



*Secure Provisioning of Cloud Services
based on SLA Management*

SPECS Project - Deliverable D5.4

Identity Management-as-a-Service

Version no. 1.1
19 July 2016



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

Deliverable information

Deliverable no.:	D5.4
Deliverable title:	Identity Management-as-a-Service
Deliverable nature:	Prototype
Dissemination level:	Public
Contractual delivery:	19 July 2016
Actual delivery date:	19 July 2016
Author(s):	Miha Stopar (XLAB), Jolanda Modic (XLAB), Silvio La Porta (EMC), Andrew Byrne (EMC)
Contributors:	Massimiliano Rak (CeRICT), Valentina Casola (CeRICT), Alessandra De Benedictis (CeRICT)
Reviewers:	Silviu Panica (IeAT), Stefano Marrone (CeRICT)
Task contributing to the deliverable:	T5.4
Total number of pages:	54

Executive summary

The focus of the task T5.4 is to develop a complete application, which offers the secure storage services using the storage solutions offered by the EMC's ViPR application, together with an additional identity management and access control mechanisms.

The EMC's ViPR application, described in deliverable D5.3, integrates the SPECS framework with ViPR on the EMC testbed. Such solution enables to deliver storage in a software defined datacenter, according to SPECS security SLA life cycle.

In deliverables D5.2.1 and D5.2.2 we illustrated a SPECS application that is able to offer storage-as-a-service functionalities to home users and developers. Thanks to the end-to-end encryption that the Secure Storage application integrates, it was possible to offer a service that enable to easily manage data in cloud, protecting integrity and confidentiality.

This deliverable illustrates how it is possible to offer the storage functionalities integrating the storage capabilities offered by the software defined datacenter and the Secure Storage application, in order to have a security SLA than combines both functionalities. In addition, the secure storage functionalities are now protected with a dedicated AAA-as-a-Service mechanism that gives the name to the new SPECS application.

In particular, this document presents:

- AAAAaS application: We briefly summarize the user story and the associated validation scenarios that define the functionalities offered by the application.
- Application development process: We present the cloud service that is provisioned through the developed AAAaaS application and the security mechanisms that are integrated with the application to enforce and monitor the offered security guarantees. Moreover, we discuss the security metrics and controls that can be negotiated through the application and that imply specific configurations of the cloud service. Additionally, we present the SLA Template that specifies all attributes to enable the automated management of all processes associated to the delivery of the defined cloud service.
- Architecture: We present the design of the developed application and outline the roles of all integrated artefacts. Namely, we present the EMC solution ViPR, the integrated elements of the SPECS platform, and the security mechanisms.
- Performance and scalability analysis: We report the results of the performance and scalability analysis performed for the AAAaaS application.
- Installation, usage, testing: We present the installation and the usage guides and report all tests with which we verified the correctness of the implementation for the developed application.

Table of contents

Deliverable information	2
Executive summary.....	3
Table of contents.....	4
Index of figures	5
Index of tables	6
1. Introduction.....	7
2. Relationship with other deliverables	8
3. SPECS AAA-as-a-Service application description.....	9
3.1. User story: Secure Storage with Identity Management	9
3.2. Validation scenarios.....	9
3.2.1. Identity management set-up.....	10
3.2.2. User registration	11
3.2.3. User access internal account.....	12
3.2.4. User access external account.....	13
4. Application development process.....	15
4.1. Cloud service definition.....	15
4.2. Security mechanism preparation.....	15
4.3. SLA Template preparation	16
4.4. SLA Template and security mechanisms deployment.....	18
5. Architecture	19
5.1. Testbed.....	19
5.2. ViPR.....	20
5.3. Integration with SPECS framework	21
5.4. E2EE integration with AAA	22
6. Performance and scalability analysis	27
7. Installation	31
8. Usage	33
9. Testing	42
10. Conclusions	44
11. Bibliography	45
Appendix 1. The AAAaaS SLA Template	46

Index of figures

Figure 1. Relationship with other deliverables	8
Figure 2. SPECS Application development process	15
Figure 3. Physical architecture of the testbed.....	19
Figure 4. Integration of AAA security mechanism with SPECS framework.....	21
Figure 5. E2EE Client sign-on	23
Figure 6. Login with SPECS to E2EE	23
Figure 7. Login on AAA server	24
Figure 8. Login with Google to E2EE	25
Figure 9. Google login and authorization.....	25
Figure 10. SPECS negotiation (service selection)	33
Figure 11. SPECS negotiation (capability selection).....	34
Figure 12. SPECS negotiation (security controls selection) (1/2)	35
Figure 13. SPECS negotiation (security controls selection) (2/2)	36
Figure 14. SPECS negotiation (definition of SLOs).....	37
Figure 15. SPECS negotiation (request offers)	38
Figure 16. SPECS negotiation (received SLA Offers)	38
Figure 17. SPECS negotiation (reviewing SLA Offer)	39
Figure 18. SPECS negotiation (sign SLA)	40
Figure 19. SPECS Negotiation (implement SLA)	40
Figure 20. SPECS Negotiation (observe SLA)	41

Index of tables

Table 1. Impact of the Secure Storage application on execution KPIs.....	10
Table 2. Capabilities offered through the AAAaaS application.....	16
Table 3. Controls and metrics for the DBB mechanism.....	17
Table 4. Controls and metrics for the E2EE mechanism.....	17
Table 5. Controls and metrics for the AAA mechanism.....	18
Table 6. ESXi Performance	27
Table 7. ESXi Virtual Machine Hardware	27
Table 8. vCenter Server Performance	28
Table 9. AAAaaS application user profiles for performance tests.....	28
Table 10. Performance results for the AAAaaS application on EMC's testbed.....	30
Table 11. Integration test <i>App-D1</i> for the AAA-as-a-Service application	42
Table 12. Integration test <i>App-D2</i> for the AAA-as-a-Service application	43

1. Introduction

Task T5.4 was driven by the integration of the two main SPECS applications, namely the (i) *SPECS ngDC application*, which consists of the extension of the EMC ViPR storage controller [1] commercial product with the support for SLA management (described in deliverable D5.3), and the (ii) *SPECS Secure Storage application*, which provides end-to-end encryption over a cloud storage service offered by Koofr [13] (described in D5.2.2).

The SPECS ngDC application allows End-users to negotiate performance and security capabilities of a cloud storage service with resources managed transparently through ViPR. During the project lifetime, it was realised that the SPECS-enhanced ViPR solution could benefit from the integration with the End-to-End Encryption (E2EE) mechanism that was developed in tasks T4.3 and T5.2. Thus, the ViPR solution has been extended not only with SLA management functionalities, but also with an easy-to-use client-side encryption and all its secure storage SLA metrics and monitoring capabilities (please refer to deliverables D4.3.3, D5.2.1, and D5.2.2 for more details).

Moreover, as the SPECS platform provides the AAA mechanism (an identity and access management component which supports OAuth2 protocol), it was decided to use this mechanism to provide authentication and authorization functionalities with single sign-on features on top of the E2EE mechanism.

The resulting application offers an end-to-end encrypted storage service on top of the ViPR solution (enhanced with support for SLA management), with identity management and access control features provided by means of the integration of an OAuth server, managed through the AAA mechanism.

The rest of the document is structured as follows.

