Secure Provisioning of Cloud Services
based on SLA Management

SPECS Project - Deliverable 2.2.2

Report on conceptual framework for Cloud SLA negotiation - Final

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.:	D 2.2.2
Deliverable title:	Report on conceptual framework for Cloud SLA negotiation - Final
Deliverable nature:	Report
Dissemination level:	Public
Contractual delivery:	31 October 2015
Actual delivery date:	31 October 2015
Author(s):	Rubén Trapero (TUDA), Ahmed Taha (TUDA)
Contributors:	Valentina Casola (CeRICT), Massimiliano Rak (CeRICT), Alessandra De Benedictis (CeRICT), Marina Bregu (CSA)
Reviewers:	Jolanda Modic (XLAB), Madalina Erascu (IeAT), Alain Pannetrat (CSA)
Task contributing to the deliverable:	T2.2
Total number of pages:	70

Executive summary

This deliverable is the second of two deliverables (D2.2.1, D2.2.2) that presents the **negotiation module and the final description of the contributed techniques to reason about cloud SLAs**.

At month M12 the document D2.2.1 presented:

- A conceptual model to represent SLAs
- A high level architecture to negotiate SLAs
- Negotiation and renegotiation processes
- Algorithms (REM and QHP) to reason about cloud SLAs.

The final version of the document (D2.2.2) presents:

- An updated conceptual model to represent SLAs compliant with the latest outcomes of standards and working groups.
- A metric catalogue compliant with the conceptual model and enriched with the feedback received from the Platform (WP1), the Enforcement module (WP4), and the validation scenarios (WP5).
- A refined architecture: the main negotiation components are the same as the ones created in Y1. However, the interaction with external modules has been refined and also the information exchanged among them.
- A refined negotiation process: thanks to the feedback received from the Enforcement module and from the T2.3 we have redefined most of the negotiation process. The refined process is compliant with the SPECS application methodology that is used to gather requirements from the End-users (EUs). The new negotiation process also takes into account the consolidated approach to create supply chains that is orchestrated by the Enforcement module.
- A refined renegotiation process: the renegotiation processes has been completely redefined comparing with the simplified process created in M12. To do so we have received a continuous feedback from the developers of the Enforcement module which oversees the remediation processes that triggers the renegotiation.
- New aspects related to the security assessment methodologies. In Y1 we presented two methodologies to evaluate cloud service providers with respect to EUs' requirements: REM and QHP. In D2.2.2 we provide details about how REM has been used to evaluate the SLA Model of SPECS. We also provide an evolution of the QHP methodology that considers uncertainty on qualitative EUs' requirements by using quantification of fuzzy numbers.

Table of contents

Index of figures

Index of tables

1. Introduction

The process of SLA negotiation has evolved during the second year of SPECS.

This document describes the final outcomes regarding the Negotiation module developed in SPECS. The current content of the deliverable updates the information presented in D2.2.1 by adding updated information, changes in the design, and improvements to the techniques and methods introduced in the first year. The feedback received from other tasks and from the implementation activities carried out during the second year has also helped to refine the main aspects of the negotiation and renegotiation process. To this end, the updated set of requirements inherited from WP1 and WP4 and especially from the deliverable D2.1.2 (submitted at M12) is also considered in the refinement of the designs and processes presented and discussed here

The current deliverable includes the final format used to represent SLAs. The architecture of the Negotiation module is also presented, emphasizing the changes with respect to the architecture introduced in the first year. Changes to the SLA format and the design of the Negotiation module are mostly due to the implementation activities in T2.3 and the feedback received from other WPs (namely WP1, WP4, and WP5).

Security reasoning techniques are also validated and improved in this deliverable. The usage of security assessment techniques under the SPECS framework is presented, as well as a novel security assessment technique that add the management of the uncertainty of end-users' requirements.

The document is structured as follows. Section 2 provides information about the relationship between this document and the rest of the deliverables. Section 3 provides the summary of the negotiation requirements. The organization of requirements is the basis for the structure of the rest of the sections in the document. Section 4 details the final version of the SLA specification, including the conceptual model to design security SLAs, and the latest version of the machine readable format that relies on the designed conceptual model. A complete metric catalogue is also presented in Section 4, which comprises the current set of security metrics used in SPECS. Section 5 provides an overview of the Negotiation architecture. This is a high level presentation of the Negotiation architecture that helps to better understand the negotiation and renegotiation processes introduced in Sections 5 and 6, respectively. A detailed low level description of the Negotiation module and its corresponding components is reported in T2.3. Section 8 details the security reasoners considered in SPECS. This includes the description of how one of them has been integrated into the negotiation processes and an evolution of the other that uses fuzzy variables to manage End-users´ requirements uncertainty. The document concludes with a short summary.

2. Relationship with other deliverables

The Figure 1 depicts the relationship between D2.2.2 with respect and the rest of deliverables of SPECS.

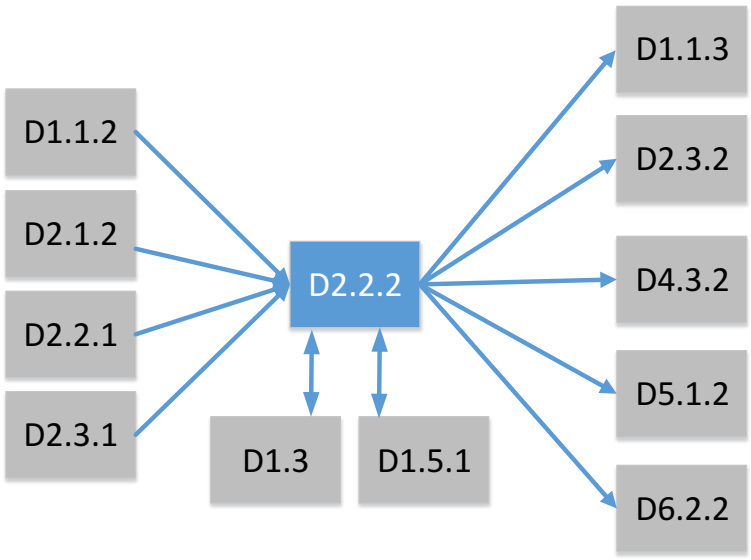

Figure 1. Relationship with other deliverables

There are three groups of deliverables: the ones that are input for D2.2.2, deliverables that use D2.2.2 as the input, and then there are deliverables that present both, input and output.

The following is a summary of the relationships:

- Deliverables used as an input for D2.2.2:
  - D1.1.2 provides the general overview of the SPECS architecture.
  - D2.1.2 provides the final set of requirements compiled for the Negotiation module.
  - D2.3.1 provides the initial feedback from the implementation activities related to the Negotiation module.
- Deliverables that use D2.2.2 as the input:
  - D1.1.3 will provide the intermediate overview of the SPECS architecture, including also the latest design of the Negotiation module.
  - D2.3.2 will provide the second prototype of the Negotiation module and will depend on the outcomes of the D2.2.2.
  - D4.3.2 will provide the second iteration of the Enforcement implementation and will include the outcomes of the D2.2.2, especially in what regards the generation of supply chains and renegotiation processes.
  - D5.1.2 provides validation scenarios that are also based on the negotiation and renegotiation processes reported in D2.2.2.
  - D6.2.2 will discuss the metrics catalogue reported in D2.2.2 as part of the standardization activities in WP6.
- Deliverables that use D2.2.2 as input and output:
  - D1.3 provides the interfaces among all SPECS modules.
  - D1.5 provides the integration details of SPECS.

3. Overview of requirements

This section describes all requirements that have been covered by SPECS Platform and regard the SPECS Negotiation module. Here is included the updated list of requirements. All active requirements were selected from the ones being obsolete, superseded or rejected after the analysis of implemented SPECS applications. Few new requirements have emerged which are also included. The rest of them are either the same (as presented in D2.1.2) or updated.
Two main sources were considered in the process of eliciting requirements affecting the negotiation processes: requirements elicited in deliverable D1.2 and in D2.1.2. However, other tasks belonging to other WPs (Platform and Enforcement) were also considered.
The requirements related to the Negotiation module were filtered and selected. They were grouped by common functionalities in order to provide an initial overview of the main functional blocks required by the Negotiation module. With the result of this analysis the following list of activities have been identified:

Activity 1. SLA conceptual model requirements: specification of a conceptual model for defining SLAs.
Activity 2. Architecture requirements: Creation of the initial architecture for the Negotiation module.
Activity 3. Negotiation process requirements: Creation of the process for the negotiation of SLAs.
Activity 4. Renegotiation process requirements
Activity 5. Security reasoning requirements: Creation of an initial approach for evaluating the security of a cloud service with respect to the CSC's security requirements.

The result of the elicitation of the requirements according to the aforementioned list of activities is as follows:

3.1 SLA conceptual model requirements

This activity includes the definition of what a security SLA is, the information that it should contain and the format chosen to represent the information. The following requirements (Table1) are covered by the specification of the SLA described in Section 4:

REQ_ID	Requirement	Description	Comment
SLANEG_R1	***SLA language should support specification of required cloud resources***	SLA language should support specification of required IaaS, PaaS or SaaS resources and mapping between SLA terms and low-level CSP resources.	Has remained the same
SLANEG_R2	***SLA language should support simple composition***	SLA language should support the combination of two SLAs in order to model a supply chain built between SPECS and a CSP.	Has remained the same
SLANEG_R3	***The negotiation process should support composite cloud services***	In analogy to SLANEG_R2, also the SLA evaluation technique should support the notion of composition (cloud supply chain with two or more SLA's).	Has remained the same
SLANEG_R4	***Negotiated SLOs should be***	This is the basic requirement to build the (automated)	Has been updated

	monitorable and enforceable	management of cloud SLAs in WP3 and WP4.	
SLANEG_R6	***Evidence associated with measured SLOs***	Customers might need to be provided with some sort of evidence related with the implementation of a specific SLO, in order to make an informed decision while negotiating a cloud SLA in SPECS. This evidence might come in the form of e.g., the associated Security control's implementation as documented in the applicable security certification (e.g., CSA OCF or ISO/IEC 27002).	Has remained the same
SLANEG_R8	***Specification of customer's security requirements***	Not all customers are security experts; therefore their security requirements (input of the negotiation process) might come in different levels of granularity, based on the SPECS security SLA hierarchy (i.e., from Control Categories to Metrics/Measurements).	Has been updated
SLANEG_R9	***Reasoning about security SLOs in cloud SLA***	A typical SLA might contain several security related SLOs, which might be cumbersome to negotiate one by one. The negotiation mechanism should provide the techniques to reason about aggregated sets of security SLOs (e.g., computing the overall effect of a composed set of individual SLOs).	Has remained the same
SLANEG_R10	***Follow standards and industrial-accepted practices***	The different elements of the negotiation process (e.g., security SLOs) should follow as much as possible both relevant standards and best practices from the industrial domain. This requirement guarantees the interoperability and adoption of the expected results.	Has been updated

SLANEG_R11	***Mapping the user's security requirements to the CSP's offered SLOs***	Despite the level of granularity utilized to specify the CSC's requirements (cf., SLANEG_R8), it is necessary to provide a mapping to the actual SLOs that can be offered by the CSP.	Has been updated
SLANEG_R12	***Adoption of a conceptual model for security SLOs***	In order to promote interoperability, the security SLOs being used in SPECS should be associated with a standardized model that describes in further detail their associated elements e.g., metrics and measurements.	Has been updated (has superseded old requirements SLANEG_R14, R15 and R17)
SLANEG_R16	***Only measurable security SLO's can be negotiated***	In order to be negotiated within SPECS, the security SLO's should be measurable (i.e., associated with one or more metrics). This feature allows for comparing the user security requirements, with respect to each one of the offered cloud service configurations.	Has been updated
SLANEG_R18	***Management of Alerts on agreed SLA's***	The SLA conceptual model does and should provide support for the management of alerts (e.g., through the definition of the corresponding thresholds), both to Monitoring and Enforcement.	New requirement
SLANEG_R19	***SLO representation using a machine-readable SLA specification***	The selected SLA machine-readable specification should support both SLO-independent, and SLO-dependent representations (cf., Section 4.2, D2.1.2).	New requirement
SLANEG_R20	***Security metrics might have quantitative or qualitative values.***	The SLO included in an SLA may include both quantitative and qualitative security attributes, as a consequence, security metrics should cope with either quantitative or qualitative values.	Has remained the same

Table 1. Requirement related with the definition of the format of the SLA

3.2 Architecture requirements

This activity deals with the creation of the architecture of the Negotiation module. This includes the definition of the main functional blocks, the initial communication among them and the information

exchanged. The following requirements (Table 3) are covered by the design of the architecture described in Section 6.

REQ_ID	Requirement	Description	Comment
SLANEG_R3	***The negotiation process should support composite cloud services***	In analogy to SLANEG_R2, also the SLA evaluation technique should support the notion of composition (cloud supply chain with two or more SLA's).	Has remained the same
SLANEG_R4	***Negotiated SLOs should be monitorable and enforceable***	This is the basic requirement to build the (automated) management of cloud SLAs in WP3 and WP4.	Has been updated
SLANEG_R7	***Interactive and customer centric process***	SPECs negotiation process is both interactive and customer-centric: it is started and finalized by the customer (e.g., evaluated different SLAs until an agreement was reached with the CSP). Notice that this requirement does not apply to SPECS' re-negotiation, which will be further analysed in D2.1.2	Has remained the same
SLANEG_R13	***Security SLO should be measurable in the real-world***	The set of security SLO's to be considered by SPECS should be feasible to assess/measure in real-world cloud deployments.	Has remained the same
SLANEG_R16	***Only measurable security SLO's can be negotiated***	In order to be negotiated within SPECS, the security SLO's should be measurable (i.e., associated with one or more metrics). This feature allows for comparing the user security requirements, with respect to each one of the offered cloud service configurations.	Has been updated
SLANEG_R18	***Management of Alerts on agreed SLA's***	The SLA conceptual model does and should provide support for the management of alerts (e.g., through the definition of the corresponding thresholds), both to Monitoring and Enforcement.	New requirement
SLANEG_R19	***SLO representation using a machine-***	The selected SLA machine-readable specification should support both SLO-	New requirement

	readable SLA specification	independent, and SLO-dependent representations (cf., Section 4.2, D2.1.2).	
ENF_PLAN_R3	***Define security mechanisms related to SLOs***	The Planning component must be able to determine which kind of security mechanisms are to be applied, given a set of high-level SLOs contained in the SLA to implement.	Has remained the same
ENF_PLAN_R14	***Validate an SLA***	The Planning component has to be able to validate an SLA by verifying that it can be enforced.	New requirement (Covers the discarded requirement ENF_PLAN_R1).

Table 2. Requirements related to the architecture of Negotiation

3.3 Negotiation process requirements

The negotiation process includes the complete dialog among entities in the processes of reach agreement on a set of SLOs that are part of an SLA. It includes the steps required from triggering a negotiation to the signing (or not) of an SLA. The following requirements (Table 2) are covered by the design of the negotiation processes described in Section 6.

REQ_ID	Requirement	Description	Comment
SLANEG_R3	***The negotiation process should support composite cloud services***	The supply chain SPECS+CSP, might involve composing two-offered cloud SLAs for the negotiation process. The negotiation process should have a rich enough specification that will allow for the definition of the interdependencies between the constituent components of composite cloud services.	Has remained the same
SLANEG_R7	***Interactive and customer centric process***	SPECs negotiation process is both interactive and customer-centric: it is started and finalized by the customer (e.g., evaluated different SLAs until an agreement was reached with the CSP). Notice that this requirement does not apply to SPECS' re-negotiation, which will be further analysed in D2.1.2	Has remained the same
SLANEG_R34	***Representing security requirements of non-expert users***	The negotiation module should provide non-expert users with an easy way to express their security requirements (not necessarily through individual SLO's). For example, the	New Requirement

		approach followed by NIST on its Cloud Adapted Risk Management Framework should be further explored.	
SLAPL_R14	***Search CSP SLA***	The SPECS SLA Platform must allow other SPECS components to search for a set of SLAs in the CSP SLA repository by specifying a set of search criteria	Has remained the same
SLAPL_ R21	***Get SLA***	The SPECS SLA Platform must allow other SPECS components to get the reference to a SPECS SLA contained in the SPECS SLA repository.	Has remained the same
SLAPL_R10	***Get CSP SLA***	The SPECS SLA Platform must allow other SPECS components to get information (e.g. the granted parameters) about a CSP's SLA stored in the CSP SLA repository by specifying an ID for the SLA (obtained for example by performing a search operation).	Has remained the same
SLAPL_R23	***Search SLA***	The SPECS SLA Platform must allow other SPECS components to search for (i.e. obtain the ID of) a set of SPECS SLAs in the SPECS SLA repository, by specifying a set of search criteria.	Has remained the same
SLAPL_R24	***Check SLA***	The SPECS SLA Platform must allow other SPECS components to check the formal validity (e.g., formatting, digital signature expiration etc.) of an SLA contained either in the SPECS SLA repository or in the CSP SLA repository.	Has remained the same
SLANEG_R23	***Output of a successful negotiation process***	The result of a successful negotiation process is a well-formed SPECS security SLA hierarchy with the metrics values agreed with the End-user/customer. The SLA is then signed and stored in the SLA repository.	Has remained the same
SLAPL_R27	***Create SLA***	The SPECS SLA Platform must allow other SPECS components to create a SPECS SLA document.	Has remained the same

SLAPL_R33	***Sign SLA***	The SPECS SLA Platform must allow the SPECS administrator and the CSC to sign a SPECS SLA.	Has remained the same
SLANEG_R32	***Platform repositories***	The negotiation process needs to extract information from the following Platform repositories: Information about which services/mechanisms are allowed for a particular End-user. SLA's offered by SPECS Partner CSP's. Templates of supported SLA's. Templates might be based on different standards (ISO/IEC 19086) and best practices (CSA CCM/CAIQ). SLO service capability offerings. SLA's negotiated with End-users through SPECS.	New requirement
SLANEG_R33	***SLA Management***	The following SLA management operations should be supported by the Platform: • Search CSP in repository. • Search component for a given SLO. • Validate supply chain. • Sign new SLA. • Create new SLA. • Search the CSP's SLA repository for user matching CSP SLO's • Create a new SLA template End-user-CSP negotiation and agreement on the (amended) SLA template.	New Requirement

Table 3. Requirements related with the Negotiation Protocol

3.4 Renegotiation process requirements

Several situations will trigger the invalidity of a current signed SLA. Such are SLA violations or changes in the customer´s requirements. This will entail the renegotiation of new SLAs. The following table are the requirements associated to the renegotiation process.

REQ_ID	Requirement	Description	Comment
SLANEG_R25	***Renegotiation triggered by CSP or the End-user***	The generic case for security renegotiation corresponds to the CSP/End-user changing the conditions applicable to	New requirement

		its service, or the original set of security requirements respectively.	
SLANEG_R26	***Input for renegotiation***	Similar to negotiation, the renegotiation process starts with the set of new/changed security requirements that resulted on the violation/alert of the original SLA. These new/changed security requirements should be managed by SPECS in the same way that the originally negotiated requirements.	New requirement
SLANEG_R27	***Output of a successful renegotiation***	Please refer to SLA_NEG_R22	New requirement
SLAPL_R34	***Change SLA***	The SPECS SLA Platform must allow the SPECS administrator to update the content of a SPECS SLA after re-negotiation.	Has remained the same
SLAPL_R35	***Generate alert***	The SPECS SLA Platform must allow other SPECS components (belonging to the Monitoring module) to generate an alert to warn about a possible incoming SPECS SLA violation.	Has remained the same
SLAPL_R36	***Detect violation***	The SPECS SLA Platform must allow other SPECS components (belonging to the Monitoring module) to detect a SPECS SLA violation when the guaranteed requirements are no longer fulfilled.	Has remained the same

Table 4. Requirements related to the Renegotiation process

3.5 Security Reasoning requirements

This activity provides the basis for the security evaluation approaches. They allow to reason about security information, taking as input both security requirements and security guarantees provided by security components and services from external CSPs. With such security assessment some decision tools can be provided to CSCs, such as ranking or dashboards. The following requirements are covered by the design of the security assessment mechanisms described in Section 8.

