total SPECS requirements is 293: 70 are related to the platform and 223 to the modules. Table 10 reports the values of the KPIs related to SPECS Services. At Y1, the percentage of the requirements (restricted to the modules) was evaluated to be 86%. To these requirements all the platform requirements are to add. All the platform requirements are considered covered since they manage basic functionalities: all the VSs use these basic functions because they are necessary to run modules and security mechanisms. Hence, the real coverage of the requirements at Y1 is almost 100% and this value had not changed during Y2.

For what concerns the execution, the number of the requirements related to components solicited by the web container and the metric catalogue VAs is 170 which brings the EC_{SS} to about 58%. At M30, the other VAs would allow us to execute all the VSs and, hence, to cover all the requirements.

	Y1	**Y2**	**M30**
SC_{SS}	86%	100%	-
EC_{SS}	-	58%	100%

Table 10. Values of KPIs related to the SPECS Services Key Concern

Moreover, the extra testing effort that will be spent in both integration and unit level testing will guarantee the total coverage of all the functional and non-functional requirements.

[2] The decrease of this value from Y1 to Y2 is due to the deletion of the SLA_2 transition.

8. Conclusions

This deliverable improves the D51.1 by:

- the definition of VAs, their framing into the testing approach and their mapping to VSs;
- the description of the six VAs available for SPECS;
- the refinement of the VSs;
- an improvement in the coverage level of the Key Concern Items.

D5.1.1 ended with some recommendations for this deliverable:

1. spanning validation scenarios on the four defined user stories in a more uniform way;
2. improving the overall coverage;
3. choosing the VSs that will be effectively executed;
4. improving the number and refining the grain of graphical models;
5. improving the description of the cross-cutting scenario;
6. further details the description of the Validation Scenarios to be executed.

These recommendations have been addressed in this deliverable as follows:

1. this deliverable has reorganized the VSs, also by moving some contents from/to VSs, distributed the VSs onto VAs as in Table 5;
2. the improvement of the Key Concern coverage level has been demonstrated in Section 7;
3. the VSs to execute are shown by the VA2VS matrix;
4. graphical descriptions of the scenarios are considered a minor point and hence, there are still some VSs which have not a graphical description;
5. the description of these VSs have been improved;
6. all the VSs have been detailed.

9. Bibliography

[1] "ISO/IEC DIS-17788, Information technology — Cloud computing — Overview and Vocabulary" (draft), International Organization for Standardization. Tech. Rep. ISOIEC 17788:2014, 2014.

[2] K. Pohl. "Requirements engineering: fundamentals, principles, and techniques." Springer Publishing Company, Incorporated, 2010.

Appendix A – List of the Key Concerns Items

This Appendix reports the list of all the Key Concern Items as they are reported also in D5.1.1. Users, Target Services and Invocation Chains are unchanged with respect to the Y1 and are here reported to make the document self-readable. SLA Lifecycle has changed and is here reported (see D1.1.3). SPECS Services are reported in the Appendix B.

Users

With the aim of validating the SPECS framework from a functional point of view, we considered the sub-roles and their activities identified in D1.1.1, namely:

U_1. CSC:user - use of the selected target services;
U_2. CSC:integrator - connect ICT systems to cloud services: integration of the target services into the SPECS applications and the developed SPECS Security services;
U_3. CSP:InterCloud Provider - perform peering, federation, intermediation, aggregation and arbitrage;
U_4. CSN:developer - design, create and maintain service components: the creation of new SPECS Security services used by the SPECS applications;
U_5. CSN:developer - compose services: use of the framework services (SLA platform, negotiation, monitoring, enforcement, etc..) to create new SPECS Security services;
U_6. CSN:developer - test services (with respect to developed SPECS Security services).

Invocation chains

The set of *invocation chains* comes out from the analysis of D1.2, where they have been defined starting from the *interaction models*:

IM1. Interaction model 1: the SPECS services run as an independent third party, consuming resources acquired from a public or private cloud provider and managed by the SPECS Owner offering its services to End-users.
IM2. Interaction model 2: the SPECS services are co-located within a hosting CSP, which internally hosts the SPECS Platform and the target service.
IM3. Interaction model 3: the SPECS services are dedicated to a single End-user, who installs and runs them to manage her/his own activities.

Target services

The third Key Concern identified while the analysis of the SPECS framework is represented by the *target service*. In particular, according to [1], the following cloud categories are considered:

TS_1. Compute as a Service;
TS_2. Communications as a Service;
TS_3. Data Storage as a Service;
TS_4. Infrastructure as a Service;
TS_5. Network as a Service;
TS_6. Platform as a Service;
TS_7. Software as a Service.