The high-level description of the application, named "*SPECS Secure Storage with AAA-as-a-Service application*" or simply "*AAA-as-a-Service application*" to stress the integration of the AAA mechanism compared to the *Secure Storage application*, is provided in Section 3. In particular, Section 3.1 reports the user story to which the application refers, while Section 3.2 lists the related validation scenarios.

The application development process, described according to the steps defined in deliverable D5.1.3, is illustrated in Section 4, which summarizes the information on the service being delivered, the capabilities offered on top of it and related metrics.

Section 5 describes the architecture of the application. It presents the testbed on which the application has been developed and tested, illustrates the main SPECS components involved, and how those components have been integrated with ViPR. Moreover, the section identifies the changes required within the E2EE mechanism in order to enable its integration with the AAA mechanism.

Section 6 presents some performance and scalability results, while Section 7 and Section 8 respectively discuss how to install and use the application. Finally, Section 9 reports on the testing results and Section 10 draws some conclusions.

2. Relationship with other deliverables

In Figure 1 we present other deliverables of the project that served as an input for this document. In particular:

- **WP1:** The architecture and the behaviour of the SPECS platform are summarized in deliverable D1.1.3. Deliverables D1.5.1 and D1.5.2 present integration aspects in SPECS and include integration scenarios for the AAAaaS application.
- **WP2:** Deliverables D2.3.2 and D2.3.3 present the final prototypes of the components of the Negotiation module.
- **WP3:** The prototypes of the components of the Monitoring module are discussed in deliverables D3.4.1 and D3.4.2.
- **WP4:** Deliverables D4.3.2 and D4.3.3 present prototypes of the core Enforcement components as well as the security mechanisms integrated with the AAAaaS application.
- **WP5:** The application is developed on top of the default SPECS application introduced in deliverable D5.1.3 and implements the associated user story and validation scenarios defined in deliverable D5.1.2. As already mentioned in the introduction, the AAAaaS application extends the ViPR solution discussed in deliverable D5.3 and secure storage security mechanisms presented in deliverables D5.2.1 and D5.2.2.

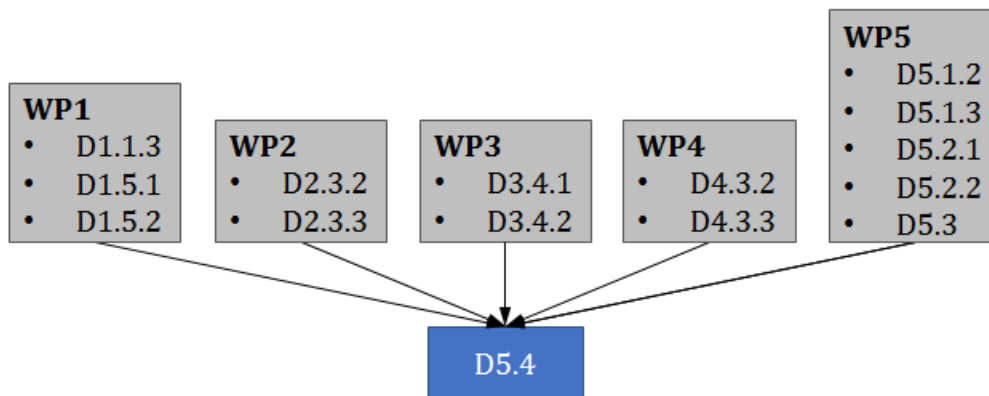


Figure 1. Relationship with other deliverables

3. SPECS AAA-as-a-Service application description

The AAA-as-a-Service (AAAaaS) application is one of the validation applications introduced in deliverable D5.1.2, whose goal is to verify the correct behaviour of the platform with respect to the specified validation scenarios. Validation applications are built on top of the defined user stories and comprise the SPECS solutions portfolio (cf. D6.2.2), released in July 2015 and offering a set of four different close-to-market software products provided by the SPECS consortium partners.

The AAAaaS application is an extension of the Secure Storage user story and is mapped to the *SPECS-enhanced ViPR* solution of the SPECS solution portfolio. It offers the end-to-end encryption, data integrity, backup, identity management and access control functionalities on top of a cloud storage service represented by the ViPR solution [1] provided by EMC.

In the following subsections, we present the refined user story for the application, along with the related validation scenarios.

3.1. User story: Secure Storage with Identity Management

The *Secure Storage with Identity Management* (STOIM) user story, to which the application refers, refines the Secure Storage user story, involving a non-expert End-user that wants to store data on a remote cloud provider with data confidentiality guarantees, such that the provider itself is prevented from accessing the stored data.

In the refined user story, a Cloud Service Provider (CSP) offering cloud storage services resorts to the SPECS platform to provide secure storage functionalities to its customers, along with an Identity Management system for the management of End-users of the service. The customers of this “enhanced” secure storage service are themselves providers, who sell it to their own customers representing the End-users of this User Story. The CSP who builds the enhanced secure storage service hosts the SPECS platform (Interaction Model 2, see deliverable D1.1.1) and is therefore the SPECS Owner.

The enhanced secure storage service offered by the SPECS Owner ensures the following properties:

- Confidentiality and integrity of the stored data;
- Consistency among data updates (i.e., write-serializability);
- Requested data freshness (i.e., read-freshness);
- Management of access control policies for the users of the storage service;

In the next section, we show a set of relevant scenarios related to this User Story and focused on the identity management and access control functionalities.

3.2. Validation scenarios

In the context of the user story described in the previous section, the following scenarios can be derived:

- *AAA.1 – Identity Management Set-up.* A customer acquires the enhanced secure storage service from the SPECS Owner, configures the service and sets the access control policies for its End-users by using the identity management features offered by the service. Moreover, the customer configures the Identity Federation by identifying the supported identity providers;

- *AAA.2 – User registration.* An End-user registers with the secure storage service. An account is created for the End-user and managed by the identity management system;
- *AAA.3 – User access with internal account.* An End-user requests the access to the storage system by using the account created when registering with the service;
- *AAA.4 – User access with federated account.* An End-user requests the access to the storage system by using an external account belonging to a supported Identity Provider.

These scenarios are detailed in the following subsections, according to the template introduced in deliverable D5.1.1.

In the table below we present the impact of the AAAaaS application on the execution KPIs (defined in deliverable D5.1.1). For each key concern defined in deliverable D5.1.1 we report the percentage of its coverage by the AAAaaS validation scenarios.

Key concern ID	Key concern	AAA-as-a-Service application	All SPECS applications
EC _U	User	20%	66.7%
EC _{TS}	Target services	0% ¹	100%
EC _{IC}	Invocation chain	33.3%	100%
EC _{SLA}	SLA life cycle	0% ²	78.9%
EC _{SS}	SPECS services	27% ³	100%

Table 1. Impact of the Secure Storage application on execution KPIs

3.2.1. Identity management set-up

General Information		
ID	AAA.1 – Identity_Management_Set-up	
Version	1.0	
User Story	STOIM	Secure Storage with Identity Management
Invocation Chain	IM2-CSP	Interaction Model 2- SPECS acting the role of CSP
Scenario Steps		
General Description	In this scenario, a customer acquires the enhanced secure storage service from the SPECS Owner, configures the service and sets the access control policies for his End-users by using the identity management features offered by the service. Moreover, the provider configures the Identity Federation by identifying the supported identity providers	
Steps		
1	Phase	Service acquisition
	Actor	Customer, SPECS Owner
	Preconditions	
	Trigger	
	Actions	The customer acquires the enhanced secure storage service from the SPECS Owner and is provided with access to an application for its configuration.
	Postconditions	

¹ The Target Service key concern is not applicable to the defined validation scenarios.