REQ_ID	Requirement	Description	Comment
SLANEG_R5	***Support the evaluation***	Customers negotiating	Has remained

	of trade-offs	security SLOs through SPECS, should be made aware of the trade-offs possibly involving non-security related SLOs (e.g., response time).	the same
SLANEG_R6	***Evidence associated with measured SLOs***	Customers might need to be provided with some sort of evidence related with the implementation of a specific SLO, in order to make an informed decision while negotiating a cloud SLA in SPECS. This evidence might come in the form of e.g., the associated Security control's implementation as documented in the applicable security certification (e.g., CSA OCF or ISO/IEC 27002).	Has remained the same
SLANEG_R8	***Specification of customer's security requirements***	Not all customers are security experts; therefore their security requirements (input of the negotiation process) might come in different levels of granularity, based on the SPECS security SLA hierarchy (i.e., from Control Categories to Metrics/Measurements).	Has been updated
SLANEG_R9	***Reasoning about security SLOs in cloud SLA***	A typical SLA might contain several security related SLOs, which might be cumbersome to negotiate one by one. The negotiation mechanism should provide the techniques to reason about aggregated sets of security SLOs (e.g., computing the overall effect of a composed set of individual SLOs).	Has remained the same
SLANEG_R12	***Adoption of a conceptual model for security SLOs***	In order to promote interoperability, the security SLOs being used in SPECS should be associated with a standardized model	Has been updated (has superseded old requirements SLANEG_R14, R15

		that describes in further detail their associated elements e.g., metrics and measurements.	and R17)
SLANEG_R34	***Representing security requirements of non-expert users***	The negotiation module should provide non-expert users with an easy way to express their security requirements (not necessarily through individual SLO's). For example, the approach followed by NIST on its Cloud Adapted Risk Management Framework should be further explored.	New Requirement
SLANEG_R21	***Ordered values for security metrics.***	All possible values (quantitative or qualitative) associated with a security metric maintain an order relationship between them. For example: $SecMetric = \{v_1, v_2 \cdots v_n\}$ where: $\{v_1 < v_2 < \cdots < v_n\}$ And "<" denotes the order relationship.	Has remained the same
SLANEG_R22	***Security metrics operators***	Security metrics values can be specified through any of the following: • Binary operators "<, =, >, ≤, ≥" • Logical operators "AND, OR, NOT" • Intervals e.g. (512 bits < Encryption Key Size < 2048 bits) including temporal conditions e.g. (Hourly backups from 8:00 hrs.to 21:00 hrs.)	Has remained the same
SLANEG_R28	***Human-assessment of security metrics***	At the state of practice, it is common to find security metrics that are assessed through human intervention e.g., by auditors verifying the CSP's security documentation and	New Requirement

		policies. These security metrics should be also considered during the SPECS SLA life cycle, and in particular in the planning of the monitoring systems/monitoring policy to activate.	
SLANEG_R29	***Uncertainty/assurance of performed measurements***	The security metrics negotiated within SPECS can be assessed/measured through different means (e.g., software sensors, documented policies) and actors (software agents, auditors). Given this wide variety of possibilities, we can expect that the resulting/measured values can be associated with different levels of uncertainty/assurance. This requirement might be important for both monitoring and enforcement.	New Requirement
SLANEG_R20	***Security metrics might have quantitative or qualitative values***	The SLO included in an SLA may include both quantitative and qualitative security attributes, as a consequence, security metrics should cope with either quantitative or qualitative values.	Has remained the same
SLANEG_R30	***Remediation through SLA renegotiation***	Enforcement should consider the renegotiation of an existing SLA as a potential remedy to apply in case of alerts and violations.	New Requirement
SLANEG_R31	***Alerts/violations affecting multiple elements of the secure SLA hierarchy***	A detected alert/violation might affect more than one element of the SPECS security SLA hierarchy. Enforcement should consider interrelationships along SLA elements to choose the optimal redressing technique (e.g., renegotiation might help to	New Requirement

		manage multiple alerts/violations).	
ENF_PLAN_R3	***Define security mechanisms related to SLOs***	The Planning component must be able to determine which kind of security mechanisms are to be applied, given a set of high-level SLOs contained in the SLA to implement.	Has been updated
ENF_PLAN_R4	***Get security components***	The Planning component must be able to retrieve the available Enforcement security components that implement the security mechanisms related to the fulfilment of the SLOs defined in the SLA to implement.	Has been updated
ENF_PLAN_R5	***Select best security component***	Based on the selected target service and on the negotiated SLA, the Planning component must be able to select the best available Enforcement components to invoke, among different technology stacks, in order to meet the SLOs defined in the SLA.	Has been updated

Table 5. Requirements related to the evaluation of security

4. Security Level Agreement specification

4.1 SLA conceptual model

In D2.2.1, we introduced the SPECS security SLA hierarchy, specifying the main elements relevant to a security SLA, namely *control categories*, *controls*, *service level objectives* and *security metrics*, along with their interrelationships. The proposed conceptual model, shown in Figure 2, reported the main attributes of the introduced concepts and put in evidence how such concepts are related to one another.

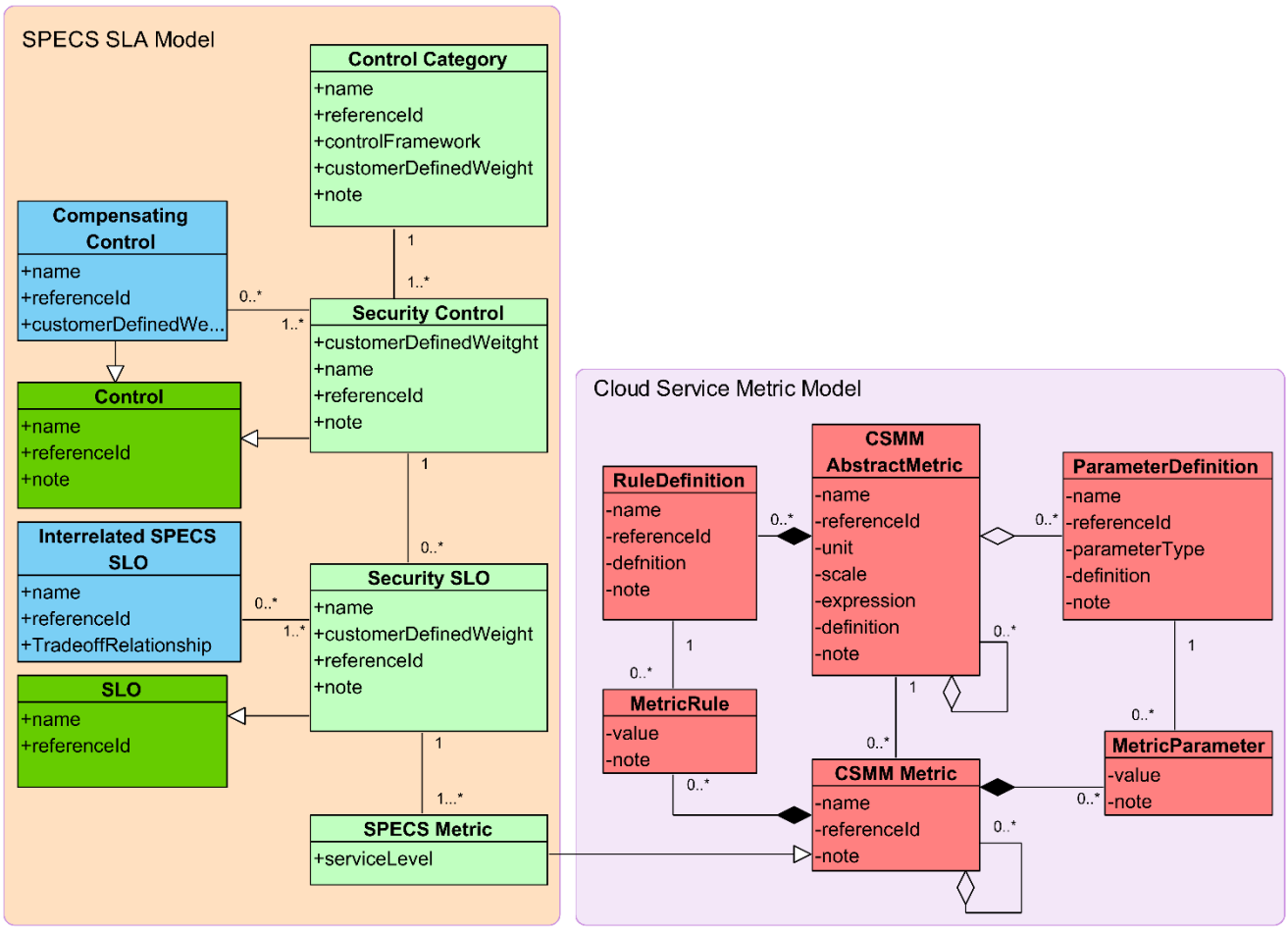

Figure 2. SPECS security SLA conceptual mode proposed in deliverable 2.2.1

In Figure 3, we report the final SPECS SLA conceptual model, based on the one introduced in D2.2.1. We represent an evolution of it in that it is more oriented to the actual implementation of security SLAs. Indeed, while the original model was more suited for the evaluation of the security level delivered with a service, the updated model allows for the specification of all the concepts needed not only for security assessment but also for the automatic negotiation, enforcement and monitoring of security features on top of cloud services.

As shown, an SLA (referred to as Security SLA in the figure) is characterized by several attributes related to the negotiation process itself (such as the agreement initiator and responder) and it declares a specific SLA Template on which it is based. Indeed, as explained in details in Section 5, negotiation is based on templates. Templates represent the set of negotiable features that can be included in an SLA. In SPECS they are built by the SPECS Owner and include the set of all security features that it is willing to offer, through the SPECS Application, to the

SPECS Customers.

As depicted, an SLA is basically composed of three main security-related concepts: security capabilities, security metrics and SLOs.

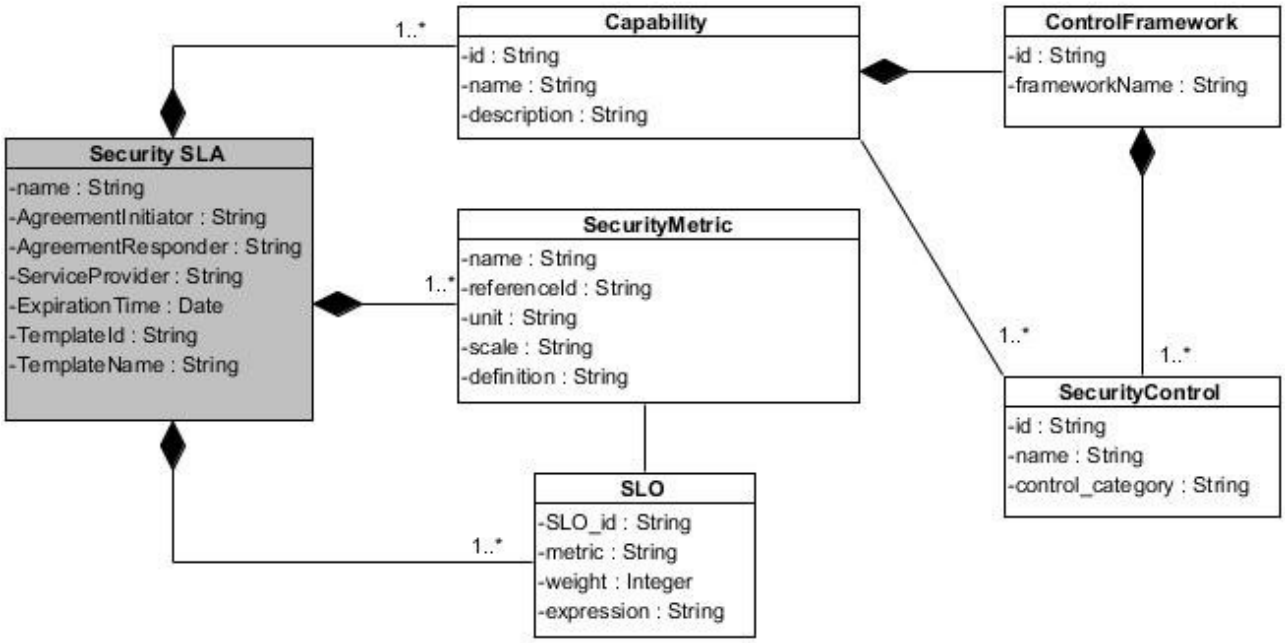

Figure 3. Refined SPECS security SLA conceptual model

Note that the security-related concepts introduced in the first version of the model are still valid in the updated version, even if some slight changes have been made in order to better reflect the way they are actually considered and implemented in SPECS.

First of all, we simplified the concept of *control*, for which the first version explicitly reported a distinction between *security controls* and *compensating controls*. In SPECS, we only consider the concept of "security controls", belonging to specific control categories of a chosen control framework (e.g., CSA's CCM [18] or NIST's control framework [15]) and representing the "building blocks" of security *capabilities*[1]. An End-user can require the activation of proper security capabilities (among those available in the template), to which specific security mechanisms provided by the Enforcement module are mapped. The controls that build such capabilities may be either security controls or related compensating controls that SPECS is able to enforce to fulfil the End-user's requests.

Furthermore, the refined conceptual model does not explicitly consider *interrelated SLOs* anymore, as the dependencies among SLOs are captured by the dependencies among the security metrics on top of which the SLOs are built, and these are managed at the template level. Finally, we updated the association between SLOs and metrics to be compliant with the WS-Agreement standard, which adopts the concept of *variable* to build SLOs depending on specific metrics. In the refined model, each SLO is based on a variable that refers to one of the security metrics reported in the service description term section. Security metrics are still represented

[1] Security capabilities are defined by NIST as combinations of mutually-reinforcing security controls (i.e., safeguards and countermeasures) implemented by technical means (i.e., functionality in hardware, software, and firmware), physical means (i.e., physical devices and protective measures), and procedural means (i.e., procedures performed by individuals) [14]

as reported in Figure 2, based on the RATAX specification; we did not include the related schema for clarity's sake.

In the next section, we provide more details on the machine-readable format based on the discussed refined model.

4.2 SLA machine readable representation

WS-Agreement (WSAG), born in the context of GRID computing, is currently the only standard supporting both a formal representation of SLAs and a protocol for their automation, and has been recently widely adopted, in the context of many Cloud-oriented FP7 projects (e.g., Contrail, mOSAIC, Optimis, Paasage), to represent SLAs in the Cloud environment. However, WS-Agreement does not allow, by its original definition, to specify security-related attributes. Hence, for the purpose of automatically managing the Security SLA life cycle, we introduced a Security SLA model and a machine-readable format based on the WS-Agreement's XML schema and extended with all security-related information.

An abstract view of the SLA machine readable format is represented in the UML diagram in Figure 4: as shown, it is completely compliant with the discussed SLA model, which is integrated within the WSAG specification (the extensions to WSAG that we proposed to address security are highlighted in light grey). Note that WSAG include terms able to specify the business values associated to the SLA (`BusinessValueList`), like the penalties associated to SLA violations, in the following we do not describe them for simplicity's sake.

Hence, as devised by WSAG, a Security SLA is provided with basic information such as the agreement name and context data (including the agreement initiator and responder) and includes a `Terms` section (refer to WS-Agreement specification), further structured in `ServiceTerm` and `GuaranteeTerm`. Service terms provide information on the services to which the agreement is referred and to which guarantee terms can apply, while guarantee terms specify the service levels that the parties agree upon.
`Service terms` are further refined in `service description terms` and `service property terms`. Service description terms define the functionalities that will be delivered under the agreement, and are characterized by a term name, a service name, and a domain-specific description of the offered/required functionalities. In order to enrich the WSAG specification with security-related information, we proposed a security-based domain-specific service term description, made of the following three sections:

- `Resources Provider`: this section describes the available infrastructure resource providers (id, name, zone, and maximum number of allowed instances reservations, if applicable) and the available appliances (i.e., VMs) offered by each provider (type of appliance, HW/SW features and description);
- `Capabilities`: this section describes the security capabilities offered/required on top of the services covered by the agreement. As already mentioned, each capability is defined as a set of security controls belonging to a Security Control Framework, such as NIST's Control Framework or Cloud Security Alliance's Cloud Control Matrix;
- `Security Metrics`: this section includes the specification of the security metrics referenced in the `service properties` section and used to define Security Service Level Objectives (SLOs) in the `guarantee terms` section. A metric specification includes all information needed to identify it and to correctly process the SLOs in which it is involved, such as the metric name, its definition, its unit and scale of measurement, and the expression used to compute its value.

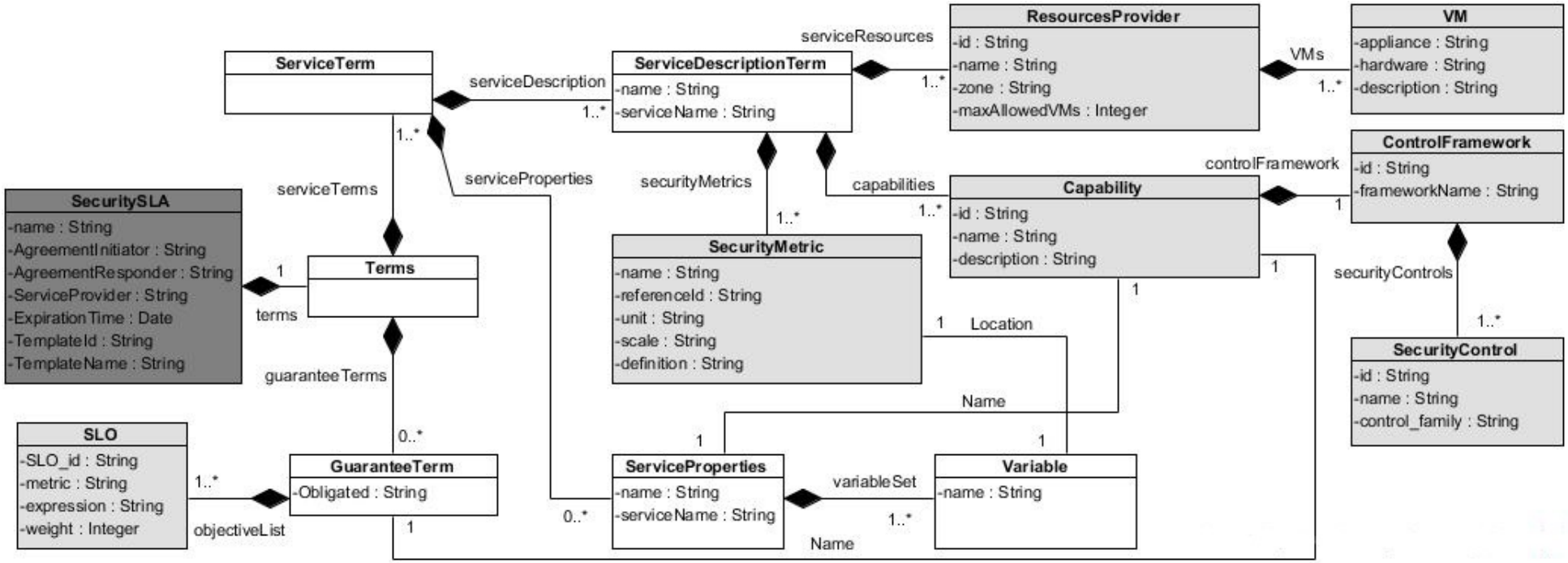

Figure 4. SLA machine-readable format model

`Service properties` are used to define measurable and exposed properties associated with a service. In our model, each service property is explicitly associated with a security capability (since it is used to check the enforcement of related security controls), and contains a set of `variables`, referring to security metrics above defined and representing the actual parameters adopted in SLO expressions.
Finally, `guarantee terms` include the conditions that must be verified to fulfil the agreement. We adopted the *CustomServiceLevel* item of the WSAG specification to define our custom Security SLOs, identified by an SLO id, a reference to the metric involved in the SLO, and the related expression, along with a weight assigned by the service customer and representing the related level of importance.

The SLA platform is described in D1.1.3. The XSD schema of the machine readable format is available online[2] and also reported in D1.3. In Appendix I, we provide an example of instantiation of such schema for a specific case study.

4.3 SPECS metrics catalogue

For the purposes of the SPECS negotiation the most important element of a security SLA is the SLO. According to the conceptual model presented in Section 3.1, a SLO is composed of one metrics (either quantitative or qualitative), where the SLO metrics are used to set the boundaries and margins of errors CSPs have to abide by (along with their limitations).

The following table shows the metrics used for the SPECS services. These metrics are measurable, monitorable and can be enforced. Each metric is associated to a security capability as described in D4.3.2. The list of metrics proposed results from the design and implementation of the security mechanisms available at state of art and it makes obsolete the list of metrics that was reported in D2.1.2.

The metrics are mapped to control categories both from the NIST [15] and Cloud Security Alliance's CCM [18].

Capabilities	**Description**	**Mapping to NIST[15]**	**Mapping to CCM [18]**	**Capabilities**
Level of Redundancy (LOR)	This metric sets the minimum number (with respect to EU's requirements and technological constraints) of web server engines, which are set-up and kept active throughout the service operation to increase the protection from attacks and vulnerabilities exploits. For example, *level_of_redundancy = 3* ensures that there are at least three web servers running.	CP-6, CP-7, CP-9, CP-10, SC-5, SC-22, SC-36, SA-2, SI-13	BCR-01	Web Resilience
Level of	This metric sets the number of	SC-23	BCR-01	Web

[2] Schema for the SPECS SLA specification: http://www.specs-project.eu/resources/repositories/

Diversity (LOD)	different web server types available on target VMs. For example, for *level of diversity = 2*, SPECS ensures that there are at least two different types of web servers deployed and available.			Resilience
TLS Cryptographic Strength (TCS)	This metric sets the cryptographic strength to be used by the TLS Terminator. TLS Terminator Configurator will choose the appropriate cryptographic ciphers that meet the negotiated level and configure TLS Terminator accordingly.	SC-13	EKM-01	TLS Security
Forward Secrecy (FS)	This metric ensures that the encrypted data sent through a session of the TLS secure channel cannot be decrypted even if the cryptographic data, used to generate the cryptographic credentials for that session, are compromised.	SC-12	EKM-03	TLS Security
HTTP Strict Transport Security (HSTS)	This metric is a feature of HTTP transport layer that declares the web content available only over a secure HTTP connection.	SC-43	IAM-02	TLS Security
HTTP to HTTPS Redirects (HHSR)	This metric is a feature of HTTP delivery service that forces clients to use only secure HTTP protocol.	SC-8	EKM-03	TLS Security
Secure Cookies Forced (SC)	This metric is a feature of HTTP protocol to force the clients to download session cookies, delivered by the HTTP services, only through a secured HTTP communication	SC-29	EKM-03	TLS Security
Certificate Pinning (CP)	This metric is a feature of HTTP protocol allowing the verification of the SSL certificates between the client and the HTTP service where the hash of the public certificate is pinned into the HTTP response.	SC-17	IAM-09	TLS Security
Scanning Frequency - Basic Scan	This metric sets the frequency of the basic software vulnerability scanning. For	CA-7, RA-5	TVM-02	Software Vulnerability Assessment

(BSF)	example, for *scanning_frequency = 24h*, SPECS ensures that software vulnerability scanning will be performed at least once every day.			
List Update Frequency (LUF)	This metric sets the frequency of updates of the list of disclosed vulnerabilities. For example, for list_update_frequency=12, SPECS ensures that the list of published vulnerabilities will be updated and presented at least once every 12 hours.	CA-7 (3), RA-5 (1)	TVM-02	Software Vulnerability Assessment
Write-Serializability (WS)	This metric ensures the EU that any WS violation to his stored data will be detected in a defined period of time (detection periods are less than *2*epoch*). In case of WS violations, the EU will be notified and the system will be restored to the state of the last finished *epoch*.	CP-2 (4), CP-2 (6), CP-6 (1), CP-9, CP-9 (6), CP-10, SI-7, SI-7 (1), SI-7 (2), SI-7 (5)	IVS-02, BCR-01, BCR-11	Database and backup as-a-Service
Read-Freshness (RF)	This metric ensures the EU that any RF violation to his stored data will be detected in a defined period of time (detection periods are less than *2*epoch*). In case of RF violations, the EU will be notified and the system will be restored to the state of the last finished *epoch*.	CP-2 (4), CP-2 (6), CP-6 (1), CP-9, CP-9 (6), CP-10, SI-7, SI-7 (1), SI-7 (2), SI-7 (5)	AIS-03, BCR-01, BCR-11	Database and backup as-a-Service
Client-side Encryption Certification (EC)	This metric ensures that the E2EE Client component available at the provided address is certified and thus grants the security of the encryption.	SC-12, SC-13	EKM-01, EKM-03	End-2-End Encryption
Scanning Frequency - Extended Scan (ESF)	This metric sets the frequency of an extended software vulnerability scan. For example, for scanning_frequency=48, SPECS ensures that software vulnerability scans will be performed at least once every two days. Scans are performed	CA-7, RA-5	TVM-02	Software Vulnerability Assessment

	with two scanners and both scanning reports are presented.			
Up Report Frequency (URF)	This metric sets the frequency of checks for updates and upgrades of vulnerable installed libraries. SPECS first updates vulnerability list, performs the vulnerability scan of the system, and then checks for available updates and upgrades of libraries on which vulnerabilities have been detected). For example, for up_report_frequency=24, SPECS ensures that checks for updates and upgrades are performed at least once every day.	CA-7, RA-5	TVM-02	Software Vulnerability Assessment
Penetration Testing Activated (PTA)	This metric activates the penetration testing activity. The metric can be chosen together with metrics related to vulnerability scans. If chosen, scanner with penetration testing functionality is deployed.	CA-8	TVM-02	Software Vulnerability Assessment

Table 6. Metrics implemented by SPECS services that are enforceable and monitorable

Table 7 displays the ngDC metrics developed in WP5 (to be reported in D5.3) as part of the storage automation software (ViPR[3]) that centralizes, automates and transforms storage into a simple extensible and open platform.