SLA lifecycle

In the following, we list the key concern items, represented by the set of transitions of the SLA lifecycle state machine in its redefined form as presented in D1.1.3:

SLA_1. Initial-Pending;
SLA_2. *deleted in the refined version* (formerly Pending-Rejected);
SLA_3. Pending-Negotiating;

SLA_4. Negotiating -Terminating;
SLA_5. Terminating-Terminate SLA;
SLA_6. Negotiating-Signed;
SLA_7. Signed-Observed;
SLA_8. Observed-SLA Completed;
SLA_9. Observed-Alerted;
SLA_10. Alerted -Proactive redressing;
SLA_11. Reacting-Observed;
SLA_12. Alerted-Violated;
SLA_13. Observed-Violated;
SLA_14. Violated-Remediating;
SLA_15. Reacting-Terminating;
SLA_16. Observed-Terminating;
SLA_17. Reacting-Renegotiating;
SLA_18. Renegotiating-Observed;
SLA_19. Renegotiating-Signed;
SLA_20. Renegotiating-Terminating.

The SLA lifecycle is shown in Figure 7.

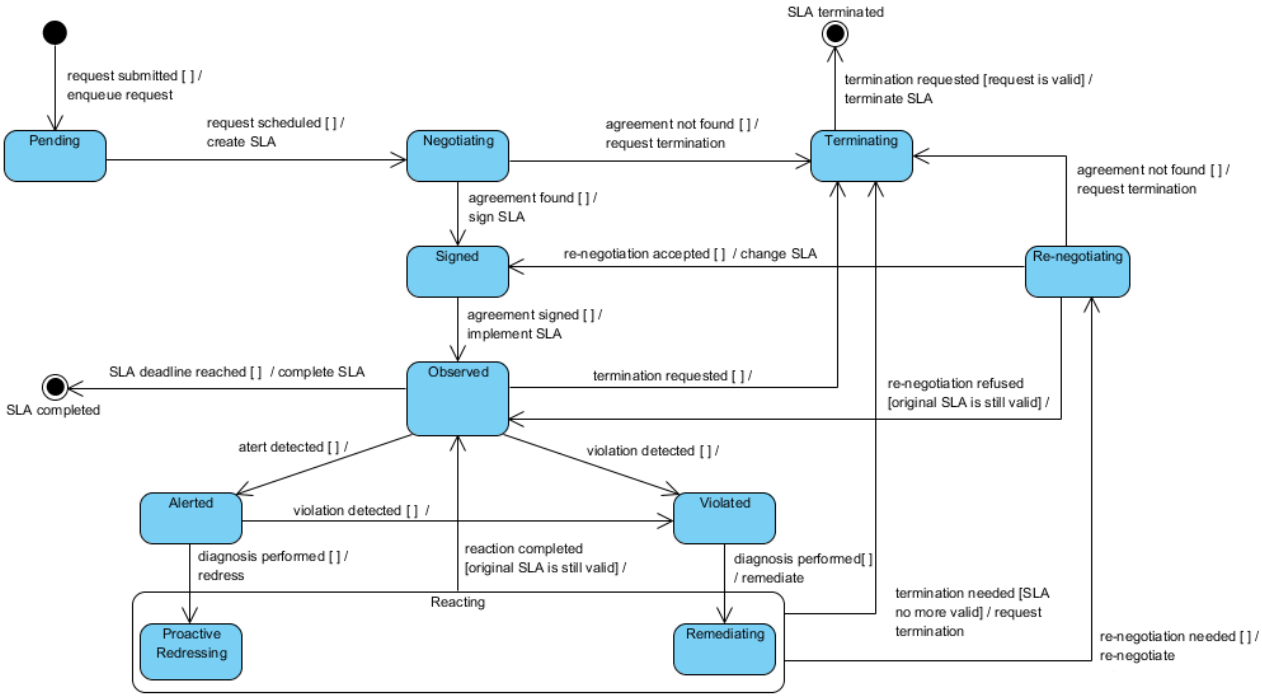


Figure 7. Refined SLA lifecycle state machine model (D1.1.3)

SPECS services

The full list of the functional and non-functional requirements of the SPECS Platform and modules is reported in the Appendix B.

Appendix B – Traceability Mappings

This appendix is provided as a separate annex (Annex_A). It is consists in two sheets. The first, *VS2C*, gives the mapping between the defined VSs and the SPECS components. The second sheet, *C2R detailed*, reports the full details on how the SPECS components implements the SPECS requirements.