² The SLA key concern is not applicable to the defined validation scenarios

³ There are 79 requirements associated to the validation scenarios of the AAA user story out of 293 all requirements for the SPECS framework.

2	Phase	<i>Identity Management Set-up</i>
	Actor	<i>Customer, AAA mechanisms component</i>
	Preconditions	
	Trigger	
	Actions	<i>The customer accesses the application and configures, via the AAA mechanisms offered, the storage service that will offer to End-users by identifying the access control policy: he sets different authorization roles for the users and configures the tools for authentication (e.g., LDAP, OAUTH) and authorization (e.g., XACML).</i>
	Postconditions	
3	Phase	<i>Identity Federation configuration</i>
	Actor	<i>Customer, AAA mechanisms component</i>
	Preconditions	
	Trigger	
	Actions	<i>The customer, via the AAA mechanisms offered with the service, identifies a set of external Identity Providers that he aims at supporting in an Identity Federation (e.g., Facebook)</i>
	Postconditions	
Graphical Model		<i>Not reported</i>
<u>Coverage Information</u>		
Users	<i>U_1 (CSC:User)</i>	
Target services	<i>Not Applicable</i>	
SPECS services	<i>AAA mechanisms</i>	
SLA	<i>Not Applicable</i>	

3.2.2. User registration

<u>General Information</u>		
ID	<i>AAA.2 – User_Registration</i>	
Version	<i>1.0</i>	
User Story	<i>STOIM</i>	<i>Secure Storage with Identity Management</i>
Invocation Chain	<i>IM2-CSP</i>	<i>Interaction Model 2- SPECS acting the role of CSP</i>
<u>Scenario Steps</u>		
General Description	<i>In this scenario, an End-user of the enhanced secure storage service performs a registration by providing his/her data.</i>	
Steps		
1	Phase	<i>Registration</i>
	Actor	<i>End-user, AAA mechanisms component</i>
	Preconditions	<i>The provider of the service (i.e., the customer that has acquired the service from the SPECS Owner in the User Story) has defined an access control policy</i>
	Trigger	
	Actions	<i>The End-user fills the registration form with her/his personal information and specifies the features of the storage service he/she is interested to use. The information is submitted to the AAA component.</i>
	Postconditions	

2	Phase	<i>Registration</i>
	Actor	<i>AAA mechanisms component, End-user</i>
	Preconditions	
	Trigger	
	Actions	<i>A new account is created for the End-user and submitted information is associated with it. The End-user is returned the credentials to access the application.</i>
	Postconditions	
Graphical Model		<i>Not reported.</i>
<u>Coverage Information</u>		
Users	<i>U_1 (CSC:User)</i>	
Target services	<i>Not Applicable</i>	
SPECS services	<i>AAA mechanisms</i>	
SLA	<i>Not Applicable</i>	

3.2.3. User access internal account

<u>General Information</u>		
ID	<i>AAA.3 - User_Access_Internal_Account</i>	
Version	<i>1.0</i>	
User Story	<i>STOIM</i>	<i>Secure Storage with Identity Management</i>
Invocation Chain	<i>IM2-CSP</i>	<i>Interaction Model 2- SPECS acting the role of CSP</i>
<u>Scenario Steps</u>		
General Description	<i>In this scenario, an End-user requests the access to the storage system by using the account created when registering with the service;</i>	
Steps		
1	Phase	<i>Authentication</i>
	Actor	<i>End-user, AAA mechanisms component</i>
	Preconditions	
	Trigger	
	Actions	<i>The End-user submits an authentication request (through, for example, an SAML request) to the SPECS AAA component by using the account previously created during registration.</i>
	Postconditions	
2	Phase	<i>Authentication</i>
	Actor	<i>End-user, AAA mechanisms component</i>
	Preconditions	<i>The End-user has a valid account on the application</i>
	Trigger	
	Actions	<i>The AAA component checks the account in the internal repository (e.g. LDAP server) and authenticates the End-user, by applying the access control policy related to his/her role.</i>
	Postconditions	
Graphical Model		<i>Not reported</i>
<u>Coverage Information</u>		
Users	<i>U_1 (CSC:User)</i>	

Target services	<i>Not Applicable</i>
SPECS services	
SLA	<i>Not Applicable</i>

3.2.4. User access external account

General Information		
ID	<i>AAA.4 - User_Access_External_Account</i>	
Version	<i>1.0</i>	
User Story	<i>STOIM Secure Storage with Identity Management</i>	
Invocation Chain	<i>IM2-CSP Interaction Model 2- SPECS acting the role of CSP</i>	
Scenario Steps		
General Description	<i>In this scenario, an End-user requests the access to the storage system by using an external account belonging to a supported Identity Provider. When the user chooses to authenticate through an external source, the application checks that the external account is associated with a supported identity provider. In this case, the user is authenticated. Otherwise the application asks if the End-user wants to associate the external account with her/his existing internal account. In this latter case, the End-user must first be authenticated on the application in order to prove the ownership of the internal account.</i>	
Steps		
1	Phase	<i>Authentication</i>
	Actor	<i>End-user</i>
	Preconditions	<i>The End-user has a valid account on the selected external authentication source.</i>
	Trigger	
	Actions	<i>The End-user requests the access to the storage service by selecting an external authentication source and performs the login with the credentials of the external account, retrieving her/his personal information.</i>
	Postconditions	<i>The End-user is authenticated on the external authentication source.</i>
2.1	Phase	<i>Authentication</i>
	Actor	<i>AAA component, End-user</i>
	Preconditions	<i>An internal account exists for the End-user. The internal account is already linked to the external account.</i>
	Trigger	
	Actions	<i>The AAA component checks if the external account is associated with any valid internal account and authenticates the End-user.</i>
	Postconditions	<i>The End-user is authenticated on the application.</i>
2.2	Phase	<i>Authentication</i>
	Actor	<i>AAA component, End-user</i>
	Preconditions	<i>An internal account exists for the End-user. The internal account is not yet linked to the external account.</i>
	Trigger	
	Actions	<i>The AAA component checks if the external account is associated with any valid internal account and does not find any match. The AAA component asks the End-user to associate the external account to her/his existing internal account, if any exists.</i>
	Postconditions	<i>The internal account of the End-user is linked to this/he external account.</i>

3.2	Phase	<i>Authentication</i>
	Actor	<i>End-user, AAA component</i>
	Preconditions	
	Trigger	
	Actions	<i>The End-user logs into the application with the credentials of the internal account. The AAA component authenticates the End-user.</i>
	Postconditions	<i>The AAA component End-user is authenticated.</i>
4.2	Phase	<i>Account association</i>
	Actor	<i>SPECS AAA component</i>
	Preconditions	
	Trigger	
	Actions	<i>The link with the external account is created for the user entry by the AAA component.</i>
	Postconditions	<i>The link to the external account is stored in the AAA component repository</i>
Graphical Model		<i>Not reported</i>
<u>Coverage Information</u>		
Users	<i>U_1 (CSC:User)</i>	
Target services	<i>Not Applicable</i>	
SPECS services		
SLA	<i>Not Applicable</i>	

4. Application development process

In this section, the process of the development of the AAAaaS application is illustrated. As reported in deliverable D5.1.3, SPECS applications are built by developers by customizing the *default SPECS application*, which includes all the functionalities needed to negotiate, implement, and monitor an SLA. The customization consists in identifying, developing, and deploying the mechanisms needed to grant the features that the SPECS Owner is willing to offer through the application. The application development process, shown in Figure 2, consists of the following four steps:

1. **Cloud Service Definition:** We define the type of the cloud service to be offered to the End-users through the application.
2. **Security Mechanism Preparation:** We identify and develop/integrate security mechanisms able to enforce and monitor the defined cloud service.
3. **SLA Template Preparation:** We summarize and formalize all features offered through the application in an SLA Template for an automated SLA negotiation and enforcement.
4. **SLA Template and Security Mechanisms Deployment:** We register the application's SLA Template and all Chef recipes⁴ that are needed for an automated management of the associated security mechanisms.