Metric Name	Description	Mapping to CCM	Mapping to NIST	Capability
RAID Level (s)	Select which RAID levels the volumes in the virtual pool will consist of.	BCR-01	SA-2, SC-6, CP-9, CP-10, SI-17	Availability
Multi-volume Consistency	Volumes can be assigned to consistency groups to ensure that snapshots of all volumes in the group are taken at the same point in time.	BCR-01	CP-1, CP-10, SI-17	Availability
High Availability (Type)	HA provides the foundation for a highly available environment.	BCR-01	SC-6, SI-17	Availability
Maximum Snapshots	Maximum number of local snapshots allowed for resources from this Virtual Pool.	BCR-09	SC-6, SI-17	Availability

[3] http://www.specs-project.eu/solutions-portofolio/vipr/

Max Native Continuous Copies	The maximum number of continuous copies for a virtual pool	BCR-09	SC-6, SI-17	Availability
HA Max Mirrors	Maximum number of data storage mirrors	BCR-09	SC-5, SC-6, SI-13, CP-6, CP-9	Availability
Provisioning Type	Storage type provisioning for the current virtual pool	IVS-04	SA-2, CM-2	Performance
Protocols	This depends on what is available to ViPR (e.g. could also support ScaleIO)	BCR-11	SA-2, CM-2	Performance
Drive Type	All current supported hardware type	IVS-09	SA-2, CM-2	Performance
System Type	Supported system type for the virtual pool	IVS-09	SA-2, CM-2	Performance
Min SAN Multi Path	The minimum number of paths that can be used between a host and a storage volume. If this many paths cannot be configured, Export requests will fail.	BCR-09	SC-6, SI-17	Performance
Max SAN Multi Path	The maximum number of paths to a given StorageArray from a host. Depending on paths_per_initiator, one or more ports may be assigned to an initiator if max_paths is sufficiently high for the number of initiators.	BCR-09	SC-6, SI-17	Performance
Data geolocation	In which data center the virtual storage and its copies are located	BCR-06	PE-17, PE-18, PE-20, SI-12	Security Storage
Anti-virus Policy	Anti-Virus scanning schedule interval in the virtual storage	TVM-02	CA-7, SC-28, SC-35	Security Storage
CloudProof Write-Serializability	This metric ensures the EU that any WS violation to his stored data will be detected and remediated in a defined period of time (detection and remediation periods are less than 2*epoch). In case of WS violations, the EU will be notified, and the system will be restored to the state of the last finished epoch.	IVS-02	CA-7, SC-28, IR-5, IR-8	Security Storage
CloudProof Read-Freshness	This metric ensures the EU that any RF violation to his stored data will be detected	AIS-03	CA-7, SC-28, IR-5, IR-8	Security Storage

	and remediated in a defined period of time (detection and remediation periods are less than 2*epoch). In case of FR violations, the EU will be notified, and the system will be restored to the state of the last finished epoch.			
CloudProof Client-side Encryption Certification	This metric ensures that the code available at an address is certified by a trusted entity.	EKM-01	SC-13, SC-17, SC-28,	Security Storage
Protection Mirror VPool	The virtual pool for protection mirrors	BCR-01	SC-6	Availability
HA VArray VPool	Indicates whether or not to use the HA side of the VPlex as the RecoverPoint protected site in an RP+VPLEX setup. In a MetroPoint context, if true, this field indicates that the HA VPlex site will be the active site.	BCR-01	SC-6	Availability
HA Protection Mirror VPool	The virtual pool for protection mirrors on the High Availability side	BCR-01	SC-6, SI-17	Availability
Fast Expansion	Indicates that virtual pool volumes should use concatenated meta volumes, not striped	BCR-07	SA-2, SC-6	Performance
Path per Initiator	The number of paths to be provisioned for each initiator that is used. In any event no more ports are used per host than max_paths. If there are excess initiators that cannot be paired with paths_per_initiator number of ports because max_paths is too low, the excess initiators are not provisioned.	BCR-09	SC-6, SI-17	Performance

Table 7. Metrics developed in WP5

5. Architecture overview

The high level overview of the Negotiation module is depicted in Figure 5. The architecture itself has not changed, what actually changed with respect to design in year 1 are the interfaces and interactions among the Negotiation components and the rest of the modules of the SPECS framework. Considering that the negotiation and renegotiation processes have been amended, the roles of components slightly changed.

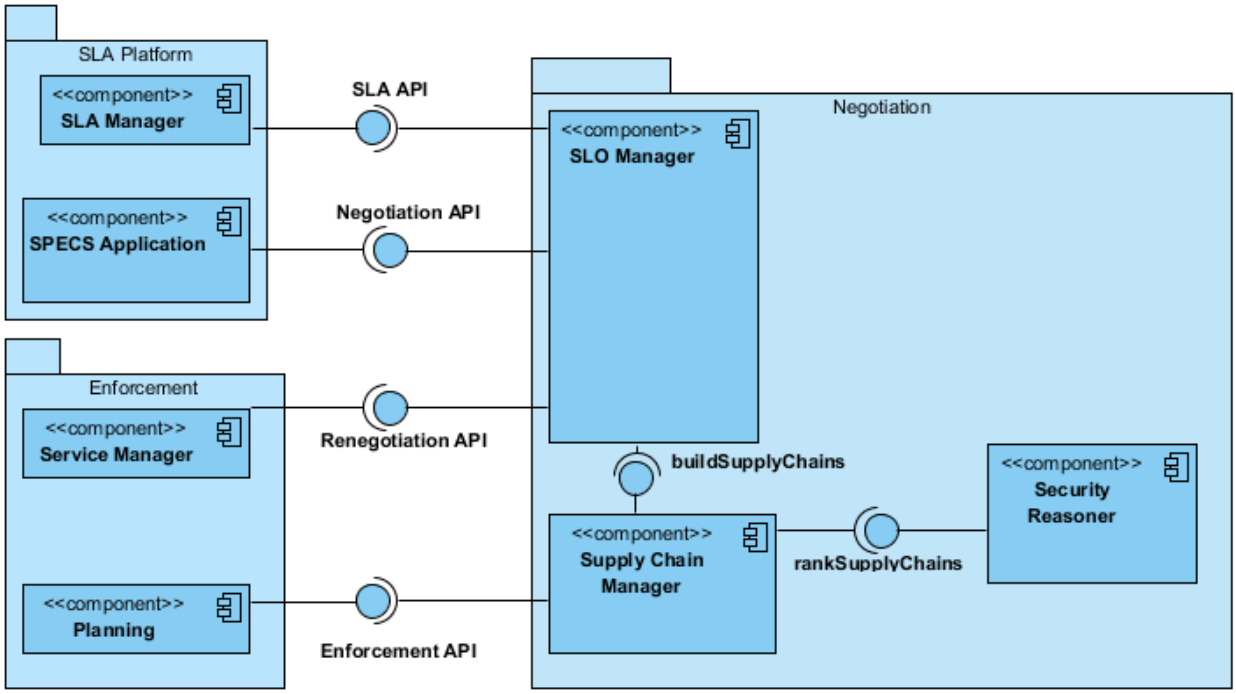

Figure 5. High level negotiation architecture

Three components comprise the final negotiation architecture:

- The SLO Manager is the component that offers the negotiation API to the SPECS Application. It orchestrates the entire negotiation and renegotiation processes. It manages the creation of SLA templates, it triggers generation of supply chains according to the End-user's security requirements, and invokes evaluation and ranking of the SLA offers that are built according to the supply chains.

- The Supply Chain Manager is the component in charge of building supply chains according to the set of security requirements chosen by the End-user. The creation of supply chains is supported by the Enforcement module (through the Planning); for further details see D4.3.2.

- The Security Reasoner component evaluates and ranks the SLA offers created during the negotiation process. The evaluation is done by using security assessment techniques that apply quantification algorithms to reason about the level of security provided by each of the SLA offer with respect to the end user requirements. Section 8 details the two techniques adopted in SPECS.

The (implementation) details of the building blocks, interfaces, and protocols are given as part of task T2. 3, and will be reported in deliverables D2.3.2 and D2.3.3 at M30.

6. SLA negotiation process

The evolution of the negotiation process during Y2 is a consequence of the implementation activities carried out in T2.3, some processes designed in Enforcement (i.e., supply chain creation and remediation, described in D4.3.2), and aspects of the SPECS Application (described in D5.1.3) approach.

The amended negotiation process has two main new features with respect to the one presented in the first year:

- End-users (EUs) can decide more aspects of the service, including the type of service, the CSP to be used for each service, the security capabilities to add to the service and the controls and metrics details for each of the selected capabilities. This approach allows to enhance the usability of the solution as seen from the EUs perspective. Separating the definition of user´s preferences into services, capabilities, controls and metrics is a flexible way to define different approaches for each feature to be specified by EUs. The D5.1.3 details how this has been implemented.
- The role of the CSP during the negotiation process has been included by adding a certification of the valid offers performed by CSPs. Like in real life, the negotiation approach proposes a bilateral agreement between the CSP and the EU, where both parties sign the contract. The signature of the CSP certifies that (1) the CSP can provide all the features (from a security perspective and also from a functional point of view) that an SLA specifies, (2) the CSP guarantees to provide the terms included in the agreement. From the EU's perspective the signature is used to certify the contractual relationship between the CSP and the EU (payment conditions, actions in case of unfulfillment, etc.).

Figure 6 represents a simplified flow diagram of the negotiation process.

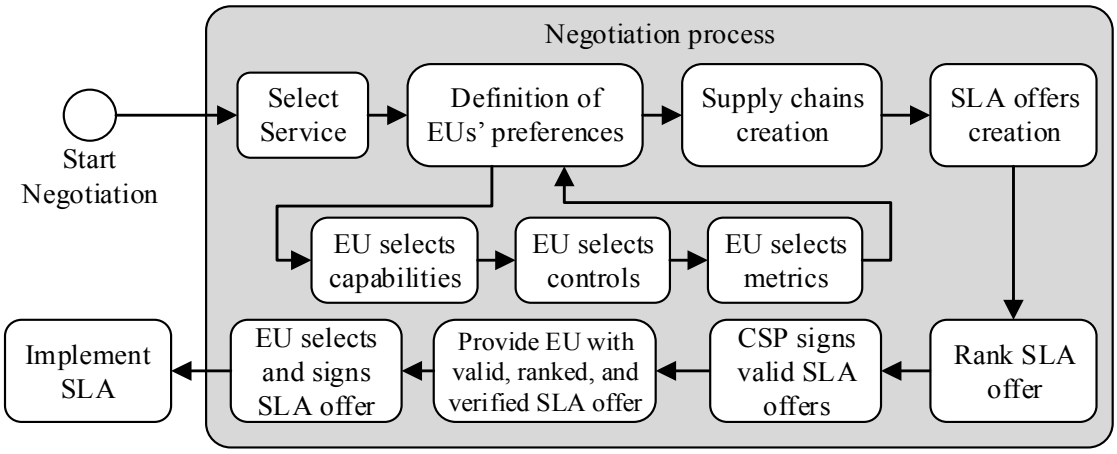

Figure 6. Simplified negotiation process

The End-user starts selecting the service to use (for example, a Secure Web Container service). Once the service is selected, the EU customizes the security aspects offered by the chosen service. The steps to define the security features of the service have been refined in Y2. The term capability is now introduced which represents a set of security controls that can be implemented with one or more security mechanisms on top of a security service. For example, in case the EU wants periodic vulnerability scans on the requested web servers under the umbrella of the Secure Web Container service, the capability to add will be the Software Vulnerability Assessment capability (which can implement a security control related to penetration tests). On top of the chosen capabilities, the EU can fine tune his preferences by specifying concrete controls and metric values. Of course, this depends on the level of expertise

of the EU. Expert EUs are able to specify specific values (being able to specify, for example, the frequency of the vulnerability scans). Non-expert EUs are able to specify qualitative requirements (in the form of not important, very important). This uncertainty of non-expert EUs is taken into account when evaluating the security level of the service offers chosen by the EU. The security reasoner described in Section 8.2 uses fuzzy-based algorithms to add the uncertainty to the analysis.

Once SPECS collects all the information required by the EU, the creation of the SLA offer starts. Each created SLA offer will correspond to a supply chain and each supply chain is composed of one CSP and a set of resources enriched with security mechanisms enforcing and monitoring EU's chosen security features. The supply chains are created by SPECS according to the available security mechanisms (either offered by SPECS or provided by external CSPs) and security requirements provided by the EU. The combination of these elements will provide a list of supply chains that will be transformed into a set of potential SLAs (i.e., SLA offers). Each SLA offer will represent one supply chain. Before each SLA offer is proposed to the EU, it has to be validated by the CSP. This is necessary to guarantee that the supply chains created are actually feasible (for example, to check that the CSP can actually provide the service with the controls specified by the EU). The EU then receives the list of valid SLAs. The list of SLAs is ranked according to the EU's requirements by applying the reasoning algorithms that perform comparisons and evaluations to determine what are the SLAs that better match EU requirements.

A more detailed negotiation process is depicted in the sequence diagram of Figure 7. The interactions with the SLA Platform and with the Enforcement module are clearly represented in the diagram.

To understand the detailed process, we will only outline the most relevant steps. For more implementation related details of this process we forward the reader to deliverables D2.3.2, D2.3.3 and D4.3.3 delivered at M30.

The negotiation process is triggered directly by the EU through the SPECS Application. The request is forwarded to the SLO Manager (step 2-4) that returns to the EU the list of security services offered by SPECS, for example, a secure web container service or a secure storage service. Note that all communication between the EU and SPECS goes through the SPECS Application.

The selection of a service triggers definition, by an EU, of the specific requirements for the selected service. To do so, the SLO Manager (after receiving the service chosen by the EU in steps 5-6) retrieves from the SLA Platform the set of security features available for the selected service (steps 7-9). This is done by the usage of SLA templates that are modified according to the chosen service and the possible combinations of security features (capabilities, controls, and metrics). Note that only one service can be enforced with one SLA and that for each service one specific SLA template is available.

In order to build this set of security features for the selected service, the SPECS Application interacts with the EU offering first security capabilities, then security controls applicable to the chosen set of security capabilities, and at the end security metrics applicable to the set of chosen security controls (steps 11-15). As mentioned before, the way these security preferences are set depends on the type of EU (expert or non-expert).

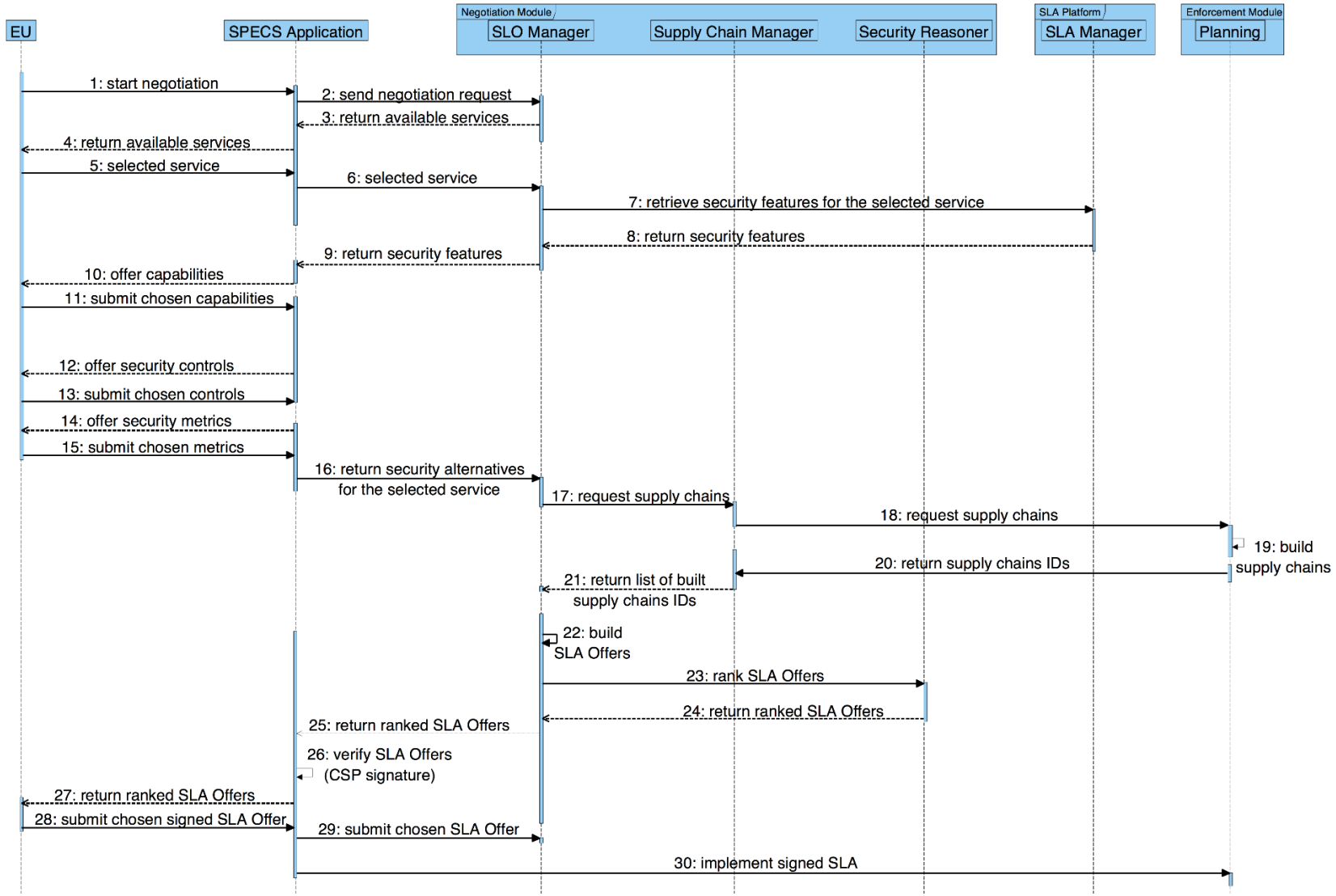

Figure 7. High level sequence diagram for the negotiation process

Steps 16-21 comprise the creation of the supply chains. Once the service and all the associated security features are set, the SLO Manager invokes the generation of the possible supply chains. The Supply Chain Manager invokes the Planning component which has all the information (about available services and resources and their implementation and configuration details) required to build all possible supply chains that fulfil EU's security requirements. The Planning replies back to the Supply Chain Manager with a list of IDs, each representing one supply chain that has been created. A detailed description of the process is available in D4.3.2.

The SLO Manager then retrieves all generated supply chains and creates an SLA offer for each supply chain (step 22). The complete set of SLA offers is given to the Security Reasoner that performs evaluation and provides the ranking of SLAs in terms of security levels they guarantee (steps 22-25). Note that each supply chain is tied to a different CSP, thus each supply chain and each associated SLA offer assures a different level of security. The techniques used to carry out this security assessment are described in detail in Section 8.

Before providing the EU with the complete ranked list of SLA offers, all CSPs are asked to check their validity (step 26). To avoid offering to the EU unfeasible compositions of services, the CSPs will check all composed provisions. Only valid SLA offers (i.e., valid supply chains) will be signed and offered to the EU (step 27). The EU selects his preferred SLA and signs it (step 28). The chosen SLA offer is then stored as a signed SLA (the process is handled by SLO Manager after step 29), and then it can be enforced (step 30).

7. SLA renegotiation processes

During the second year of the project, the process of renegotiation has been carefully studied. Strong synchronization activities have been conducted between the Enforcement and Negotiation modules. Renegotiation occurs when an enforced SLA needs to be changed for some reason. Two cases represent the situations where a signed SLA has to be renegotiated:

- CSP triggered renegotiation: in this case, a violation invalidates the current enforced SLA. This happens when a violation occurs and there is either no remediation action available or the remediation process requires a change in some SLO. As a result, the SLA is not valid anymore and a new agreement has to be negotiated.
- EU triggered renegotiation: in this case, the EU wants to change some of the conditions of the SLA (to add or remove capabilities, controls, metrics, or to simply change the conditions of one or more SLOs).

In both cases, the initially enforced SLA is not valid and a new SLA has to be signed. This is a mandatory requirement, since any change in an SLA, no matter how small it is, invalidates the signature and the contract.

To simplify the processes and optimize the need for implementation efforts, we tried to reuse the current negotiation process as much as possible. The following subsections detail the two types of renegotiation.

7.1 CSP triggered renegotiation

In a CSP triggered renegotiation, a notification from the RDS component of the Enforcement module starts the process. This notification may be the result of an SLO violation that entailed the invalidation of a signed SLA. The simplified process is depicted in Figure 8.