Figure 2. SPECS Application development process

In the following subsections, we detail these steps for the SPECS AAAaaS application.

4.1. Cloud service definition

The target service offered by the AAAaaS application is a cloud storage service hosted on the EMC datacentre and managed through the SPECS-enhanced ViPR controller, provided with end-to-end encryption features and identity and access control management functionalities. Secure storage services are provided by SPECS through the **DBB (Database and Backup as a Service)** and the **E2EE (End-2-End Encryption) security mechanisms**. These mechanisms, introduced in deliverable D4.3.2, offer the following security properties:

- Client-side encryption enforcing confidentiality (CO).
- Detection and proof of violations related to integrity (IN), write-serializability (WS), and read-freshness (RF).
- Backup of stored data.

Identity management and access control functionalities are provided by means of the **AAA (Authentication, Authorization, and Auditing) security mechanism**, described in deliverable D4.3.3.

4.2. Security mechanism preparation

As discussed in the previous section, the AAAaaS application integrates three security mechanisms developed in SPECS, namely the DBB, E2EE, and AAA mechanisms, with the EMC ViPR-based storage solution. These mechanisms enforce capabilities as shown in Table 2.

⁴ In SPECS, the automated deployment and configuration of resources and services is orchestrated with Chef [2].
SPECS Project – Deliverable 5.4

Security Mechanism	Capability
DBB	Surviving incidents that compromise the availability and/or integrity of data stored remotely by providing backup service and the detection of violations associated to write-serializability and read-freshness.
E2EE	Providing client-side encryption enforcing confidentiality.
AAA	Providing identity management and access control functionalities.

Table 2. Capabilities offered through the AAAaaS application

Note that all technical details for the DBB and the E2EE mechanisms are provided in deliverables D4.3.2 (initial prototypes), D4.3.3 (final prototypes – automated implementation and remediation), D5.2.1, and D5.2.2 (final prototypes – integration with the Secure Storage application).

Similarly, the prototype for the AAA mechanisms, which enables management of user accounts and provides a federated authentication features, is presented in deliverable D4.3.3.

Each mechanism comes with its own cookbook, which enables its automatic deployment and configuration through Chef [2]. The recipes/cookbooks are available on the project’s Bitbucket account [3].

Since the functionalities offered by the AAA mechanisms affect also the E2EE mechanism (because it requires that all interactions with End-users are authenticated), in order to activate both of them it is necessary to update the E2EE implementation for integration with AAA. The needed changes in E2EE are reported in Section 5.4.

4.3. SLA Template preparation

The SPECS application negotiates the services with End-users by following a process based on the WS-Agreement standard, which adopts proper templates summarizing the features that can be offered to customers (see deliverable D2.3.3).

To enable negotiation, the SPECS developer has to build a WS-Agreement-compliant SLA template, which summarizes what kind of service is to be delivered, what are the security capabilities that can be offered to End-users through the application, and what are the related guarantees. To enforce features discussed in Section 4.1 (that are enforced and monitored by security mechanisms reported in Section 4.2), we defined three security capabilities shown in Table 2. For each of these three capabilities we identify a set of NIST [4] and CSA [5] security controls and a set of security metrics that they implement. We report them in the following three tables (separately for each capability/mechanism).

Security capability	Database and Backup as a Service
NIST 800-53r4	<ul style="list-style-type: none"> • CP-2(4) Contingency plan Resume all missions / Business functions • CP-2(6) Contingency plan Alternate processing / Storage site • CP-6(1) Alternate storage site Separation from primary site • CP-9 Information system backup • CP-9(6) Information system backup Redundant secondary system • CP-10 Information system recovery and reconstitution • SI-7 Software, firmware, and information integrity

	<ul style="list-style-type: none"> • SI-7(1) Software, firmware, and information integrity Automated notifications of integrity violations • SI-7(5) Software, firmware, and information integrity Automated response to integrity violations
CSA CCM v3.0	<ul style="list-style-type: none"> • IVS-02 Infrastructure & Virtualization security Change detection • BCR-01 Business continuity management & Operational resilience Business continuity planning • BCR-11 Business continuity management & Operational resilience Policy • AIS-03 Application & Interface security Data integrity
Security metrics	<ul style="list-style-type: none"> • Write-Serializability (WS): This metric ensures the End-user consistency among updates of the stored data. In case of WS violations, the End-user will be notified and the system will be restored to the state of the last completed backup. • Read-Freshness (RF): This metric ensures the End-user that the requested data will always be fresh as of the last update. In case of RF violations, the End-user will be notified and the system will be restored to the state of the last completed backup. • Integrity (IN): This metric ensures the End-user integrity of the stored data.

Table 3. Controls and metrics for the DBB mechanism

Security capability	End-2-End Encryption
NIST 800-53r4	<ul style="list-style-type: none"> • SC-12 Cryptographic key establishment and management • SC-13 Cryptographic protection
CSA CCM v3.0	<ul style="list-style-type: none"> • EKM-01 Encryption & Key management Entitlement • EKM-03 Encryption & Key management Sensitive data protection
Security metrics	<ul style="list-style-type: none"> • Confidentiality (CO): This metric ensures the End-user confidentiality of the stored data. Confidentiality is enforced with end-2-end encryption provided by the Client component. We guarantee that the Client component is audited and thus (used as is) grants the security of encryption.

Table 4. Controls and metrics for the E2EE mechanism

Security capability	Authentication, Authorization, and Auditing
NIST 800-53r4	<ul style="list-style-type: none"> • AC-1 Access control policy and procedures • AC-2(6) Account management Dynamic privilege management • AC-2(7) Account management Role-based schemes • AC-2(8) Account management Dynamic account creation • AC-2(12) Account management Account monitoring / Atypical usage • AC-3 Access enforcement • AC-3(7) Access enforcement Role-based access control • AC-7 Unsuccessful logon attempts • AU-1 Audit and accountability policy and procedures • AU-2 Audit events • AU-2(3) Audit events Reviews and updates • AU-3 Content of audit records

CSA CCM v3.0	<ul style="list-style-type: none"> • IAM-02 Credential lifecycle / Provision management • IAM-04 Policies and procedures
Security metrics	<ul style="list-style-type: none"> • Secure Delegated Access (SDA): This metric ensures that an OAuth Server is configured to ensure authentication and authorization of users and secure delegated access to the users' resources to registered clients. • Access Report Generation Frequency (ARGF): This metric sets the frequency of access reports generation. For example, for <i>access_report_gen_frequency=12</i>, SPECS ensures that a report is generated at least once every 12 hours. • AAA Log Completeness (ALC): This metric represents how detailed the access reports must be.