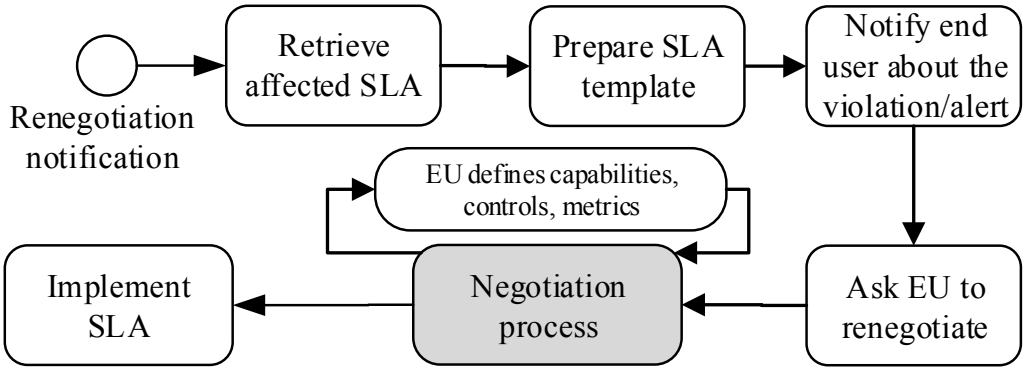

Figure 8 Simplified renegotiation process (CSP triggered)

Note that the Enforcement module only notifies the Negotiation that some part of the signed SLA is no longer valid. It is up to the End-user to either terminate the SLA, accept the risks associated to the violation, or renegotiate the SLA.

After receiving the notification from the Enforcement module, the process begins by retrieving the affected SLA. According to the violated SLA an SLA template for the initially enforced security service is retrieved and filled with the initial set of security features extracted by the violated SLA. This allows the EU (in case she/he decides to renegotiate the SLA) to use the initially chosen security settings, to check the affected SLOs, and to provide a new set of security requirements related to those affected SLOs. Of course, the EU is allowed to remove old and/or

add new security capabilities, controls, and metrics.

Once the EU has accepted to renegotiate the violated SLA, the negotiation process is carried out (following the process descried in Section 6). If the renegotiation ends successfully, the newly signed SLA is ready to be implemented.

This approach has allowed us to completely reuse the negotiation process, thus making the integration of both processes in the implementation stage much easier.

The CSP triggered renegotiation process is detailed in the sequence of Figure 9. The diagram highlights the negotiation process as reused from the one described in Section 6.

Steps 1-9 illustrate the steps that are exclusive to the CSP triggered renegotiation. Once the notification has been received from the RDS to trigger the renegotiation process (step 1), the SPECS Application retrieves the affected SLA from the SLA Manager (steps 2-3).

Steps 4-7 comprise the customization of the SLA template associated to the initially enforced security service. This customization (filling the template with EU's initially chosen security features) permits to show to the EU the initial service settings and makes the selection of new features easier. This can also be used to show to the EU the affected SLOs. The EU has the possibility to accept a renegotiation or terminate the SLA (step 8). The process of terminating an SLA is described in D4.3.2 as part of the Enforcement activities.

In case the EU accepts the renegotiation, the process of negotiating the new SLA starts (steps 10-30) with the same steps already described for the negotiation process (see Section 6). The difference is hidden in the way the EU is choosing the security features for the service. While in the negotiation process the preferences are chosen from scratch, in the renegotiation process the preferences are set to the values of the initial SLA, so that the EU can simply modify parameters that she/he prefers. During the renegotiation process the possible supply chains are built again and offered to the EU in the form of ranked SLA offers like in the negotiation. This is done to cover the possibility of changes in the number of required resources or in a combination of security mechanisms enforcing and monitoring the selected security features, or even due to the change of a CSP. For more details on the implementation details of this process we forward the reader to the deliverables produced by tasks T2.3 and T4.3.

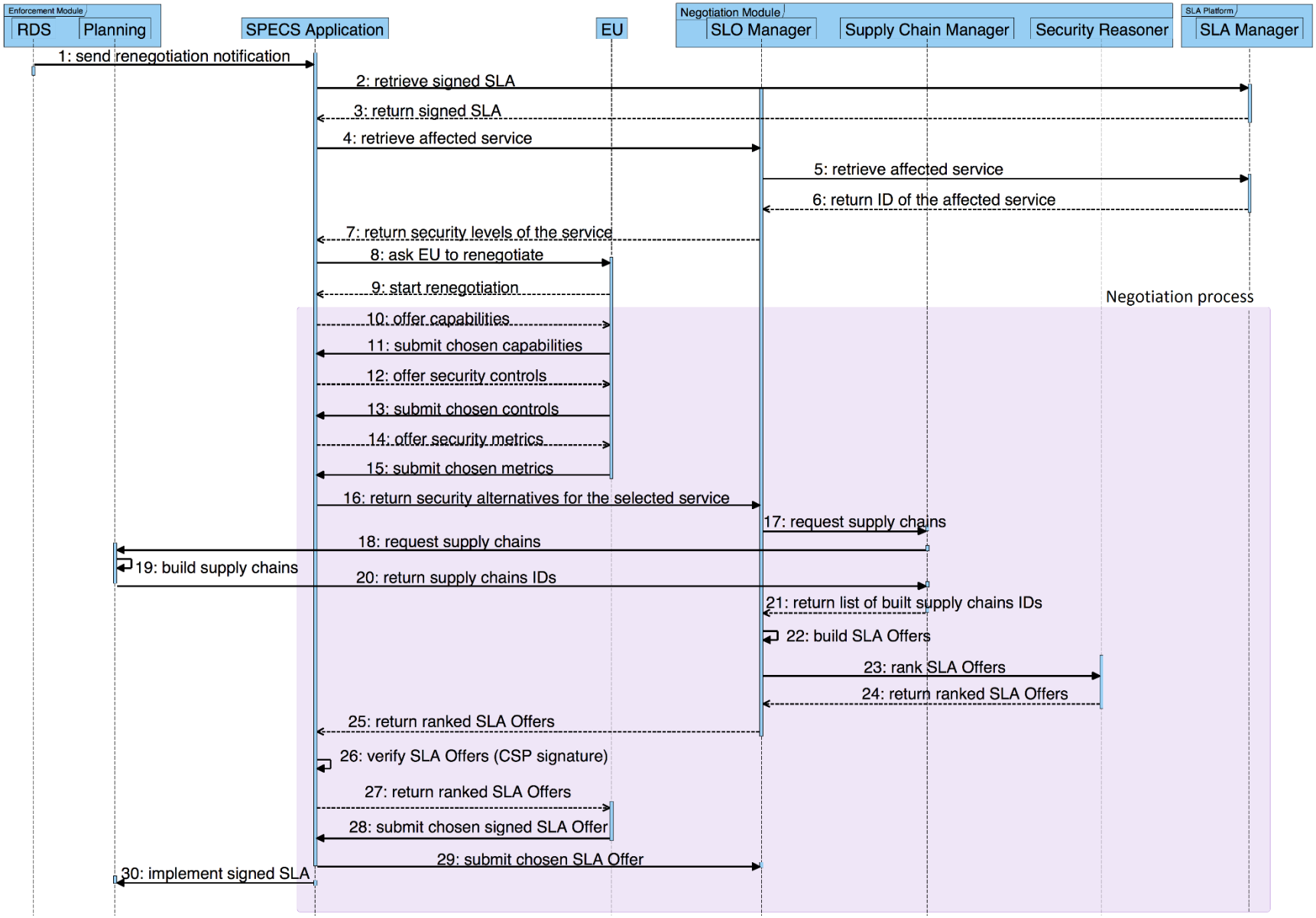

Figure 9. Renegotiation process triggered by CSPs

7.2 EU triggered renegotiation

An EU triggered renegotiation occurs when it is the EU who freely decides to change some aspect of his signed SLA (for example, to remove or add a new capability, control, metric, or modify conditions of some SLO).

The process of the EU triggered renegotiation (shown in Figure 10) is even simpler than the CSP triggered renegotiation and also reuses the negotiation process presented in Section 6.

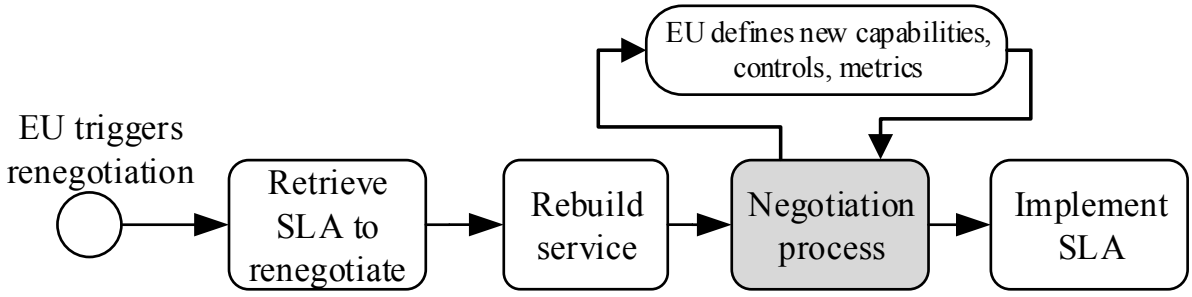

Figure 10. Simplified renegotiation process (EU triggered)

The EU sends a renegotiation request though the SPECS Application. The SLA is retrieved from the SLA Manager by using the ID of the service. An SLA template is then customized with the contents of the signed and monitored SLA. Similarly to the CSP triggered renegotiation, the customized SLA template contains the EU's initial security preferences and is used to prompt the EU to modify the existing ones or remove and/or add new features. The EU redefines capabilities, controls, and metrics, and the negotiation process continues up to the implementation of the signed SLA.

Figure 11 shows the detailed EU renegotiation process. Steps 1-9 represent the interactions that are exclusive to the EU triggered renegotiation while steps 10 to 30 correspond to the negotiation of the new SLA (to the process described in Section 6).

Renegotiation process starts with the EU's invocation (step 1). In steps 2-3 the previously enforced SLA that the EU wants to modify is retrieved from the SLA Platform. Same as in the CSP triggered negotiation, the content of this SLA is used to build a new SLA template (steps 4-7) that contains the initially negotiated SLOs. The customized SLA template is sent to SPECS Application that is used to give the EU the possibility to change the conditions of the initially enforced security service (step 8-15).

In this case, the process of creating new supply chains is done again since it is possible that with the new security preferences selected by the EU, new service offers can be provided (different CSPs, different settings for the capabilities, etc.).

For more implementation related to the details of this process we forward the reader to the deliverables produced by tasks T2.3 and T4.3.

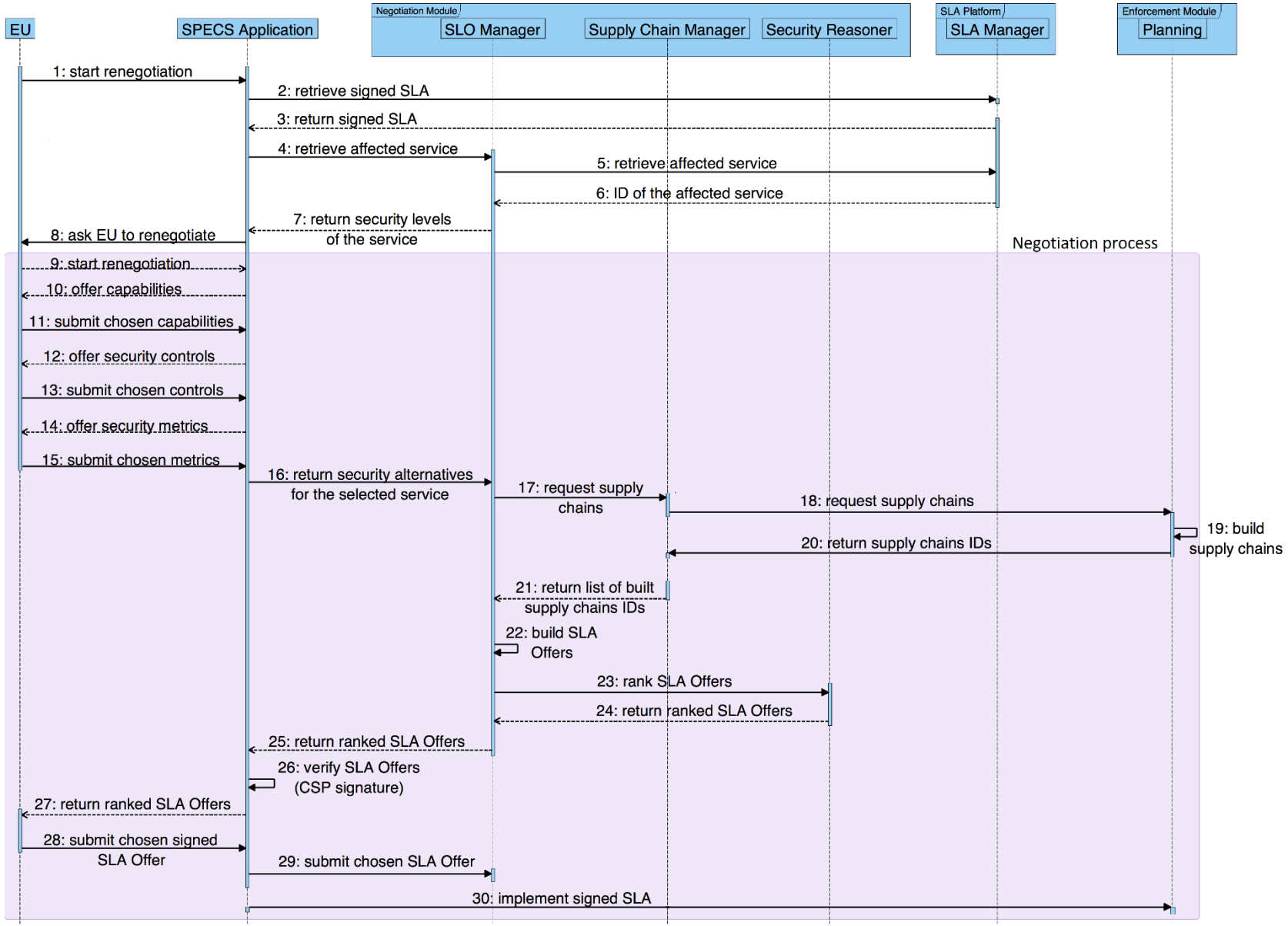

Figure 11. Renegotiation process triggered by an EU

8. Security Reasoners

According to the negotiation process a set of supply chains are created depending on the EU's requirements. These supply chains, as described in D4.3.2, are composed of one CSP and capabilities offered by SPECS which enhance some specific security features according to the EU's preferences. As described in Section 6, a different SLA offer is built for each supply chain created during the negotiation process; as a result each SLA offer is also linked to the security controls of a CSP and to the SLOs of the capabilities offered by SPECS. The proposed set of SLA offers are sent to the EU so that she/he can choose which one to sign.

While SPECS provides with mechanisms to enforce SLOs that are part of the SPECS services, there exists a certain degree of uncertainty with regards to the security controls provided by the CSPs that are part of an SLA offer but are out of the control of SPECS (because they are not enforceable or monitorable). To deal with this issue, the security reasoner provides with the necessary information that can help EU's to decide which CSP better matches his requirements. By comparing, considering controls implemented by the CSPs we are able to build a ranking of SLA offers. The score of each SLA offer will depend on the fulfilment of the security controls that are not enforceable by SPECS but are provided by the CSP that is part of the supply chain. Figure 12 summarizes the evaluation workflow, as requested in the SPECS behaviour. The reasoner has to extract the information from the SLA offers, as reported in the SLA machine readable format described in Section 4 and then evaluate them according to the reasoned methodology.

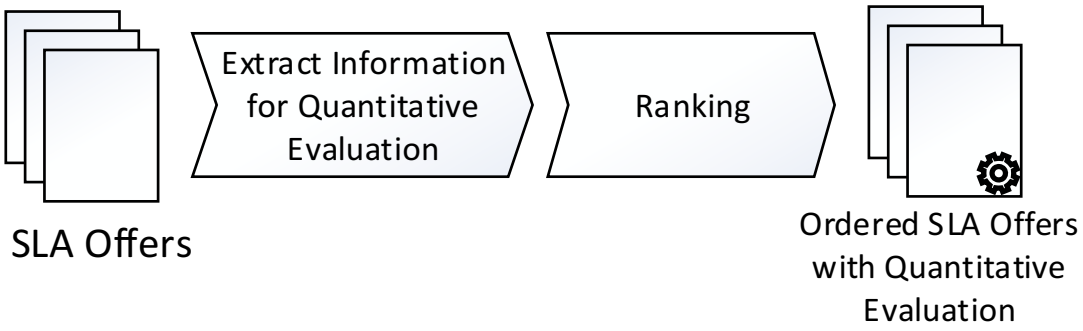

Figure 12. Simplified Evaluation Workflow

EUs' security requirements are used in a different way for reasoning depending on the type of control that is being considered:

- Requirements for SLOs of SPECS services. They can be enforceable and monitorable by SPECS and therefore are part of a signed SLA. SPECS can adapt the metrics of the security controls chosen by EUs and provide them with a compatible SLA offer.
- Requirements for security controls of CSP. These controls cannot be enforceable and monitorable, since they are under the CSP domain. As a result they are not part of the signed SLA. However they play an important role in the decision support mechanism that is provided to EUs by SPECS. Security reasoners are able to compare CSPs according to the level of fulfilment of these controls with respect to EU's requirements, thus providing them with a ranking that sorts SLA offers according these requirements.

The evaluation performed by the security reasoner is based on two assessment algorithms that

are able to compare EU security requirements with respect to the security controls provided by CSPs. SPECS has designed two algorithms:

- REM [18] (cf., section 8.1), that uses aggregation techniques to perform an evaluation of the security level provided by a provider.
- A fuzzy logic based security assessment methodology based on fuzzy-AHP [9][10] (cf., section 8.2) that is able to manage uncertainty of EU's requirements to provide a multi-layered comparison of the security provided by providers and requirements demanded by EUs.

The information used as input for both REM and fuzzy-QHP is based on the conceptual model defined in section 4 to represent SLAs. Both techniques will produce similar hierarchical structures to process the information, as it will be described for each technique in the following sections.

8.1 Use of REM for the evaluation of the SPECS SLA Model

This section describes how the REM technique has been adapted to be used in the SPECS context. The description of the REM methodology was reported in D2.2.1.

8.1.1 REM Evaluation of CAIQs Evaluation

In order to evaluate different providers that can be adopted with SPECS to host a target service, we used the information structure based on the SPECS conceptual model to represent SLAs. For the specific usage by REM the available security controls provided in the *Cloud Controls Matrix (CCM),* is used. Furthermore, we build a hierarchical structure of security controls by referring to the *Consensus Assessments Initiative Questionnaire (CAIQ)* [11] that provides a series of "yes or no" control assertion questions to assess Cloud Service Providers security.

The CAIQ can be considered a very simple form of Security Service Level Agreement representation: it declares all the security controls that a CSP is able to provide, even if it does not offer any concrete guarantee about their real enforcement (it is not a contract among customer and provider, it is just a public declaration). Moreover it cannot be monitored from a customer, not offering any concrete security metric. At most it is possible to perform an audit process which verifies the correctness of the CSP declarations. So, a security SLA contains all the information a CAIQ includes, but the contrary is not true. Furthermore a repository of questionnaire compiled by more than 100 CSPs is already available for comparison (c.f., STAR repository [17]). The positive aspect of the CAIQ and the STAR repository is that they represent a public repository of declarations that enables an EU to perform a comparison among the security offered by each CSP; nevertheless, the CAIQ contains about 300 questions (categorized in controls and control domains) making it very difficult to analyse them for CSP comparison from the EU's perspectives. The REM easily supports such a process offering a quantitative representation that takes into account the EU's relative needs.

Figure 13 illustrates the REM methodology steps applied on the CAIQ: Structuring, Formalization, Weighting and Evaluation.

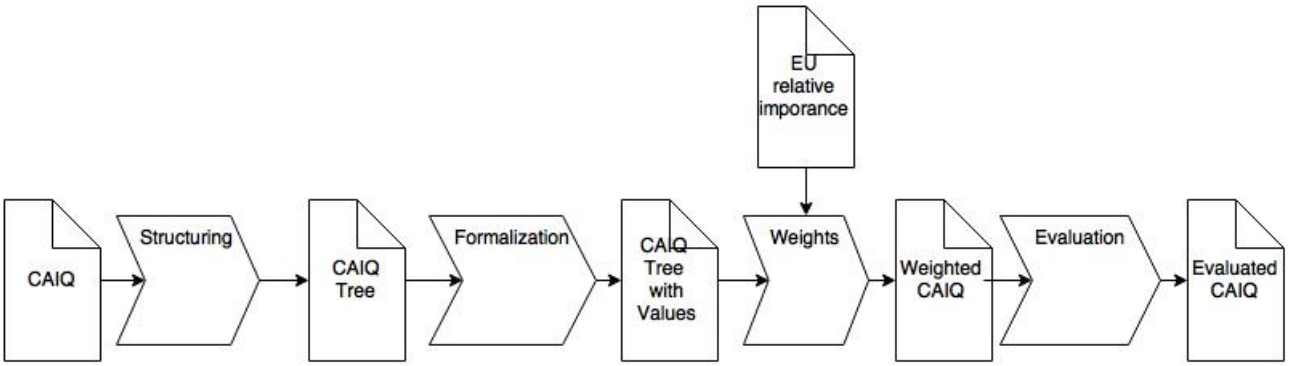

Figure 13. REM Evaluation steps applied on the CAIQ

The goal of the Structuring phase is to create a tree data structure starting from the CAIQ and assign an enumerative data type to each node of the tree. In the case of the CAIQ, this process is very simple; as illustrated in Figure 14, the CAIQ already has a tree structure, the root node is associated to the full questionnaire, and second level of the tree includes the control categories, the following one to the Control groups and the latest one to the specific security controls.

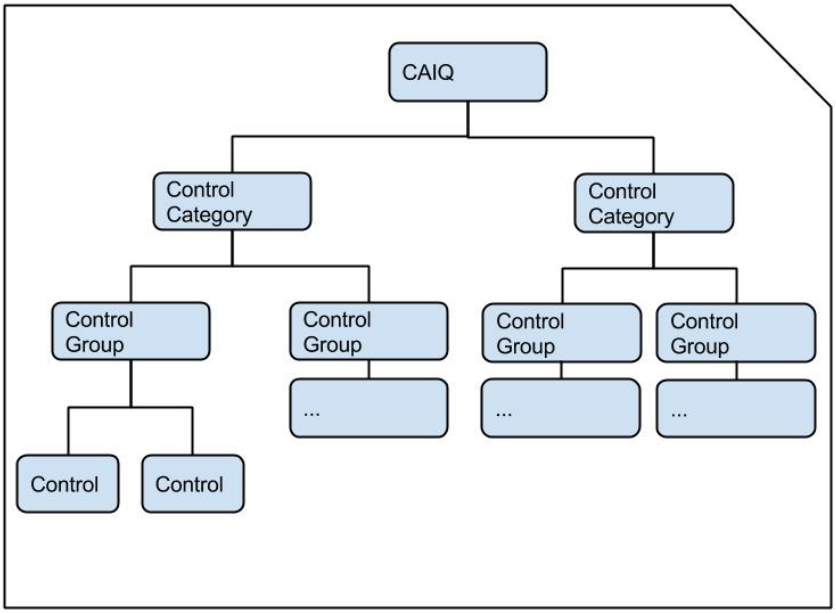

Figure 14. The CAIQ tree

In the Formalization phase, the CAIQ tree is turned into a tree enriched with homogeneous values. All leaf nodes in the CAIQ tree have the same data type (*Yes*/*No*) but in some cases they are not specified (*N*/*A*); we can represent these with an ordered enumerative values (*N*/*A*, *No*, *Yes*) and assign a numerical value for comparison. The set *{N*/*A*, *No*, *Yes}* can be ordered in this phase, according to different evaluation criteria. In SPECS, we proposed a default ascending order: *N*/*A* means that CSP is not able to reply and we consider this worse than the explicit choice of NOT adopting the control.

These values are then mapped on a scale of four security levels (c.f., Local Security Levels)[4]; a possible mapping is: *Yes= 3, No=1 and N/A=0.*

[4] With the REM, the assignment of values is configurable. We suggest four local security levels and this mapping, since this choice better highlight the difference between very similar SLAs.

An End-user can give a different importance to each security control and control group in the tree. In the Weighting phase the End-User can provide its own weight on both single controls or on the larger category and express, in this way, his/her priorities and desiderata.

In the last step, it is possible to evaluate the Global Security Level provided by the CSP.

The Global Security Level has been defined on the basis of a Euclidean distance among matrices and some reference levels. This function gives a numerical result to the security but can be easily applied to different sub-trees of the CAIQ in order to help the End-user to visualize the weaknesses and strengths of different providers.

8.1.2 Evaluating SLA offers with the REM

In order to completely use the REM to evaluate SLA offers represented according to the proposed conceptual model, we need to pre-elaborate the SLA offers in order to extract information for the evaluation.

According to the proposed SLA model, an SLA offer contains the following information:

- *Target Service* and *Resource* providers, i.e., the service offered to EUs and the resources/services requested to External CSPs.
- *Security Capabilities*, i.e., the set of security controls associated to the service which can be granted to the EU.
- *Security Metrics*, i.e., the quantitative values used to monitor the enforcement of the security controls.
- *SLOs*, i.e., the objectives, expressed with respect to security metric values that must be respected to grant the SLA.

Furthermore, in the proposed model these fields are enriched with an *importance* attribute to specify weights, i.e. which controls, metrics, and SLOs are considered more relevant from the EU's perspective. Such attributes are extremely useful in the negotiation phase, because they help the EU in the selection of an SLA Offer. Indeed, an SLO *must* be respected independently from its importance value.

In order to apply the REM methodology, we need a representation of the SLA as a tree. In D2.2.1 we introduced the SLA Hierarchy to transform the SLAs in trees that the proposed methodology is able to evaluate. The Global Security Level associated to the root node of the tree is the quantitative evaluation associated to the SLA offer.

In the case of the REM methodology, the comparison among different offers is meaningful only if the tree structure is the same (i.e. we can compare two SLA offers only if they contain exactly the same number of nodes and only the value of the leaves are different).

Figure 15 graphically illustrates the process to transform a SLA Offer, represented according to the proposed SLA model, into a SLA (tree) hierarchy, as introduced in D2.1.2, to enable a clear evaluation of the SLA.

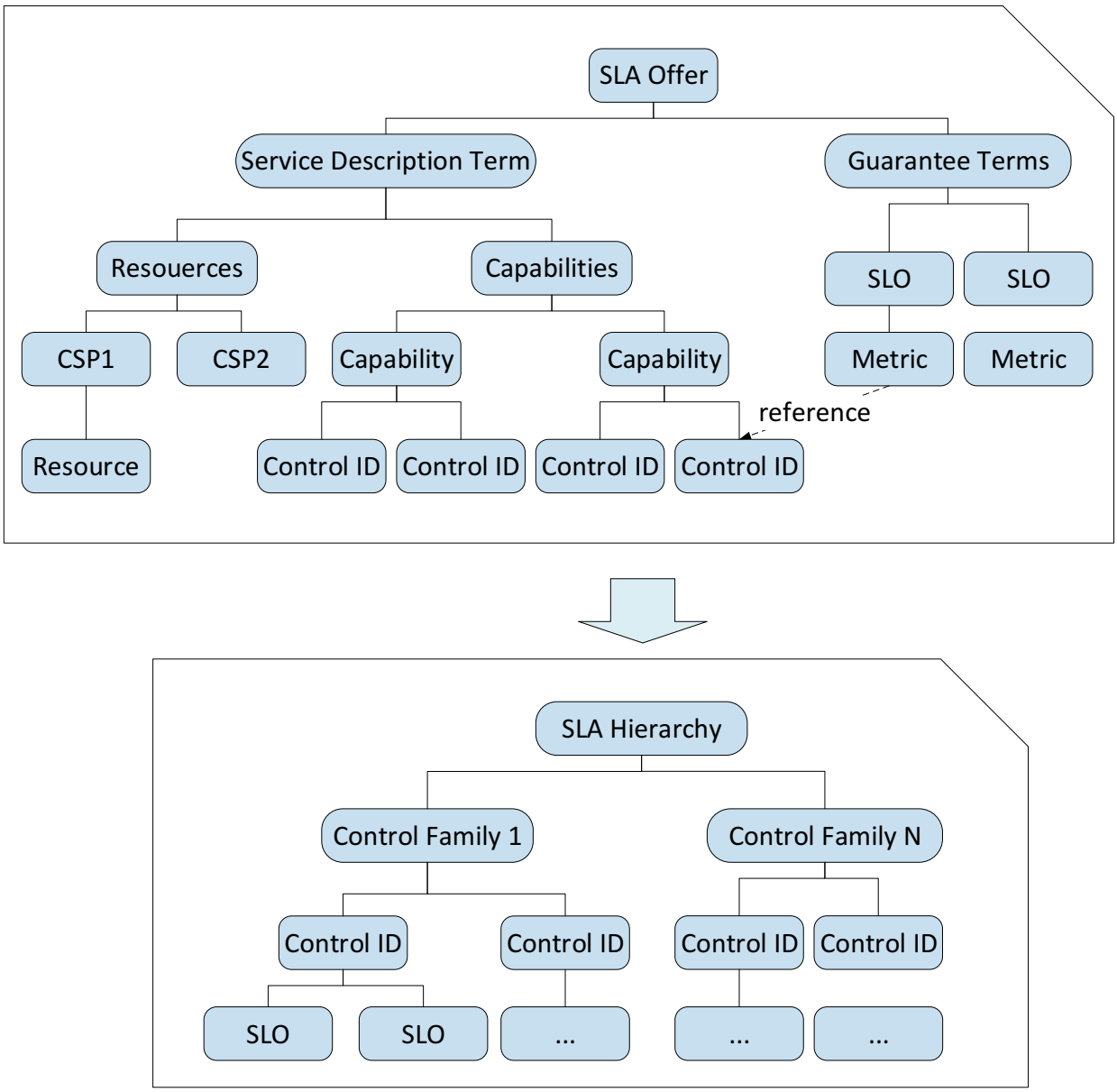

Figure 15. From SLA Offers to SLA Hierarchy

The difference between the two formats is given by the different usage context: one aims at automating the process of SLA management (SLA offers), the other focuses only on the evaluation.

It is important to outline that each SLA offer has only one External CSP that hosts target services and resources (we always assume a single cloud provider, multi cloud application are out of our scope). The CSP offering resources may have its own security features that should be taken into account and the STAR repository contains a very wide set of such declarations from many European cloud providers.

Finally, the approach we adopted for SLA offers evaluation is very simple and takes into account both the selected CSP hosting the target service, through its CAIQ available in the STAR repository, and the specific Service Level Objectives.

As illustrated in Figure 16, we locate and retrieve from each SLA offer the CAIQ associated to the CSP (we have in this way a shared tree that outlines which are the controls that are granted

by the external CSP). Note that they are declared by a en external CSP and the SLA offer does not offer any concrete grant on top of them so, for this, in the negotiation we just give support to choose the CSP.

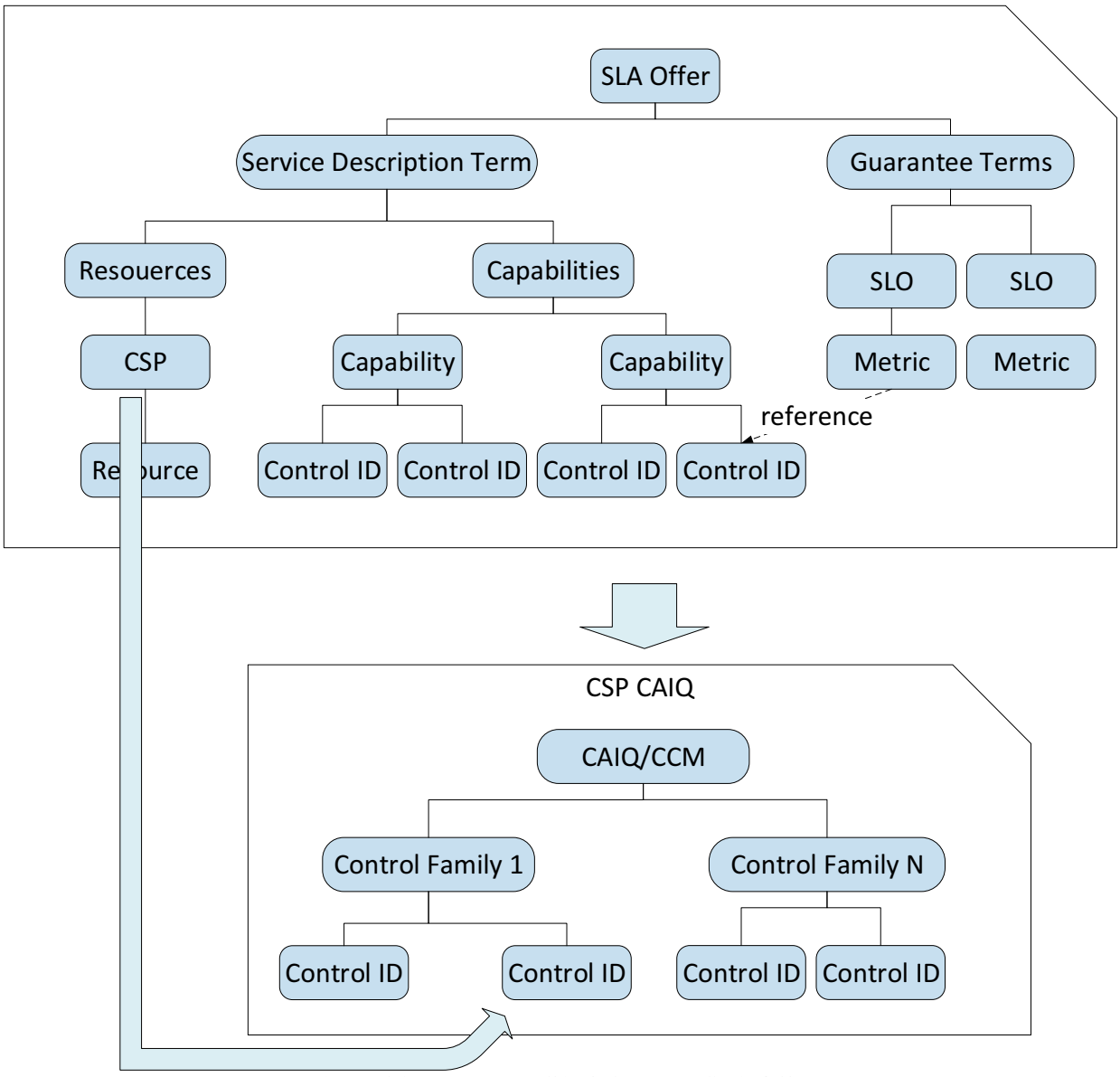

Figure 16. Extract the CAIQ from the SLA Offer

As a second step we extract each capability from the SLA Offer, in order to know which are the additional controls that we are able to enforce through the SPECS security mechanisms.

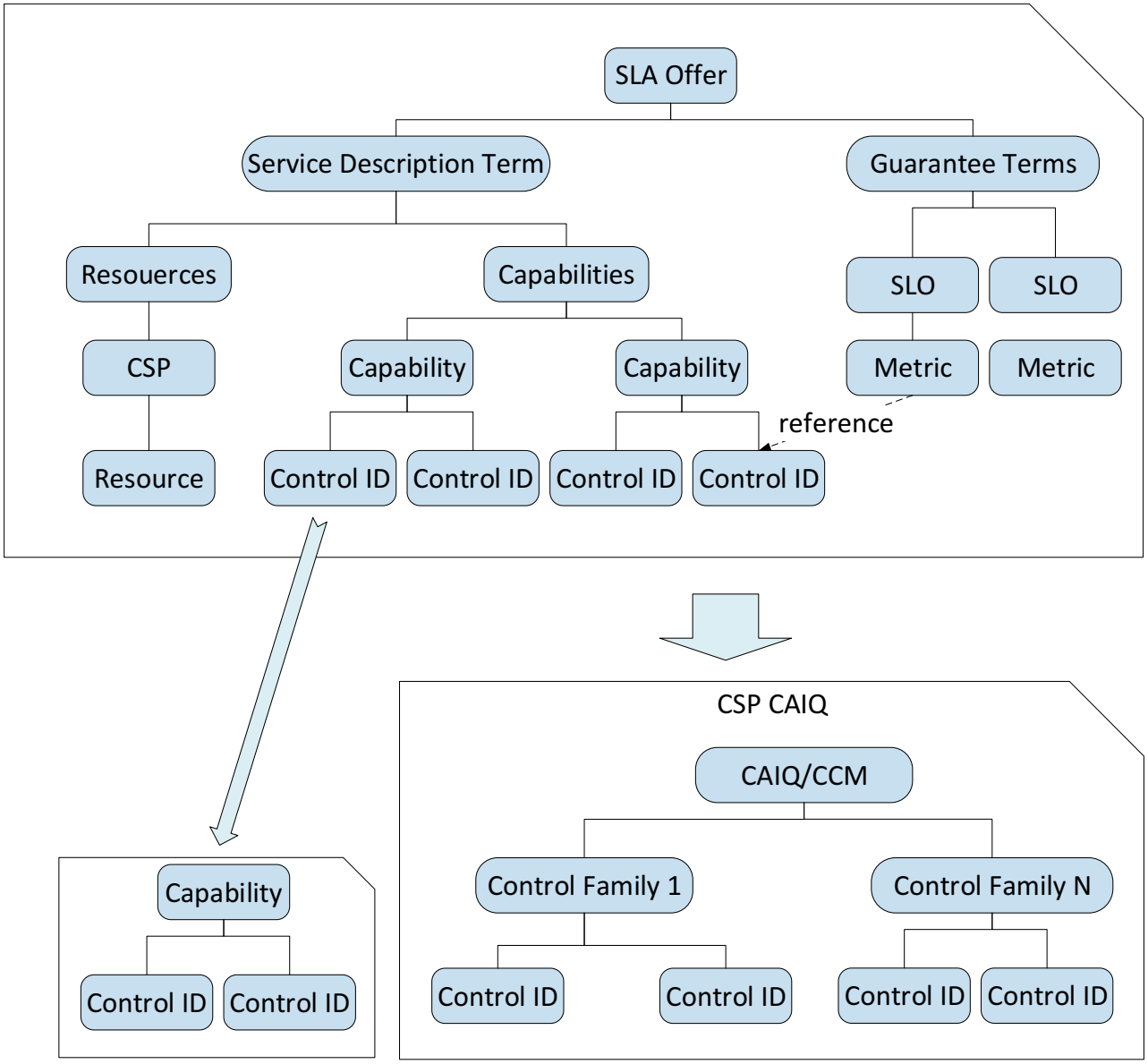

Figure 17. Capability extraction from an SLA offer

The final result is that we are able to generate a new CAIQ, which refers, this time, not to the external CSP, but to SPECS as provider of the specific service.

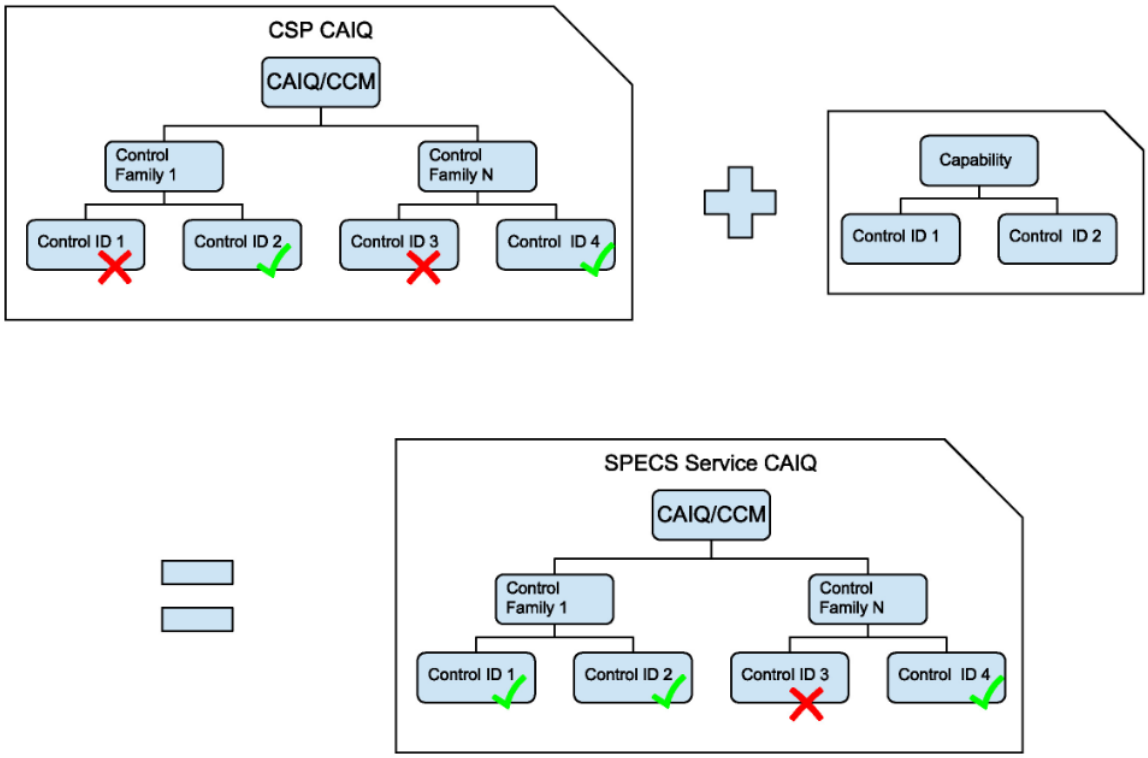

Figure 18. Generation of the SPECS CAIQ

We can finally use the REM evaluation technique to evaluate and compare the SPECS CAIQs associated to the different SLA offers.

8.2 Fuzzy logic based security assessment of SLAs

The conceptual model defined to represent SLAs (as described in section 4), considers the specification of EUs' security requirements by defining levels of importance in the form of weights. To encourage the smooth adoption of SPECS services by EUs, we took into consideration the EUs' desire to express their general requirements or imprecise preferences in natural language phrases (for example, to only assign a certain level of importance to a set of offered features). Most of the used security assessment techniques require EUs to provide detailed description of their requirements and submit static weights to model their priorities [1][2][3][5][6], which require expert knowledge and are time consuming. In addition, it is also difficult for some EUs to determine security requirements in accurate values. In light of the above, it is becoming an important issue for both CSPs and EUs to be able to make decisions regarding how to assess and rank SLAs with respect to EUs' uncertain requirements.

In this section we specify a quantitative reasoning approach to cloud SLAs that facilitates the following:

1. Assessment, comparison, and ranking of various SLAs by using an assessment technique to identify the one that better match the EU's security requirements. This assessment technique is based on the fuzzy analytic hierarchy process (fuzzy AHP).
2. Submission and specification of EU's requirements and preferences using natural language phrases and linguistic descriptors at various levels of security services, -thus allowing both novice and expert EU's to provide their security requirements according to their expertise and specific or uncertain needs.

3. Capture EU's subjective requirements through employing membership functions that use a fuzzy inference system to derive the EUs' required security levels.

The SLA offers are constructed as a SLA hierarchy as described in Section 4. This hierarchical structure allows EUs to have the ability to specify their security requirements (according to their expertise) at different levels of the SLA hierarchy. The results can be used to provide with a graphical interface that EUs can use to analyse the results or even to obtain the required SLAs. In SPECS, the comparison is made to provide a ranking that sorts SLA offers according to EU's requirements.

Figure 19 illustrates the general overview of the methodology. There are two major steps. The first step captures EUs' descriptive requirements. The second step computes quantitative values for SLA offers based on their security levels measured according to the EU security requirements.

The main steps are performed in progressive stages, as shown in Figure 19. In Stage A, we receive the EU's requirements as well as the SLA offers. In Stage B we address the security-level quantification that is associated with each SLA offer, then we use this data to serve as an input to the ranking algorithm based on fuzzy AHP in Stage C.