Table 5. Controls and metrics for the AAA mechanism

The created SLA Template for the AAAaaS application is reported in full in Appendix 1.

4.4. SLA Template and security mechanisms deployment

The last application development step is the deployment of the security mechanisms and of the SLA Template to make them available to the SPECS application. Since this step is not application specific (the process of the registration of recipes and the SLA Template is the same for all SPECS applications), we refer the interested reader to deliverable D5.1.3 for further details about this step.

5. Architecture

This section presents the design of the AAAaaS application. In particular, we present the testbed on which the application has been developed and tested (in Section 5.1), the EMC software used for the AAAaaS application (in Section 5.2), the integrated artefacts of the SPECS framework (in Section 5.3), and the modifications of the SPECS developed security mechanisms integrated into the application (in Section 5.4).

Note that further details about ViPR are available in deliverable D5.3, details about the SPECS framework are discussed in deliverable D1.1.3, and details about the mechanisms integrated into the AAAaaS application are presented in deliverables D4.3.2, D4.3.3, D5.2.1, and D5.2.2.

5.1. Testbed

In order to emulate a real world scenario for D5.4, the testbed (described in deliverable D5.3, Section 5.3) was extended by adding another server running ESXi to the existing architecture. This new physical architecture, shown in Figure 3, enables the minimum features required for a datacentre managed by VMWare vCenter.

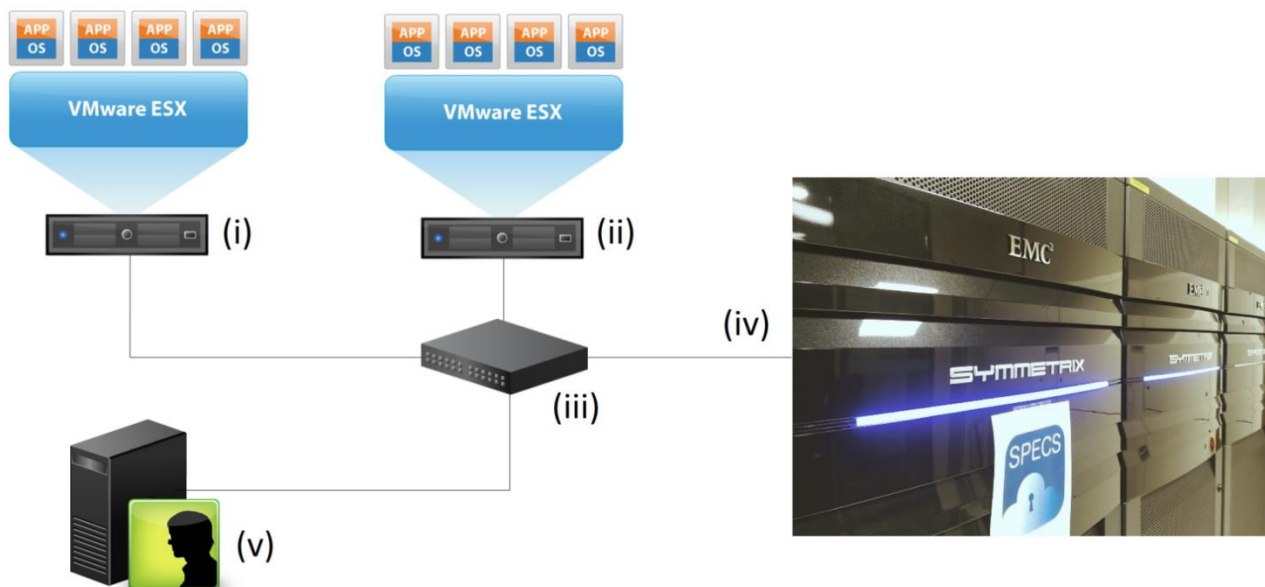


Figure 3. Physical architecture of the testbed

The core hardware components of the architecture, illustrated in Figure 3, are:

- (i) ESXi server [8] (running vCenter)
- (ii) ESXi server
- (iii) Cisco Switch
- (iv) VMAX array [9]
- (v) Management server

The new server (ii) added to the architecture is a Cisco UCS C200 M2⁵ equipped with 4 Intel Xeon E5620 2.4 GHz CPUs, ~50 GB of RAM and ~400GB of storage, the machine is installed with ESXi 5.5 and is managed by the vCenter running on the original server (i). As with the

⁵ http://www.cisco.com/c/dam/en/us/products/collateral/servers-unified-computing/ucs-c-series-rack-servers/c200m2_1ff_specsheet.pdf

components in the previous architecture in D5.3, (i) is connected to the Cisco switch (iii) in the same IP sub network.

The VMAX array (iv) provides external storage to the hosts (i) and (ii) in the architecture, emulating a Cloud storage scenario. Storage is acquired and managed through the EMC commercial storage automation software, ViPR. ViPR abstracts the physical layer and pools resources into the virtual blocks of storage and delivers automated, policy driven storage services on demand via a self-service catalogue. In D5.3, the ViPR application (and its open source counterpart, CoprHD [10]) was integrated with the SPECS framework, enabling End-users of the SPECS web application to acquire virtual storage resources according to a set of performance and security requirements guaranteed by an SLA.

A VMware *Distributed Resource Scheduler* (DRS)⁶ cluster was created and the two hosts added to it. The DRS cluster is a collection of ESX/ESXi hosts and associated VMs with shared resources and a shared management interface. The host's resources become part of the cluster's resources. In addition to this aggregation of resources, a DRS cluster supports cluster-wide resource pools and enforces cluster-level resource allocation policies.

The following cluster-level resource management capabilities are available.

- **Load Balancing:** The vCenter Server system monitors distribution and usage of CPU and memory resources for all hosts and VMs in the cluster. DRS compares these metrics to an ideal resource utilization given the attributes of the cluster's resource pools and VMs, the current demand, and the imbalance target. DRS then performs (or recommends) VM migrations. When you first power on a VM in the cluster, DRS attempts to maintain proper load balancing either by placing the VM on an appropriate host or by making a recommendation.
- **Power Management:** When the VMware Distributed Power Management (DTM) feature is enabled, DRS compares cluster- and host-level capacity to the demands of the cluster's VMs, including recent historical demand. DTM places (or recommends placing) hosts in standby power mode if sufficient excess capacity is found. DTM powers on (or recommends powering on) hosts if capacity is needed. Depending on the resulting host power state recommendations, VMs might need to be migrated to and from the hosts.
- **Virtual Machine Placement:** You can control the placement of VMs on hosts within a cluster, by assigning DRS affinity or anti affinity rules.

The other minimal feature is the VMWare High Availability (HA)⁷ capability which supports high availability for VMs by pooling them and the hosts they reside on into a cluster. VMware HA monitors the hosts. In the event of host failure, VMware HA migrates virtual machines to hosts with available capacity. When you add new VMs to a VMware HA enabled cluster, the VMware HA capability checks whether there is enough capacity available to power on that VM on a different host in case of host failure.