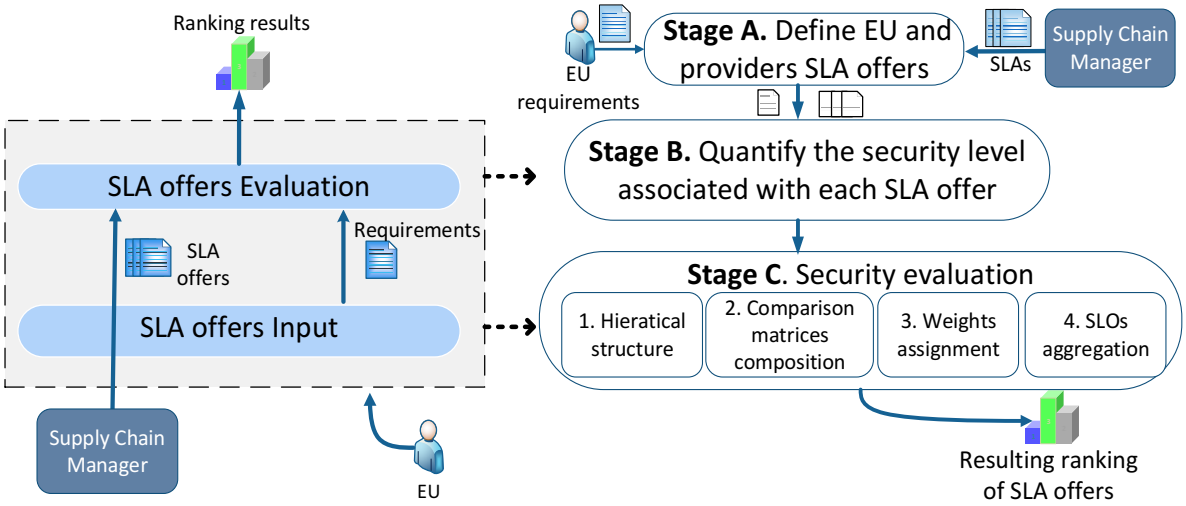

Figure 19. Fuzzy QHP: Methodology stages

Stage A. SLAs requirements specification

Fuzzy-QHP transforms the SLA offers into a hierarchical structure as shown in Figure 20. Following the SPECs negotiation process, EUs' will be provided with a set of SLA offers that include SPECS services that fulfil with their security requirements. CSPs control levels are also selected and used to rank SLA offers. In SPECS EUs are provided with CSPs controls while the Fuzzy-QHP is able to handle also SLO values to rank SLAs. With the hierarchical structure built for the fuzzy-QHP methodology, EUs have the ability to specify their security requirements (according to their expertise) at varied levels of the SLA representation (for example, the EU can specify his requirements not only at the control group level but also at the SLO level or both). For the sake of completion we will provide a description of the fuzzy-QHP methodology considering the complete SLA hierarchy as depicted in Figure 20.

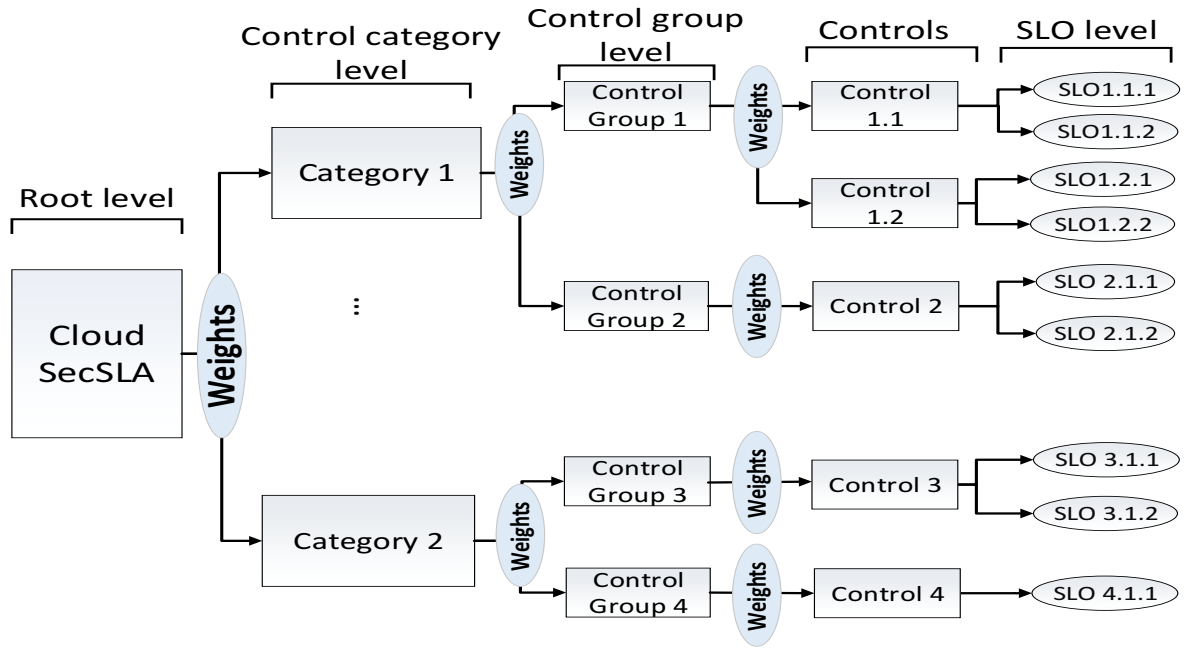

Figure 20. Cloud SLA hierarchy using fuzzy based QHP

This hierarchy can be used in two different ways: on one hand, as mentioned before, it can be used to represent SLA offers by defining SLOs at the lowest level of the hierarchy. On the other hand, the SLA hierarchy can also be used to define EUs' requirements by representing their relative importance at different levels of granularity.
The fuzzy-QHP methodology supports two types of EU security requirements: (a) qualitative, which are modelled as fuzzy numbers or (b) quantitative, which are assigned as values. For further explanation, we provide two examples at the SLO level: an SLO for "TLS Cryptographic Strength", which is composed of 8 possible values according to the ECRYPT II recommendations 2012[5] *{level1, level2, … , level8}*, such that *level8 is better than level1*. These metrics are then modelled as fuzzy numbers. For an SLO with two metrics defined using *yes/no* (as in the metric "Penetration testing activated"), the metrics are specified as Boolean *true/false* and modelled as fuzzy numbers.

Blended submission of different types of requirements for the same SLA offer is also supported in this methodology.

Stage B. Fuzzy security requirements quantification
To assess and compare the security levels provided by different SLA offers according to the EUs' fuzzy security requirements, the measurement model for different security SLOs is defined. Fuzzy requirements are represented by membership functions μ, which translate the vagueness and imprecision of EUs' requirements according to their security expertise.

In this study, the triangular fuzzy numbers (TFNs) are used to represent the fuzzy requirements. TFNs are used in the literature to capture the vagueness of the parameters. A TFN is graphically shown in Figure 21, where the TFN $\widetilde{M}$ is represented as *(l, m, u)*, l<m<u, in which the parameters *l, m,* and *u* respectively denote the smallest possible value, the most promising value, and the largest possible value that describe the fuzzy event (i.e., when *l=m=u*, the fuzzy number becomes a real number). Thus, a fuzzy number $\widetilde{M}$ on the set of real numbers

[5] http://www.keylength.com/en/3/

R is defined as a TFN if its membership function $\mu_{\widetilde{M}}(x)$, μ:*R*→*[0,1]*, whereas *x* is any positive real number and *l*<*m*<*u*, is equal to (as shown in Figure 21):

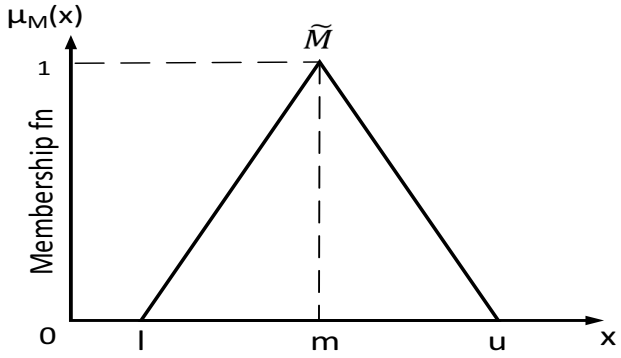

Figure 21. Triangular fuzzy number

$$\mu_{\widetilde{M}}(x) = \begin{cases} \frac{x-l}{m-l}, & if\ l \le x \le m, \\ \frac{u-x}{u-m}, & if\ m \le x \le u, \\ 0, & otherwise. \end{cases} \qquad (1)$$

This means a fuzzy set is specified as a TFN if (i) there exists only one element that the membership function $\mu_{\widetilde{M}}(x)= 1$ (at x = m) and (ii) $\mu_{\widetilde{M}}(x)$ is a continuous function. Table 8 details the terms used in this description.

Term	Definition
k_i	Security SLO i, such that i ∈ {1,2,....,j}, where j is the number of SLOs.
SLA_j	SLA offer j, such that j ∈ {1,2,....,n} , where n is the number of SLA offers.
$\widetilde{V}_{j,i}$	Value of SLO k_i provided by SLA_j, which is defined as TFN (l_i, m_i, u_i) using its membership function $\mu_{\widetilde{V}_{j,i}}(x)$.
$\widetilde{V}_{EU,i}$	EU's required value of SLO k_i defined as TFN.
$SLA_{p,i}/SLA_{q,i}$	Indicates the relative rank of SLA_p over SLA_q regarding SLO k_i, such that p and q ∈{1,2,....,n}, where n is the number of SLAs and p≠q.
$SLA_{1,i}/EU_i$	Indicates the relative rank of $SLA_{1,i}$ over EU_i, which specifies if SLA_1 satisfies EU requirements, with respect to k_i.

Table 8. Definition of terms used in the fuzzy QHP

The relationship between SLA offers (or SLA offers and EU) with respect to security SLO k_i with values $\tilde{V}_{k,i}$ and $\tilde{V}_{j,i}$ is represented as a ratio:

$$SLA_{1,i}/SLA_{2,i} = \tilde{V}_{1,i}/\tilde{V}_{2,i} \qquad (2)$$

such that

$$SLA_{1,i}/SLA_{2,i} = \begin{cases} (1,1,1), & \widetilde{V}_{1,i} \equiv \widetilde{V}_{2,i} \\ (l_{12}, m_{12}, u_{12}), & \text{otherwise} \end{cases}$$

where

$$l_{12} = \frac{l_1}{u_2},\ m_{12} = \frac{m_1}{m_2} \text{ and } u_{12} = \frac{u_1}{l_2}.$$

The following example illustrates the security requirement quantification using TFN. Consider an SLO, termed as k_1, specified in Stage A, that is composed of three metrics values that are

defined using the notion of security levels (*level₃*, *level₂*, *level₁*). These security levels are respectively modelled as fuzzy numbers which are calculated as TFN as shown in Table 8. Consider two SLAs, SLA_1 and SLA_2 providing SLO k_1 with *level₃* and *level₂* respectively. This means SLA_1 and SLA_2 are offering k_1 with values $\tilde{V}_1=\tilde{3} \equiv$ *level₃ and* $\tilde{V}_2=\tilde{2} \equiv$ *level₂*. Moreover, the EU requires *level₃* regarding SLO k_1 so that $\tilde{V}_u=\tilde{3} \equiv$ *level₃*. Thus, using Equation 2 and the terms defined in Table 1, the relative rank of SLA_1 over EU is defined as: $SLA_1/EU = \tilde{3}/\tilde{3} = (1,1,1)$. Therefore, SLA_1 is satisfying the EU requirement. Moreover, the relative rank of SLA_2 over EU is defined as: $SLA_2/EU = \tilde{2}/\tilde{3}$.

Stage C. Security evaluation based on fuzzy-AHP

In the conventional Analytic Hierarchy Process (AHP), the pairwise comparisons for each level with respect to the goal of the best alternative selection are conducted using a nine-point scale. However, according to [9]: (1) The AHP method is mainly used in nearly fixed decision applications, (2) the AHP method creates and deals with a very unbalanced scale of judgment, (3) the AHP method does not take into account the uncertainty associated with the mapping of one's judgment to a number, and (4) the subjective judgment, selection, and preference of decision makers have great influence on the AHP results. Furthermore, it is also recognized that the human assessment of qualitative attributes is always subjective. Generally, it is impossible to reflect the decision makers' uncertain preferences through fixed values. Therefore, fuzzy-AHP is to relieve the uncertainty and inability of the AHP in handling linguistic variables. The fuzzy-AHP approach allows a more accurate description of the decision-making process, where fuzzy set theories are used to express the uncertain comparison judgments as fuzzy numbers. There are several procedures to attain CSPs ranking in fuzzy-AHP, in this deliverable the methodology of fuzzy-AHP based on Chang's extent analysis [10] is utilized (Appendix II provides the details of this methodology). The proposed security evaluation method consists of four main phases, as shown in Figure 19.

Phase 1. Structuring decision hierarchy. Similar to conventional AHP, the first step is to break down the complex decision-making problem into a hierarchical structure. The SLA offers are constructed as a hierarchical structure as specified in Stage A and represented in Figure 20. The hierarchical structure defines the structure of cloud SLAs from the highest level (the Root level, which defines the main goal and aims to find the overall rank) to the lowest level (the control level).

Phase 2. Linguistic weights assignment. In order to compare two SLA offers' security SLOs, the relative importance level of the EU's requirements for each security SLO should be assigned as weights, as shown in Figure 20. We utilize linguistic terms to specify the importance of each SLO and the uncertainty of the EU needs, as shown in Figure 22 and Table 9. Thus, novice EUs can assign linguistic terms at the Control category level or at the Control group level without specifying the lowest level attributes (which requires an extremely high level of expertise). Furthermore, in order to let EUs adopt cloud services, it would be desirable to let them express their general requirements or preferences in a descriptive manner.
To address this issue, we consider the *assignment of linguistic terms by EUs.* EUs can assign fuzzy linguistic terms as weights to indicate their priorities. The number of possible terms depends on the level of accuracy required for the analysis. A great number of levels will result on a more accurate analysis but will force the EU to be more precise when defining his preferences. For the current description we will use a seven-level scale as follows: Extremely-Important (EI), Highly-Important (HI), Important (I), Low-Important (LI), Not-Important (NI), Not- Required (NR), and Do-not-know (Dk). These labels define uncertain requirements of the EUs, and are

represented as TFNs, as shown in Figure 22 and Table 9. The proposed framework allows the EUs to: (i) assign linguistic weights at varied levels of the hierarchical specification, and ii) individually adjust the linguistic terms according to their requirements. To further ease the task, especially for novice EUs, the system can set default values for each linguistic requirement, according to the specified SLO specified in Figure 22 and Table 9.

Extremely-Important denotes that all security SLOs are mandatory requirements for the EU. Not-Required (NR) indicates that the security SLOs are not required by the EU. Not-Important, Low-Important, and Highly-Important specify the EU's different degrees of requirements importance where the EU can accept varied values specifying several degrees of importance that depend on the considered scale. Do-not-know specifies the EU's unknown requirements. In our model, we represent Do-not-know as TFN that can have all possible ranges from 1 to 9 thus we defined it as (1, 5, 9), which means the most promising value is 5, that is the ordinate of the highest intersection point between Low-Important and Important.

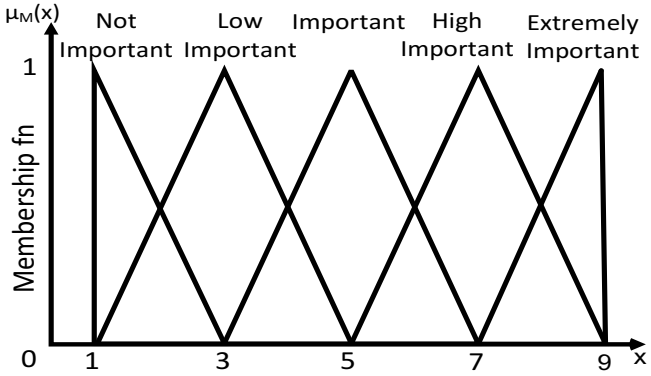

Figure 22. Linguistic terms for criterion importance

Linguistic scale for importance	Fuzzy numbers	Membership function	Domain	TFN (*l*, *m*, *u*)
Not-Important (NI)	$\tilde{1}$	$\mu_{\tilde{M}}(x) = \frac{3-x}{3-1}$	$1 \leq x \leq 3$	(1, 1, 3)
Low-Important (LI)	$\tilde{3}$	$\mu_{\tilde{M}}(x) = \frac{x-1}{3-1}$ $\mu_{\tilde{M}}(x) = \frac{5-x}{5-3}$	$1 \leq x \leq 5$	(1, 3, 5)
Important (I)	$\tilde{5}$	$\mu_{\tilde{M}}(x) = \frac{x-3}{5-3}$ $\mu_{\tilde{M}}(x) = \frac{7-x}{7-5}$	$3 \leq x \leq 7$	(3, 5, 7)
High-Important (HI)	$\tilde{7}$	$\mu_{\tilde{M}}(x) = \frac{x-5}{7-5}$ $\mu_{\tilde{M}}(x) = \frac{9-x}{9-7}$	$5 \leq x \leq 9$	(5, 7, 9)
Extremely-Important (EI)	$\tilde{9}$	$\mu_{\tilde{M}}(x) = \frac{x-7}{9-7}$	$7 \leq x \leq 9$	(7, 9, 9)
Do-not-know (Dk)	$\tilde{4}$	$\mu_{\tilde{M}}(x) = \frac{x-1}{5-1}$ $\mu_{\tilde{M}}(x) = \frac{9-x}{9-5}$	$1 \leq x \leq 9$	(1, 5, 9)

Table 9. Linguistic variables describing weights of the criteria and values of ratings

Phase 3. Pairwise fuzzy comparison matrices. The process of modelling values to a quantitative meaningful metric denoting the specified security level is not straightforward, as SLOs can have various types of values. Therefore, we used a relative-ranking model based on a pairwise comparison matrix of security SLOs provided by different SLA offers and required by EUs using TFNs. Thus, pairwise comparison judgments are represented by triangular fuzzy numbers indicating the relative rank between two providers or a provider and a EU such that $\tilde{a}_{ij} = (l_{ij}, m_{ij}, u_{ij})$, whereas $l_{ij} = \frac{l_i}{l_j}$, $m_{ij} = \frac{m_i}{m_j}$, and $u_{ij} = \frac{u_i}{u_j}$ (cf. Equation 2). As in the conventional AHP, using a comparison matrix $\tilde{A} = \tilde{a}_{ij}$ for each SLA and the CSC, we obtain a one-to-one comparison of each SLA and EU for a particular SLO. This will result in a comparison matrix of size *(n + 1)* x *(n + 1)* if there is a total of n CSPs and one EU. Such that $\tilde{a}_{ij} = \frac{1}{\tilde{a}_{ji}} = \left(\frac{1}{\tilde{u}_{ji}}, \frac{1}{\tilde{m}_{ji}}, \frac{1}{\tilde{l}_{ji}}\right)$

$$\tilde{A} = \begin{array}{c} \\ 1 \\ 2 \\ \dots \\ n \\ n+1 \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & \dots & n \quad n+1 \end{array} \\ \begin{pmatrix} \tilde{a}_{11} & \tilde{a}_{12} & \dots & \tilde{a}_{1n} \quad \tilde{a}_{1u} \\ \tilde{a}_{21} & \tilde{a}_{22} & & \tilde{a}_{2n} \quad \tilde{a}_{2u} \\ \vdots & & \ddots & \vdots \\ \tilde{a}_{n1} & \tilde{a}_{n2} & \dots & \tilde{a}_{nn} \quad \tilde{a}_{nu} \\ \tilde{a}_{u1} & \tilde{a}_{u2} & & \tilde{a}_{un} \quad \tilde{a}_{uu} \end{pmatrix} \end{array}$$

Where $\tilde{a}_{12} = SLA_1/SLA_2$, which indicates the relative rank of SLA_1 over SLA_2 as indicated in Table 9, so that:

$$\tilde{A} = \begin{array}{c} \\ SLA_1 \\ SLA_1 \\ \dots \\ SLA_n \\ EU \end{array} \begin{array}{c} \begin{array}{cccc} SLA_1 & & \dots & SLA_n \quad EU \end{array} \\ \begin{pmatrix} SLA_1/SLA_1 & \dots & SLA_1/SLA_n & SLA_1/EU \\ SLA_2/SLA_1 & & SLA_2/SLA_n & SLA_2/EU \\ \vdots & \ddots & \vdots & \\ SLA_n/SLA_1 & \dots & SLA_n/SLA_n & SLA_n/EU \\ EU/SLA_1 & & EU/SLA_n & EU/EU \end{pmatrix} \end{array} \qquad (3)$$

Next, the relative ranking of all the SLA offers and the EU for a particular SLO are calculated as a priority vector (PV) of the fuzzy comparison matrix $\tilde{A}$. The PV indicates a numerical ranking of providers that specifies an order of preference among them, as indicated by the ratios of the numerical values. There are several procedures to attain PV in fuzzy-AHP. The methodology based on Chang's extent analysis method [10] is the one utilized in the presented methodology. The PV is of the form:

$$PV_{k_i} = (N_1 \quad N_2 \quad \dots \quad N_n \quad N_u), \qquad (4)$$

where N_i, i=1,2, …, n, is a numerical value representing the relative rank of the SLA_i and with respect to the EU regarding an SLO k_i. Similarly, N_u is the relative rank of the EU required security level with respect to the security levels offered by the SLA offers.

Phase 4. SLOs Aggregation. In the final phase, we follow up with a bottom-up aggregation to give an overall assessment of the security levels and a final ranking of the SLA offers. To achieve that, the priority vector of each SLO (Phase 3) is aggregated with its relative normalized weight assigned in Phase 2. This aggregation process is repeated for all SLOs in the hierarchy with their relative weights, which results in the ranking of all the cloud providers based on EU- defined requirements and weights:

$$PV_{aggregated} = \left(PV_{k_1} \quad \dots \quad PV_{k_n}\right)(w_i) \qquad (5)$$

Here w_i is the EU-assigned weights of criteria *i* and PV_{ki} is the priority vector calculated for SLO k_i, i=1,2,…, n. The methodology presented in this section is validated using a case study presented in Appendix IV.