5.2. ViPR

On completion of the SLA negotiation and enforcement of storage services through ViPR according to the steps outlined in D5.3, a virtual block of storage provided by physical drives

⁶ https://www.vmware.com/pdf/vmware_drs_wp.pdf

⁷ https://www.vmware.com/files/pdf/VMwareHA_twp.pdf

in the VMAX array is made available to the End-user. This storage service is monitored according to the SLOs outlined in the associated SLA that specify all the parameters in order to guarantee the users' security requirements. This storage, allocated through ViPR, will be presented to the vCenter running on (i) as a data store on the designated ESXi server. This data store can then be used store ISO images and templates, and to allocate VMs. The data store is added to the DRS cluster resource pool so that it appears to ESXi and vCenter in the same way as any storage directly connected to the host.

In this way the End-users have an end-to-end process for acquiring Cloud storage through ViPR (or CoprHD) and then deploying VMs and services on top of this storage. The negotiation, enforcement and monitoring mechanisms offered through SPECS provide assurances and fine grained control over the underlying resources on the CSP side upon which the End-user will run their environment.

5.3. Integration with SPECS framework

In Figure 4 we present a high level architecture of the AAAaaS application in terms of integration with the SPECS framework. Through the AAAaaS web application, End-users will be able to acquire cloud storage services and the system will automatically deploy applications and external packages to implement security services providing end-to-end encryption functionalities. Furthermore, via the AAA security mechanism, the access to cloud storage resources will be protected via an authentication and authorization system based on the OAuth protocol.

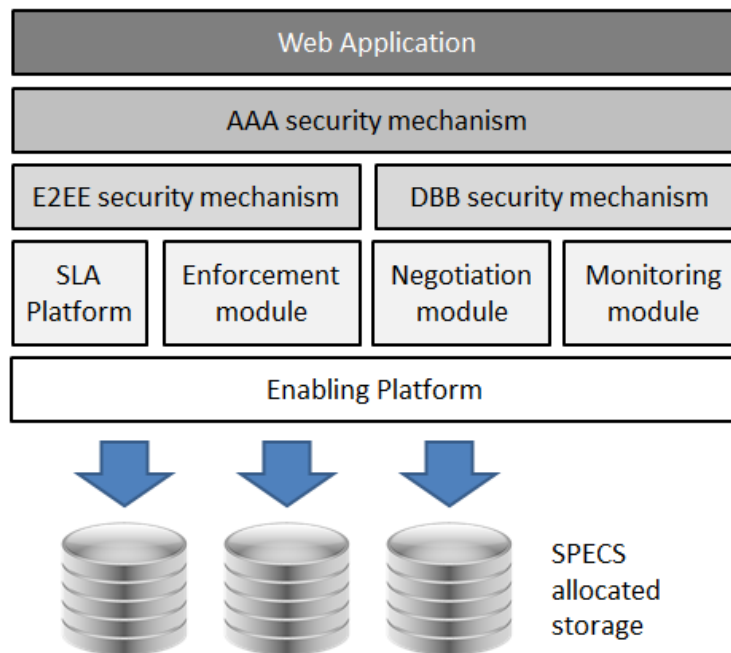


Figure 4. Integration of AAA security mechanism with SPECS framework

Apart from the AAA, DBB, and E2EE security mechanisms, the AAAaaS application integrates the core modules of the SPECS framework to manage the SLA life cycle. Namely, the AAAaaS application integrates the SLA Platform, the Negotiation module, the Monitoring module, the Enforcement module, and the Enabling Platform. For further details, see dedicated deliverables (D1.4.1, D2.3.3, D.3.4.2, D4.3.3, D1.6.2, respectively).

The integration of EMC's infrastructure, namely VMWare infrastructure, with the SPECS framework was possible by using the vSphere API. This was accomplished by developing a Java connector (VMWare connector) that allows SPECS to interact directly with the VMWare infrastructure through the vSphere API. The VMWare connector was developed as a module in the SPECS Broker, by using the VMware vSphere Management SDK.

The VMWare connector enables the following specific operations into the SPECS Broker: *Create VM*, *Clone VM from Template* and *Delete VM*. Within these operations, it is also possible to specify certain configuration options and target datastores, resource pools, networks, etc. The VMWare connector is used by the SPECS Broker to deploy configured VMs from existing templates on a target host.

The above described data storage selection feature allows the SPECS Broker to target the ViPR allocated storage and to deploy the VMs on top of it, in order to implement the security features included in the corresponding SLA.

A SPECS custom OS⁸ raw image was converted to a VMWare template with two network interfaces. The deployment of the VM is really fast, taking less than 10 seconds, due to the size of the template.

After the VMs creation, the basic software installation/configuration, and the Chef client installation (carried out over SSH), the new VMs will be registered with the Chef server and SPECS will complete their configuration by installing the required software. In the case of the AAAaaS application, this includes the installation of the AAA, DBB, and E2EE mechanisms, which are deployed using proper Chef recipes.

The SPECS Monitoring module will continuously monitor and verify whether the SLAs signed by the End-users are always enforced correctly. Any detected violations will be notified to the SPECS Enforcement module.

5.4. E2EE integration with AAA

The first version of the E2EE mechanism supported only Secure Remote Password [6] authentication. This means that the user needed to register to the E2EE mechanism which then stored and managed the user account. Later the support for token authentication has been implemented to meet the Koofr authentication requirements (please see deliverable D5.2.2 for more details). However, this still required that user accounts are managed by the E2EE mechanism itself. As SPECS platform provides the AAA mechanism (an identity management component which supports OAuth2 protocol), it was decided that this component will be used to provide a single sign-on using OAuth2 in the E2EE mechanism. This way the Koofr requirement for the E2EE mechanism to support the OAuth2 authentication would be met.

Using the AAA component, the sign-on window in the E2EE Client was changed from the one depicted in Figure 5 to the one depicted in Figure 6.

⁸ https://bitbucket.org/specs-team/specs-core-enabling_platform-custom-os
SPECS Project – Deliverable 5.4