9. Conclusions

This document reports the final results with respect to the conceptual framework for the Cloud SLA negotiation. The main target of this deliverable is T2.3 that implements the negotiation components and processes designed. The results of this deliverable will also be used in dedicated WP1, WP4, and WP5 activities, and prototypes delivered at M24 and M30.

This deliverable reports the following results (and the evolution with respect to the initial report D2.2.1):

- The final version of the Negotiation module architecture, including the negotiation components (implemented in T2.3) and the high level interfaces that are the basis for the APIs that are completely defined in T1.3. Though the basics of the architecture have not changed with respect to D2.2.1, the interfaces and relationship with the rest of the modules of the SPECS framework have been defined in D2.2.2.
- The final version of the SLA specification. This is the basis for the definition of the content of the SLA and the relationship among the elements that comprise the SLA. The SLA is one of the main information structures used in SPECS, since it is used to define the service commitments signed with the EU. It is also used to trigger the enforcement of the security mechanisms included in the SLA (that will also trigger the monitoring and remediation activities). The new specification reported at M24 is compliant with the latest versions of the specifications (namely, the NIST RATAX and ISO/IEC 19086). The main changes comprise the introduction of the capability concept and the definition of the relationship between security metrics and SLOs. The latest conceptual model also defines the elements included in the SLA (including also non-functional properties such as the expiration time of the signed SLA).
- The machine readable specification for the SLA (based on the latest SLA specification reported above) is also provided. The changes with respect to the machine readable format reported in D2.2.1 comprise the introduction of new elements (such as capabilities) and properties, while the language used to represent it (WS-Agreement) is the same as in D2.2.1
- A new metric catalogue that contains new security metrics to be provided by SPECS services and developed during the second year (included in the signed SLA, enforceable and measurable in order to check their fulfilment) and metrics developed in WP5 for the ViPR service (to be reported in D5.3). Of course, the metric catalogue is compliant with the latest specification of the SLAs. An online version of the metric catalogue is also available[6] as reported in WP5.
- The final version of the negotiation process. The feedback from the implementation tasks and the integration of activities among Negotiation, Enforcement, and Platform are the main sources that have been used to design the new process as it is reported in D2.2.2. The new process draws also from the new specification of SLAs and the approach to gather EU's security requirements (defined in D5.1.3 as part of the SPECS Application).
- The final version of the renegotiation processes. At M24 two types of renegotiation have been identified and D2.2.2 reports the details of both processes. The new renegotiation processes are the result of the information received from the Enforcement module (especially in what regards remediation and implementation activities).
- Redefinition of the reasoning algorithms used to rank SLA offers during the negotiation process. On one hand, D2.2.2 details how the REM methodology (already reported in

[6] Security Metrics Catalogue Application: http://apps.specs-project.eu/specs-app-security_metric_catalogue/

D2.2.1) has been applied by using SLA model also reported in D2.2.2. On the other hand, the QHP methodology (already reported in D2.2.1) has been revised to include fuzzy logic in order to manage the uncertainty of qualitative requirements.

10. Bibliography

[1] A. Taha, R. Trapero, J. Luna, and N. Suri, "AHP-Based Quantitative Approach for Assessing and Comparing Cloud Security," Proc. of Trust, Security and Privacy in Computing and Communications, pp. 284–291, 2014.

[2] J. Luna, R. Langenberg, and N. Suri, "Benchmarking Cloud Security Level Agreements Using Quantitative Policy Trees," Proc. of Cloud Computing Security Workshop, pp. 103–112, 2012.

[3] Casola, V., Preziosi, R., Rak, M., & Troiano, L. (2005). A Reference Model for Security Level Evaluation: Policy and Fuzzy Techniques. J. UCS,11(1), 150-174.

[4] O.Hussain, F.Hussainetal., "Iaas Cloud Selection using mcdm methods," Proc. of International Conference on e-Business Engineering, pp. 246– 251, 2012.

[5] S. Garg, S. Versteeg, and R. Buyya, "Smicloud: A framework for comparing and ranking cloud services," Proc. of Utility and Cloud Computing, pp. 210–218, 2011.

[6] F. Hussain, O. Hussain et al., "Towards multi-criteria cloud service se- lection," Proc. of Innovative Mobile and Internet Services in Ubiquitous Computing, pp. 44–48, 2011. M.Menzel and R.Ranjan, "Cloudgenius: decision support for web server cloud migration," Proc. of World Wide Web, pp. 979–988, 2012.

[7] J. Mendel, "Fuzzy logic systems for engineering: a tutorial," Proc. of IEEE, vol. 83, no. 3, pp. 345–377, 1995.

[8] C. Qu and R. Buyya, "A cloud trust evaluation system using hierarchical fuzzy inference system for service selection," Proc. of Advanced Information Networking and Applications, pp. 850–857, 2014.

[9] G. Kabir and M. Hasin, "Comparative analysis of AHP and fuzzy AHP models for multicriteria inventory classification," Journal of Fuzzy Logic Systems, pp. 1–16, 2011.

[10] Y. Chang, "Applications of the extent analysis method on fuzzy AHP," European journal of operational research, pp. 649–655, 1996.

[11] Cloud Security Alliance, "The Consensus Assessments Initiative Questionnaire," Online: https://cloudsecurityalliance.org/research/cai/, 2011.

[12] "Guidelines on information security controls for the use of Cloud computing services based on ISOIEC 27002," International Organization for Standardization, Tech. Rep. ISOIEC 27002, 2014.

[13] "Security and Privacy Controls for Federal Information Systems and Organizations," National Institute of Standards and Technology, Tech. Rep. NIST 800-53v4, 2014.

[14] "Cloud Service Level Agreement Standardisation Guidelines," European Commission, C-SIG SLA, Tech. Rep. C-SIG SLA 2014, 2014.

[15] "(Draft) Cloud Computing: Cloud Service Metrics Description," NIST, Tech. Rep. NIST, 2014.

[16] J. Luna, A. Taha, R. Trapero and N. Suri, "Quantitative Reasoning About Cloud Security Using Service Level Agreements," IEEE Transactions on Cloud Computing. (to be published)

[17] "Cloud Security Alliance. Security, Trust & Assurance Registry (STAR)," Online: https://cloudsecurityalliance.org/star/, 2011.

[18] "Cloud Security Alliance. Cloud Control Matrix", Online: https://cloudsecurityalliance.org/group/cloud-controls-matrix/. 2014

[19] V. Casola, A. Mazzeo, N. Mazzocca, and V. Vittorini, "A policy-based methodology for security evaluation: A security metric for public key infrastructures," Journal of Computer Security, vol. 15, no. 2, pp. 197–229, 2007.

Appendix I. Example of a specific SLA: CyptoBruteForceResistance

This appendix introduces concrete examples of a definition of a security SLAs that follows the conceptual model presented in Section 4.1. The example starts with a category and derives the associated controls, SLOs, metrics and the abstract metrics. The following diagram represents the complete security SLA hierarchy of the example.

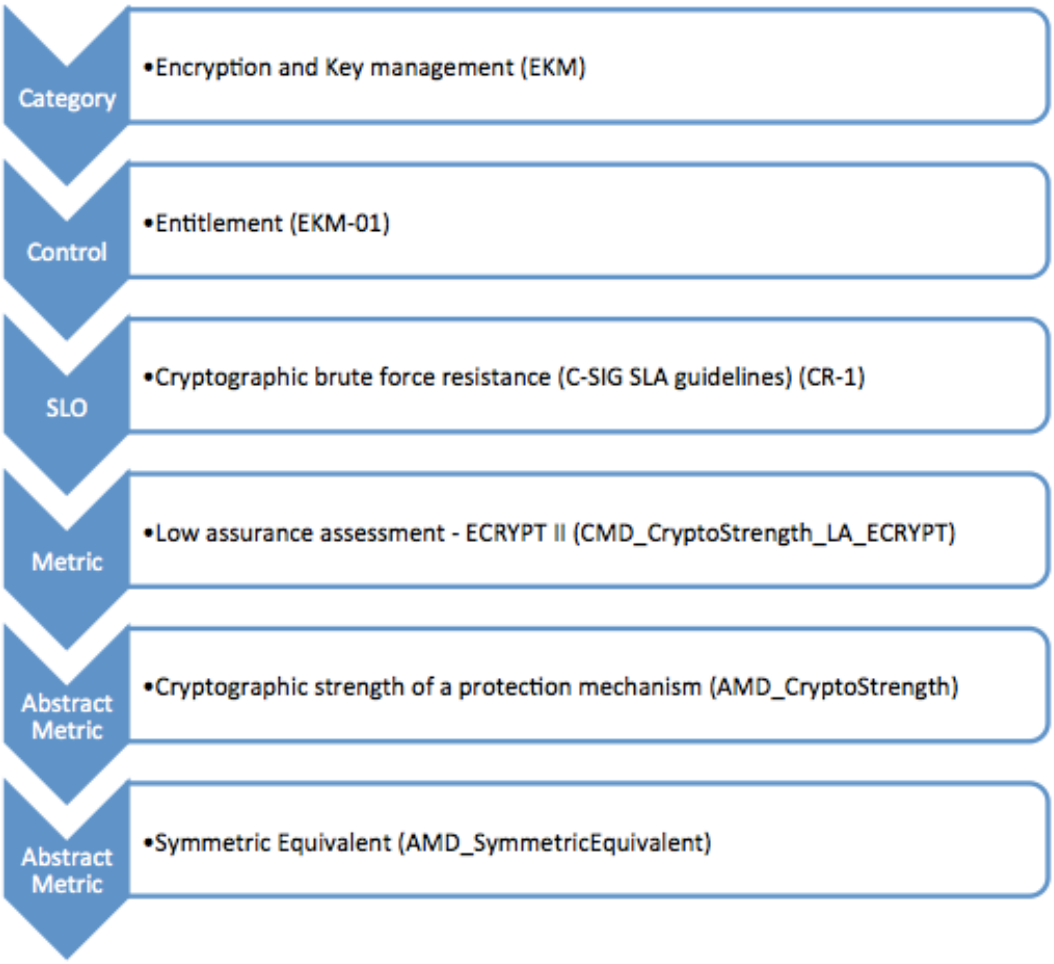

Figure 23. Example of a complete SPECS SLA hierarchy for an SLO

Applying the conceptual model described in Section 4.1 to the security SLO CryptoBruteForceResistance, the following table details the attributes of each element of the Security SLA hierarchy.

secSLO_CryptoBruteForceResistance
Control Category name: Encryption and Key Management referenceId: EKM controlFramework: CSA Cloud Controls Matrix v3.0 customerDefinedWeight: note: n/a
Security Control name: Entitlement referenceId: EKM-01

customerDefinedWeight:
Compensating Control name: n/a referenceId: n/a customerDefinedWeight: n/a note: n/a
Security Service Level Objective customerDefinedWeight: n/a objective: tls_crypto_strength_level ≥ M3_value note: expresses the strength of a cryptographic protection applied to a resource based on its key length, e.g. using the ECRYPT II security level recommendations or the FIPS security levels for encryption. This normalizing scale allows comparison of the strengths of different types of cryptographic algorithms.
Security Metric serviceLevel:

Table 10. Security SLO definition of CryptoBruteForceResistance

CMD_CryptoStrength_LA_ECRYPT of secSLO_CryptoBruteForceResistance
Metric name: Cryptographic Strength of a protection mechanism - Low assurance assessment – EECRYPT II. referenceId: CMD_CryptoStrength_LA_ECRYPT note: this metric provides a low security assurance (high uncertainty) method to assess the cryptographic strength of a resource. **Primary Abstract Metric** name: Cryptographic strength of a protection mechanism referenceId: AMD_CryptoStrength **Metric Rules** name: Configuration-based assessment (Assessment method) referenceId: AMR_Assessment_CryptoStrength definition: The value associated to the parameter "Security Bits (Symmetric Equivalent)" is obtained by performing look up at the configuration/properties file. This assessment method is associated with a low security assurance (high uncertainty). note: A Concrete Metric MUST specify the assessment method **Metric Parameters** name: Security Levels (Security Bits Equivalent) referenceId: AMP_CryptoStrength definition: This parameter refers to the mapping between "security levels" and corresponding "security bits" note: The parameter must be specified in form of a list of couples ["security levels":"security bits"]

Table 11. Metrics definition for the security SLO CryptoBruteForceResistance

AMD_CryptoStrength of secSLO_CryptoBruteForceResistance
Abstract Metric name: Cryptographic strength of a protection mechanism referenceId: AMD_CryptoStrength unit: Security Level (1 … 8) scale: Qualitative expression: The cryptographic strength (security level) is computed based on the security bits defined by the underlying abstract metric "Symmetric Equivalent". For this purpose is used the

ECRYPT II mapping[7] shown in following table:

Security Level	Security bits (symmetric equivalent)
1	32
2	64
3	72
4	80
5	96
6	112
7	128
8	256

For computing the "Security bits" associated to the cloud resource under evaluation, please refer to the underlying abstract metric definition below.

definition: This abstract metric expresses the strength of a cryptographic protection applied to a resource based on its key length, using the ECRYPT II security level recommendations for encryption. Instead of using key lengths alone, which are not always directly comparable from one algorithm to another, this normalizing scale allows comparison of the strengths of different types of cryptographic algorithms.

note: This metric is related to C-SIG SLA standardization guidelines' CR-1 (Cryptographic brute force resistance) SLO

Abstract Metric Rule Definitions

name: Assessment method.

referenceId: AMR_Assessment_CryptoStrength

definition: This rule defines how to assess/measure the strength of the cryptographic mechanism. Each assessment method can be associated with a different level of assurance. The following methods are possible {configuration_file_lookup,runtime_test}

note: A Concrete Metric MUST specify the assessment method.

Abstract Metric Parameter Definitions

name: Security Levels (Security Bits Equivalent)

referenceId: AMP_CryptoStrength

definition: This parameter refers to the mapping between "security levels" and corresponding "security bits"

note: The parameter must be specified in form of a list of couples ["security levels":"security bits"]

underlyingAbstractMetrics

name: Symmetric Equivalent

referenceId: AMD_SymmetricEquivalent

Table 12. Abstract metric definition for the security SLO CryptoBruteForceResistance

[7] ECRYPT II recommended key sizes (symmetric equivalent), please refer to Table 7.4 in http://www.ecrypt.eu.org/documents/D.SPA.20.pdf

Appendix II. Foundations of Fuzzy Logic: the fuzzy inference system

The fuzzy inference system (FIS) is a prominent application of fuzzy logic and fuzzy sets theory. FIS is used to solve reasoning problems in uncertain environments due to its ability to handle inaccurate and imprecise inputs ([7] [8]). We further detail the main building blocks of as shown in Figure 24:

- **Rules:** are expressed as a collection of *if-then* statements that define the inference model, e.g., "**If** x_1 is *warm* **then** y_1 is *quite low*". The rule structure is: **if** *antecedent* **then** *consequent*, where antecedent and consequent are fuzzy propositions. These rules help in quantifying linguistic variables (e.g., x_1 may have a finite number of linguistic variables associated with it, ranging from extremely warm to extremely cold), by using fuzzy membership functions. Additionally we can combine multiple rules using AND or OR operators.
  That is, when the system is applied to a particular situation (a given input), all rules are fired in parallel (applied all at once to this given input), and for each rule its conclusion is computed. The computation takes into account the degree in which the antecedent is satisfied in such a way that if it is not at all satisfied, the conclusion is the empty set.
- **Membership function:** defines to which degree the fuzzy element belongs to the corresponding fuzzy set. It maps specific real values to membership degrees between 0 and 1. In a fuzzy inference system, each input and output variable has its own set of membership functions.
- **Fuzzifier:** comprises the process of transforming crisp input values into the membership functions to obtain corresponding membership degrees for each fuzzy input sets.
- **Inference engine:** defines the fuzzy logic operators and handles the way in which rules are combined in order to aggregate fuzzy output sets.
- **Defuzzifier:** maps the aggregated fuzzy output sets into crisp values (usually a numerical value) using the output membership functions. This process is called defuzzification and can be seen as either an element selection from a set (in fact, from a fuzzy set), or a fusion process in which the information to be fused is the fuzzy set and the outcome is the numerical value.

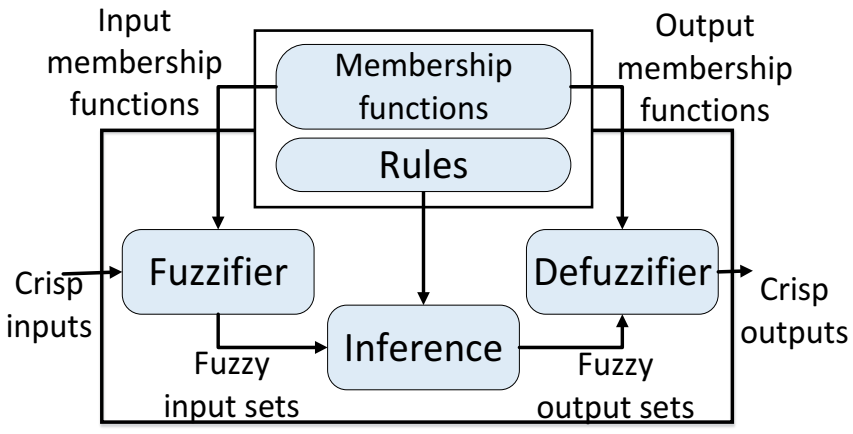

Figure 24. Fuzzy inference system.

Appendix III. Principles for handling fuzzy Analytic Hierarchy Processes

The following appendix outlines Chang's [10] extent analysis method on fuzzy-AHP. We will explain Chang's method using an example of two TFNs (l_1, m_1, u_1) and (l_2, m_2, u_2), which represent a provider security-level and a user requirement for a particular SLO. The process start by calculating the comparison matrix,

$$\tilde{A}= \begin{array}{c} \\ 1 \\ 2 \\ \dots \\ n \\ n+1 \end{array} \begin{array}{c} \begin{array}{cccccc} 1 & 2 & \dots & n & n+1 \end{array} \\ \begin{pmatrix} \tilde{a}_{11} & \tilde{a}_{12} & \dots & \tilde{a}_{1n} & \tilde{a}_{1u} \\ \tilde{a}_{21} & \tilde{a}_{22} & & \tilde{a}_{2n} & \tilde{a}_{2u} \\ \vdots & & \ddots & \vdots & \\ \tilde{a}_{n1} & \tilde{a}_{n2} & \dots & \tilde{a}_{nn} & \tilde{a}_{nu} \\ \tilde{a}_{u1} & \tilde{a}_{u2} & & \tilde{a}_{un} & \tilde{a}_{uu} \end{pmatrix} \end{array}, \text{being } \tilde{a}_{ij} = \frac{1}{\tilde{a}_{ij}} = \frac{1}{u_{ji}}, \frac{1}{m_{ji}}, \frac{1}{l_{ji}}$$

The resulting comparison matrix for the example is:

$$\tilde{A}= \begin{array}{c} \\ i=1 \\ i=2 \end{array} \begin{array}{c} \begin{array}{cc} j=1 & j=2 \end{array} \\ \begin{pmatrix} (1,1,1) & (l_{12},m_{12},u_{12}) \\ (l_{21},m_{21},u_{21}) & (1,1,1) \end{pmatrix} \end{array}$$

After $\tilde{A}$ calculation, the steps of Chang's extent analysis to attain the PV are detailed as follows:

Step 1. The value of the fuzzy synthetic extent with respect to i[th] object is calculated such that:

$$S_i=\left(\sum_{j=1}^{m} l_j , \sum_{j=1}^{m} m_j , \sum_{j=1}^{m} u_j\right)\otimes\left(\frac{1}{\sum_{i=1}^{n} u_j}, \frac{1}{\sum_{i=1}^{n} m_j}, \frac{1}{\sum_{i=1}^{n} l_j}\right) \quad (6)$$

Whereas ⊗ denotes fuzzy multiplication, i = 1 . . . n, and j = 1 . . . m. We explain this step using the two considered TFNs' comparison matrix $\tilde{A}$ (m = n = 2) so that:

$$S_1=(1+l_{12},1+m_{12},1+u_{12})\otimes\left(\frac{1}{1+u_{12}+1+u_{21}}, \frac{1}{1+m_{12}+1+m_{21}}, \frac{1}{1+l_{12}+1+l_{21}}\right)$$
$$S_2=(1+l_{21},1+m_{21},1+u_{21})\otimes\left(\frac{1}{1+u_{12}+1+u_{21}}, \frac{1}{1+m_{12}+1+m_{21}}, \frac{1}{1+l_{12}+1+l_{21}}\right)$$

By the end of this step, M_1 and M_2 will be represented as TFN with values (l_1, m_1, u_1) and (l_2, m_2, u_2).

Step 2. The degree of possibility of $M_2 = (l_2, m_2, u_2) \geq M_1 = (l_1, m_1, u_1)$ is defined as $V(M_2 \geq M_1)= \sup[\min(\mu_{M1}(x),\mu_{M2}(x))]$ (as shown in Figure 25) and is represented as follows:

$$V(S_2 \geq S_1)= \begin{cases} 1, & \text{if } m_2 \geq m_1 \\ 0, & \text{if } l_1 \geq u_2 \\ \frac{l_1-u_2}{(m_2-u_2)(m_1-l_1)}, & \text{otherwise} \end{cases} \quad (7)$$

Where d is the ordinate of the highest intersection point *D* between μ_{M1} and μ_{M2} (see Figure 25). For the comparison we need the values of both of $V(S_1 \geq S_2)$ and $V(S_2 \geq S_1)$.

Step 3. The degree possibility for a fuzzy number to be greater than k fuzzy numbers S_i where i = 1,2,...,k can be defined by:

$$V(S \geq S_1, S_2, \ldots, S_k) = V[(S \geq S_1), (S \geq S_2), \ldots, (S \geq S_k)] = \min\left(V(S \geq S_i)\right) \; i = 1, 2, \ldots, k \quad (8)$$

Assuming that $d'(A_i) = \min(V(S_i \geq S_k))$, for k = 1, 2, ..., n; $k \neq i$. Then the priority vector is given by $PV_0 = (d'(A_1), d'(A_2), \ldots, d'(A_n))^T$ where A_i(i = 1,2,...,n) are n elements.