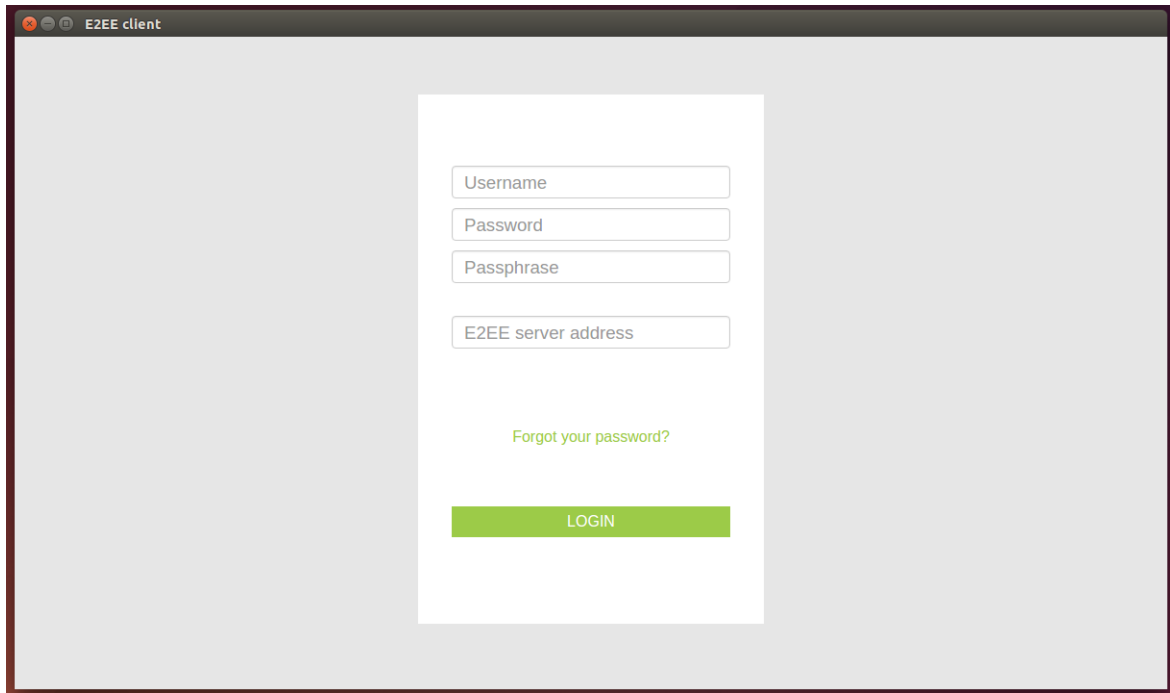


Figure 5. E2EE Client sign-on

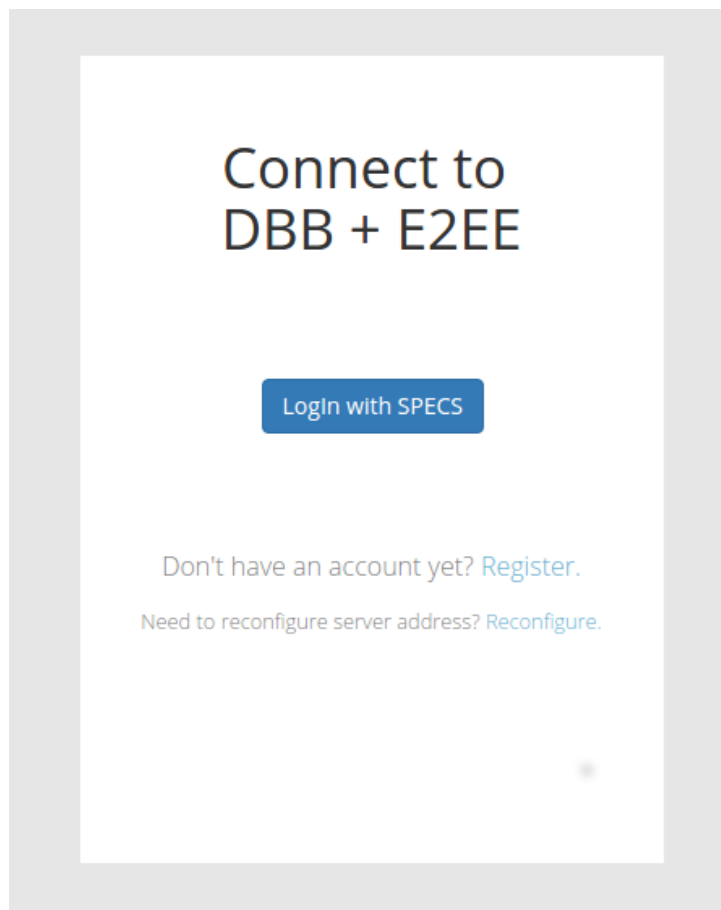
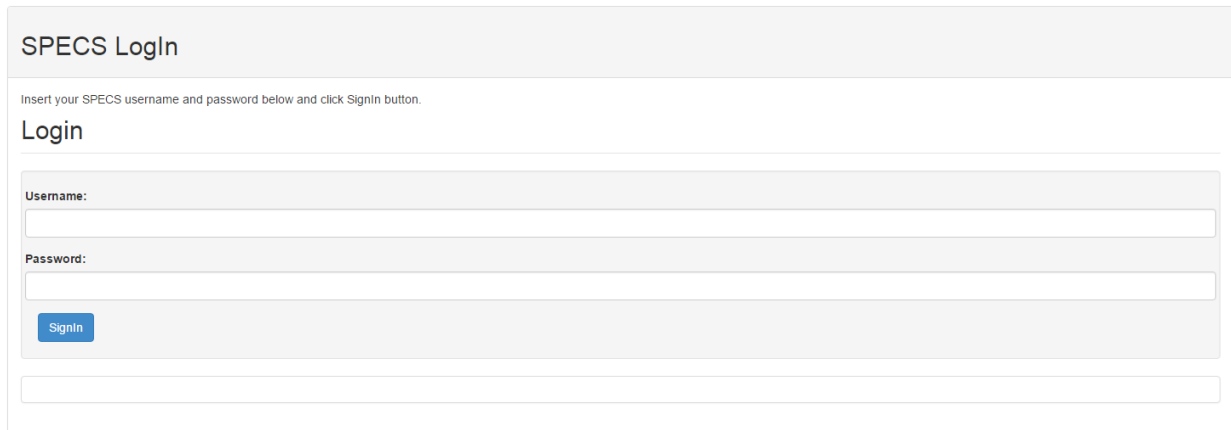


Figure 6. Login with SPECS to E2EE

Note that the passphrase and E2EE server address are still needed, but were moved to the second screen of the application. The username and password are not needed because the user is directed using OAuth2 protocol to the SPECS Login page depicted in Figure 7.



© SPECS Project 2015

Figure 7. Login on AAA server

Once the user clicks the button on the E2EE Client sign-on page and submits the credentials on the SPECS Login page, the AAA Server component returns the authorization code (if the credentials are valid) to the E2EE Client which can then execute a request to the AAA Server to retrieve a token. The code for retrieving the token is given below:

```
var code = "{{ code }}";
var auth = btoa(CLIENT_ID + ":" + CLIENT_SECRET);
var request = new XMLHttpRequest();
request.onreadystatechange = function() {
    if (request.readyState == 4) {
        console.log(request.response);
    }
};
var url = TOKEN_ENDPOINT + '?grant_type=authorization_code&redirect_uri=' +
REDIRECT_URI + '&code=' + code; request.open("POST", url, true);
request.setRequestHeader('Authorization', 'Basic ' + auth);
request.setRequestHeader('Content-Type', 'application/x-www-form-urlencoded');
request.send(null);
```

Once the AAA Server returns the token to the E2EE Client, each request to the E2EE Server needs to be equipped with a token. The E2EE Server checks the validity of the token using the following code:

```
response, err := http.Get(TOKEN_ENDPOINT + "?token=" + token)
if err != nil {
    fmt.Printf("Error while making request")
    return false
}
defer response.Body.Close()
body, _ := ioutil.ReadAll(response.Body)
fmt.Printf(string(body) + "\n") return !strings.Contains(string(body), "error")
```

If the token is valid, the E2EE Server accepts the request and returns a corresponding response (for E2EE Server API please see deliverable D5.2.2).

Once SPECS login has been enabled, it was simple to support other external identity providers. Because Koofr support Google and Twitter sign-on, Google sign-on was added also to the E2EE mechanism as depicted in Figure 8.