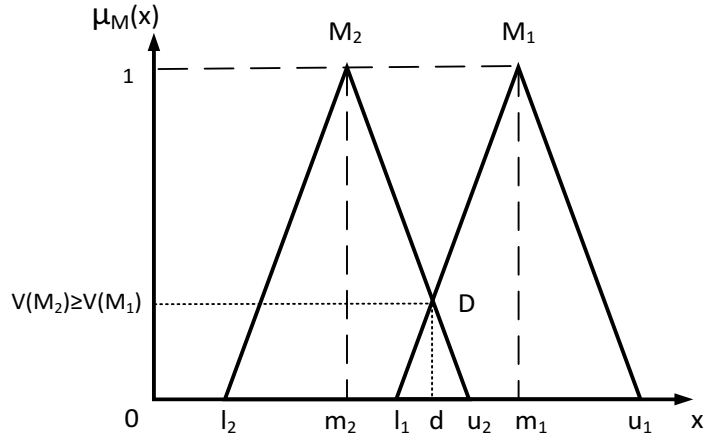

Figure 25. The intersection between M_1 and M_2

Step 4. Via normalization, the normalized priority vectors are $PV = (d(A_1), d(A_2), \ldots, d(A_n))^T$ where PV is a non-fuzzy number that gives priority weights of an attribute with respect to other attributes.

At the end of Step 4, we attain the priority vector of the fuzzy comparison matrix for a particular SLO. This method is done for all the SLOs' matrices. After this step, the priority vector of each SLO is aggregated with EU-assigned weights, as in Phase 4.

Appendix IV. Fuzzy-QHP: a case study

This appendix will validate the proposed methodology described in Section 8.2 by applying it to a set of SLAs for a given service. The case study will assume a secure web server service. According to the negotiation process, the supply chain manager will provide the reasoner with a set of SLA offers that will be ranked according to EU's security requirements. The reasoner will evaluate the SLA offers according to the requirements set both to the controls implemented by the SPECS services and to the controls implemented by CSPs.

In order to show the possibilities of the fuzzy-QHP model we will extend the CSPs' security controls by considering requirements at the SLO level of the SLA hierarchy (cf., Figure 20). This will allow us to consider also requirements different to *yes/no* answers as it used in SPECS at the control level. Table 13 shows the SLA for each SLA offer and the EUs' requirements for the two use cases shown in this appendix. Four SLAs will be compared (SLA1, SLA2, SLA3, and SLA4), and each SLA will include a different CSP. The comparison will consider two EU (EU1 and EU2) with different requirements and different expertise. According to the EU's expertise requiring the evaluation, the validation will consider two cases:

- *Case I*. The SLAs (SLA1, SLA2, SLA3 and SLA4) will be evaluated according to an expert EU (EU1) giving a detailed specification of low-level requirements (either linguistic or numerical requirements).
- *Case II*. The SLAs (SLA1, SLA2, SLA3 and SLA4) will be evaluated according to an EU (EU2) specifying linguistic weights at three different levels of granularity (corresponding to the hierarchy shown in Figure 20) namely Control category, Control group, Controls and SLO levels. This is the case of a customer having expert knowledge of only some controls (specified at the low level), no knowledge for other controls (specified at the Control category level), or specifying at the intermediate level according to the knowledge she/he has.

Cloud SLA				***SLAs***				***End users (EU)***	
Control category	***Control group***	***Control***	***SLO***	***SLA1***	***SLA2***	***SLA3***	***SLA4***	***EU1***	***EU2***
Supply Chain Management, Transparency and Accountability (STA)	Data Quality and Integrity	STA-01	SLO1	$level_3$	$level_4$	$level_3$	$level_4$	$level_4$	HI
	Network / Infrastructure Services	STA-03	SLO2	$level_3$	$level_4$	$level_2$	$level_4$	$level_4$	
Data security and information lifecycle management (DSI)	Information leakage	DSI-05	SLO3	yes	yes	yes	no	yes	Dk
Governance and risk management (GRM)	Data focus risk management	GRM-02	SLO4	no	yes	yes	yes	yes	yes
	Risk management framework	GRM-11	SLO5	$level_3$	$level_3$	$level_3$	$level_3$	$level_3$	$level_3$

Table 13. Fuzzy QHP case study: excerpt of SLAs and customer requirements

In order to evaluate the SLAs' with respect to the customer's requirements, we proceed to apply the fuzzy QHP methodology presented in section 8.2. For the case-study calculations we note the following:

1. Linguistic weights are specified as TFN as shown in Table 9.

2. Numerical requirements are specified, as TFN such that yes and no are denoted as (7 9, 9) and (1, 1, 1); similarly 48, 24, and 12 are calculated as (7, 9, 9), (5, 7, 9), and (3, 5, 7). Furthermore, levels 1, 2, 3, 4, and 5 are represented respectively using TFN by the membership function $\mu_{\widetilde{M}}(x) = \tilde{1}, \tilde{3}, \tilde{5}, \tilde{7}$, and $\tilde{9}$ (as defined in Table 9).
3. All CSPs' security SLOs are normalized to the customer requirements to eliminate masquerading. The masquerading effect happens when the overall aggregated security level values mostly depend on those security controls with a high-number of SLOs, thus negatively affecting groups with fewer although possibly more critical provisions. Other methodologies for the Cloud security assessment suffer from this effect.

Case study I: Expert EU1 details requirements at the lowest level

For this case, the end user (EU) specifies his requirements at the lowest level of the SLA hierarchy (i.e., SLOs) and considers the same relative importance (i.e., weights) for all of these. For the "Supply Chain Management, Transparency and Accountability (STA)" category there are two controls categories, which are further divided into another control (STA-01 and STA-03). Each control has one SLO (SLO1 for STA-01 and SLO2 for STA-03). For SLO1 the providers and the EU can specify their metrics from $level_1$ to $level_5$. Using the data shown in Table 3, Equation 2 is used to define the SLO1 pairwise relation such that:

$$SLA_1/SLA_2 = \tilde{5}/\tilde{7} = (\frac{3}{9}, \frac{5}{7}, \frac{7}{5}), \quad SLA_2/SLA_1 = \tilde{7}/\tilde{5} = (\frac{5}{7}, \frac{7}{5}, \frac{9}{3})$$

$$EU/SLA_3 = \tilde{7}/\tilde{5} = (\frac{5}{7}, \frac{7}{5}, \frac{9}{3}), \quad EU/SLA_4 = \tilde{7}/\tilde{7} = (1,1,1)$$

Therefore, the comparison matrix of STA-01 $\widetilde{A}_{SLO1}$ is:

$$\tilde{A}_{SLO1} =$$

	SLA_1	SLA_2	SLA_3	SLA_4	EU
SLA_1	$(1,1,1)$	$(\frac{3}{9}, \frac{5}{7}, \frac{7}{5})$	$(1,1,1)$	$(\frac{3}{9}, \frac{5}{7}, \frac{7}{5})$	$(\frac{3}{9}, \frac{5}{7}, \frac{7}{5})$
SLA_2	$(\frac{5}{7}, \frac{7}{5}, \frac{9}{3})$	$(1,1,1)$	$(\frac{5}{7}, \frac{7}{5}, \frac{9}{3})$	$(1,1,1)$	$(1,1,1)$
SLA_3	$(1,1,1)$	$(\frac{3}{9}, \frac{5}{7}, \frac{7}{5})$	$(1,1,1)$	$(\frac{3}{9}, \frac{5}{7}, \frac{7}{5})$	$(\frac{3}{9}, \frac{5}{7}, \frac{7}{5})$
SLA_4	$(\frac{5}{7}, \frac{7}{5}, \frac{9}{3})$	$(1,1,1)$	$\frac{5}{7}, \frac{7}{5}, \frac{9}{3}$	$(1,1,1)$	$(1,1,1)$
EU	$(\frac{5}{7}, \frac{7}{5}, \frac{9}{3})$	$(1,1,1)$	$\frac{5}{7}, \frac{7}{5}, \frac{9}{3}$	$(1,1,1)$	$(1,1,1)$

Then, using Chang's extent analysis method explained in Appendix 1, we get the relative ranking of the Cloud providers for SLO1, which is given by the priority vector of $\widetilde{A}_{SLO1}$ (PV_{SLO1}). PV_{SLO1} is calculated as follows, using Step 1 in Appendix 1, we get the value of the fuzzy synthetic extent for $\widetilde{A}_{SLA01}$ such that:

$$S_{SLA_1} = (3,4.14,6.2)\otimes(\frac{1}{39.4},\frac{1}{25.69},\frac{1}{19.29}) = \begin{matrix} l_{SLA_1} & m_{SLA_1} & u_{SLA_1} \\ (0.0761, & 0.1613, & 0.3215) \end{matrix}$$

$$S_{SLA_2} = (4.43,5.8,9)\otimes(\frac{1}{39.4},\frac{1}{25.69},\frac{1}{19.29}) = \begin{matrix} l_{SLA_2} & m_{SLA_2} & u_{SLA_2} \\ (0.1124, & 0.2258, & 0.4667) \end{matrix}$$

$$S_{SLA_3} = (3,4.14,6.2)\otimes(\frac{1}{39.4},\frac{1}{25.69},\frac{1}{19.29}) = \begin{matrix} l_{SLA_3} & m_{SLA_3} & u_{SLA_3} \\ (0.0761, & 0.1613, & 0.3215) \end{matrix}$$

$$S_{SLA_4} = (4.43,5.8,9)\otimes(\frac{1}{39.4},\frac{1}{25.69},\frac{1}{19.29}) = \begin{matrix} l_{SLA_4} & m_{SLA_4} & u_{SLA_4} \\ (0.1124, & 0.2258, & 0.4667) \end{matrix}$$

$$S_{EU} = (4.43,5.8,9)\otimes(\frac{1}{39.4},\frac{1}{25.69},\frac{1}{19.29}) = \begin{matrix} l_{EU} & m_{EU} & u_{EU} \\ (0.1124, & 0.2258, & 0.4667) \end{matrix}$$

Afterwards, using Step 2 we get the degree of possibility so that:

$$V(S_{SLA_1} \geqslant S_{SLA_2}) = 0.7642\ ,\ \ V(S_{SLA_1} \geqslant S_{EU}) = 0.7642$$
$$V(S_{SLA_1} \geqslant S_{SLA_3}) = 0.91\ (\text{as } m_{SLA_1} \geqslant m_{SLA_3})$$

Then, the possibility for a fuzzy number to be greater than other fuzzy numbers is calculated using Step 3:

$$d'(A_{SLA_1}) = min(V(S_{SLA_1} \geqslant S_{SLA_2}, S_{SLA_3}, S_{SLA_4}, S_{EU}))$$
$$= min(0.7642, 1, 0.7642, 0.7642) = 0.7642$$

Similarly $d'(A_{SLA2})$, $d'(A_{SLA3})$, and $d'(A_{EU})$ are calculated using Steps 2 and 3:

$$d'(A_{SLA_2}) = min(1,1,1,1) = 1$$
$$d'(A_{SLA_3}) = min(1,0.7642,0.7642,0.7642) = 0.7642$$
$$d'(A_{SLA_4}) = min(1,1,1,1) = 1$$
$$d'(A_{EU}) = min(1,1,1) = 1$$

Thus, the *SLO1* priority vector PV is given by:

$$PV_{\text{SLO1}} = \begin{matrix} SLA_1 & SLA_2 & SLA_3 & SLA_4 & EU \\ (0.7642 & 1 & 0.7642 & 1 & 1) \end{matrix}$$

This reflects which of the SLAs provide the *SLO1* security SLO relative to other SLAs and to the customer requirements. After normalization, $PV_{\text{STA-01}}$ is:

$$PV_{\text{SLO1}} = \begin{matrix} SLA_1 & SLA_2 & SLA_3 & SLA_4 & EU \\ (0.1688 & 0.2208 & 0.1688 & 0.2208 & 0.2208) \end{matrix}$$

This means that both SLA_2 and SLA_4 equally satisfy EU's SLO1 requirement. However, SLA_1 and SLA_3 do not fulfil that requirement. Similarly, the priority vector of SLO2 is calculated using its

comparison matrix $\widetilde{A}_{SLO2}$. The STA priority vector is then premeditated by aggregating PV_{SLO1} and PV_{SLO2} with customer-defined normalized weights (w_{STA}) using Equation 5. As specified earlier, in Case I the customer considers the same relative importance (i.e., weights) for all of these SLOs, such that:

$$w_{STA} = \begin{array}{c} \\ \end{array} \begin{array}{c} \begin{array}{cc} SLO1 & SLO2 \end{array} \\ \begin{pmatrix} 0.5 & 0.5 \end{pmatrix} \end{array}$$

$$PV_{STA} = \begin{array}{c} \\ SLA_1 \\ SLA_2 \\ SLA_3 \\ SLA_4 \\ EU \end{array} \begin{array}{c} \begin{array}{cc} PV_{SLO1} & PV_{SLO2} \end{array} \\ \begin{pmatrix} 0.1688 & 0.1979 \\ 0.2208 & 0.2209 \\ 0.1688 & 0.1394 \\ 0.2208 & 0.2209 \\ 0.2208 & 0.2209 \end{pmatrix} \end{array} \begin{pmatrix} 0.5 \\ 0.5 \end{pmatrix}$$

Therefore,

$$PV_{STA} = \begin{array}{c} \\ \end{array} \begin{array}{c} \begin{array}{ccccc} SLA_1 & SLA_2 & SLA_3 & SLA_4 & EU \end{array} \\ \begin{pmatrix} 0.1834 & 0.2209 & 0.1541 & 0.2209 & 0.2209 \end{pmatrix} \end{array}$$

The priority vector for *SLO3* (belonging to the control *DSI-05)* is calculated the same way, such that:

$$PV_{SLO3} = \begin{array}{c} \\ \end{array} \begin{array}{c} \begin{array}{ccccc} SLA_1 & SLA_2 & SLA_3 & SLA_4 & EU \end{array} \\ \begin{pmatrix} 0.25 & 0.25 & 0.25 & 0 & 0.25 \end{pmatrix} \end{array}$$

This means that only SLA4 does not fulfil EU *SLO3* requirement. In a similar way the priority vectors aggregated for the category *GRM,* PV_{GRM}, is:

$$PV_{GRM} = \begin{array}{c} \\ SLA_1 \\ SLA_2 \\ SLA_3 \\ SLA_4 \\ EU \end{array} \begin{array}{c} \begin{array}{cc} PV_{SLO4} & PV_{SLO5} \end{array} \\ \begin{pmatrix} 0 & 0.2 \\ 0.25 & 0.2 \\ 0.25 & 0.2 \\ 0.25 & 0.2 \\ 0.25 & 0.2 \end{pmatrix} \end{array} \begin{pmatrix} 0.5 \\ 0.5 \end{pmatrix}$$

$$PV_{GRM} = \begin{array}{c} \\ \end{array} \begin{array}{c} \begin{array}{ccccc} SLA_1 & SLA_2 & SLA_3 & SLA_4 & EU \end{array} \\ \begin{pmatrix} 0.1 & 0.225 & 0.225 & 0.225 & 0.225 \end{pmatrix} \end{array}$$

The SLAs rankings according to the customer requirements at the Control group level are shown in Figure 26 and at the category level are shown in Figure 27. Finally, the priority vectors of *DSI, STA,* and *GRM* are aggregated to obtain the total priority vector, as shown in Figure 28.

$$PV_{total} = \begin{array}{c} \\ SLA_1 \\ SLA_2 \\ SLA_3 \\ SLA_4 \\ EU \end{array} \begin{array}{c} \begin{array}{ccc} PV_{STA} & PV_{DSI} & PV_{GRM} \end{array} \\ \begin{pmatrix} 0.1834 & 0.25 & 0.1 \\ 0.2209 & 0.25 & 0.225 \\ 0.1541 & 0.25 & 0.225 \\ 0.2209 & 0 & 0.225 \\ 0.2209 & 0.25 & 0.225 \end{pmatrix} \end{array} \begin{pmatrix} 0.3333 \\ 0.3333 \\ 0.3333 \end{pmatrix}$$

$$PV_{total} = \begin{array}{c} \begin{array}{ccccc} SLA_1 & SLA_2 & SLA_3 & SLA_4 & EU \end{array} \\ \begin{pmatrix} 0.1778 & 0.2319 & 0.21 & 0.1486 & 0.2319 \end{pmatrix} \end{array}$$

Consequently, only SLA_2 fulfils the customer's requirements, as shown in Figure 26. That was expected, as SLA1 is not offering *SLO4* and is under-provisioning *SLO1* and *SLO2*. SLA_3 is not fulfilling customer requirements for *SLO1* and *SLO2*. Moreover, SLA_4 is not providing *SLO3*. Only SLA2 fulfils customer's requirements and, as a result, SLA2 is the best matching provider according to the customer's requirements, followed by SLA3, as shown in Figure 26.

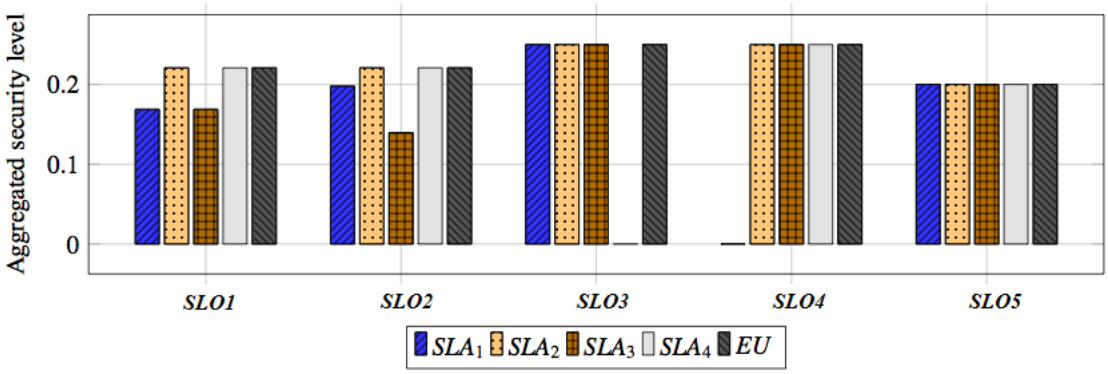

Figure 26. Aggregation at the SLO level regarding the customer's Case I requirements

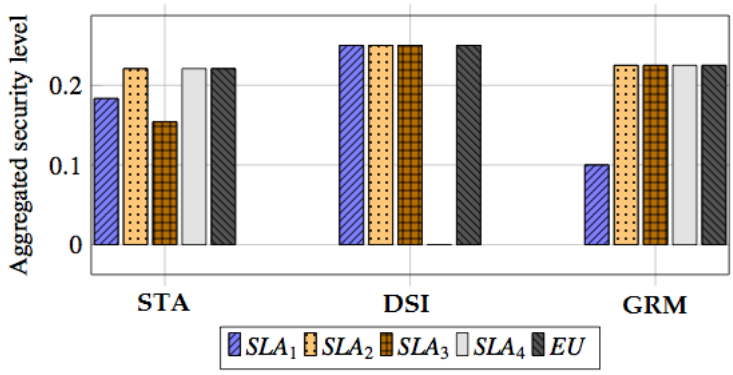

Figure 27. SLA's comparison with respect to customer Case I requirements at the Control category level

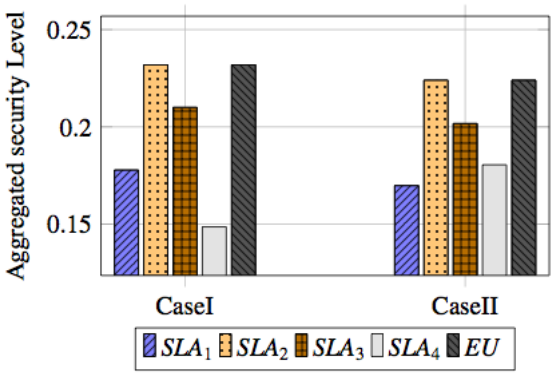

Figure 28. The total aggregated security level with respect to customer requirements

Case study II: Non expert EU2 specifies requirements and different levels
We assume the customer specifies linguistic weights at the Control category level, by denoting *High-Important* to *STA*. In addition to, denoting *Do-not-know* at the Control *DSI*. And similarly, as Case I the customer specifies low-level requirements for *GRM*, as shown in Table 13.
Since *STA* is assigned *EI*, the respective weight is set to (5, 7, 9), while the DSI weight is set to (1, 4, 7). Therefore, PV_{STA}, PV_{DSI} and PV_{GRM} are weights aggregated such that:

$$PV_{STA} = \begin{matrix} SLA_1 & SLA_2 & SLA_3 & SLA_4 & EU \\ \end{matrix} \begin{pmatrix} 0.1834 & 0.2209 & 0.1541 & 0.2209 & 0.2209 \end{pmatrix}$$

Since *DSI* is assigned *Dk,* the respective weight is set to (1, 4, 7), such that:

$$PV_{DSI} = \begin{matrix} SLA_1 & SLA_2 & SLA_3 & SLA_4 & EU \\ \end{matrix} \begin{pmatrix} 0.2261 & 0.2261 & 0.2261 & 0.0958 & 0.2261 \end{pmatrix}$$

Similarly, *GRM* is evaluated as explained in Case I: PV_{SLO4} and PV_{SLO5} are aggregated to obtain the *GRM* priority vector.

$$PV_{GRM} = \begin{matrix} SLA_1 & SLA_2 & SLA_3 & SLA_4 & EU \\ \end{matrix} \begin{pmatrix} 0.1 & 0.225 & 0.225 & 0.225 & 0.225 \end{pmatrix}$$

Finally, the priority vectors of *DSI, STA,* and *GRM* are aggregated to obtain the total priority vector.

$$PV_{total} = \begin{matrix} SLA_1 & SLA_2 & SLA_3 & SLA_4 & EU \\ \end{matrix} \begin{pmatrix} 0.1698 & 0.224 & 0.2017 & 0.1805 & 0.224 \end{pmatrix}$$

Therefore, only SLA2 satisfies the customer needs, whereas all SLA1, SLA3 and SLA4 do not fulfil customer requirements, as shown in Figure 28. That was expected, as *STA* is highly important to the *EU* and under provisioned by SLA3 and SLA1. Moreover, *SLO3* is not provided by SLA4. Thus, the presented framework can give accurate SLAs ranking even if the low level is not defined and vague preferences are specified at the highest levels, which means a customer can define weights at the higher levels instead of answering multiple low-level questions.