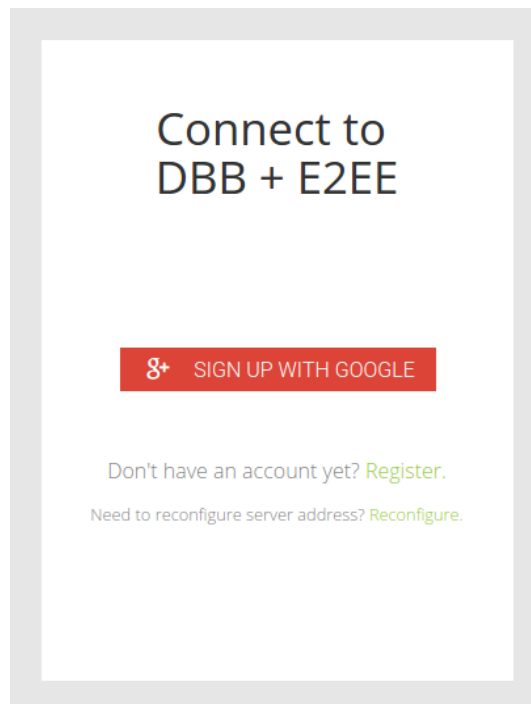


Figure 8. Login with Google to E2EE

Submitting the form takes user to the Google dialog page as depicted in Figure 9.

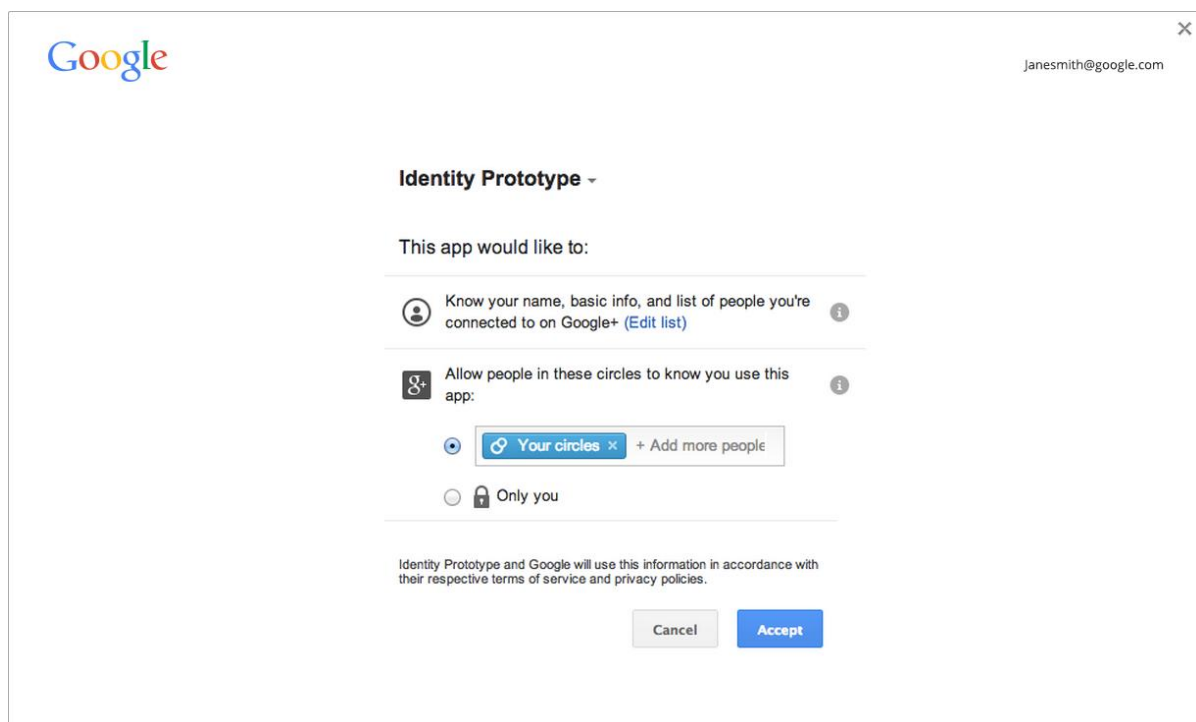


Figure 9. Google login and authorization

Secure Provisioning of Cloud Services based on SLA Management

The code remains largely unmodified, only while for SPECS login a function `launchWebAuthFlow` [12] is needed to be used, function `getAuthToken` is used for a Google login.

```
chrome.identity.launchWebAuthFlow( {'url':specsURL,'interactive':true},
function(redirectURI){
    if (chrome.runtime.lastError) {
        callback(new Error(chrome.runtime.lastError));
        return;
    }
    var matches = redirectUri.match(redirectRe);
    if (matches && matches.length > 1)
        handleProviderResponse(parseRedirectFragment(matches[1]));
    else
        callback(new Error('Invalid redirect URI'));
});
```

```
chrome.identity.getAuthToken({ 'interactive': true }, function(token)
{
    crypton.token = token
    chrome.identity.getProfileUserInfo(function(userInfo) {
        var username = userInfo["email"]
        if (serverUrl !== "") {
            chrome.storage.local.set({"serverUrl": serverUrl})
        }
        crypton.openSession(username, passphrase)
    })
})
```

6. Performance and scalability analysis

As with deliverable D5.3, the inherent scalable, resource pooling nature of the products used in the SPECS architecture, described in Section 5, means that the SPECS application can scale to meet even the most strenuous demands. With the latest version of vSphere (version 6.0), the maximum configuration limits have been increased to the point where it can be said that the scalability limits of the computing power available to end users is negligible.

Table 6 shows the performance comparison between the most recent versions of vSphere, including vSphere 5.5 used in our testbed (VMWare, 2015). As can be seen, even in vSphere 5.5, the resources available to a single instance of vSphere are capable of accommodating significant infrastructure demands. Implementing the same infrastructure on vSphere 6.0 represents a huge performance boost: double the number of VMs, four times as much RAM and 50% more logical CPUs.

	vSphere 5.0	vSphere 5.1	vSphere 5.5	vSphere 6.0
Logical CPU	160	160	320	480
Physical RAM	2TB	2TB	4TB	12TB ⁹
Virtual CPU	2048	2048	4096	4096
Virtual Machines	512	512	512	1024

Table 6. ESXi Performance

Similarly, the maximum hardware limits for individual VMs are highlighted in Table 7.

	vSphere 5.0	vSphere 5.1	vSphere 5.5	vSphere 6.0
Virtual CPU	32	64	64	128
Virtual RAM	1TB	1TB	1TB	4TB
Max VMDK size	2TB	2TB	62TB	62TB
Virtual SCSI target	60	60	60	60
Virtual NICs	10	10	10	10

Table 7. ESXi Virtual Machine Hardware

As mention with respect to Table 6 above, the performance of a single ESXi host allows for the configuration of a significant number of resources that would accommodate the demands of a typical SME for example. However, even greater scalability and performance can be achieved through the management of multiple hosts and clusters. As illustrated in Table 8, vCenter has the capability to manage up to 1000 hosts (even with vSphere 5.5) and up to 10000 powered on VMs. Further improvements in scale can be gained by linking instances of vCenter Server together to pool the resources of up to 10 vCenter Servers.

Chef server is another key software component in the SPECS infrastructure, orchestrating the deployment of mechanism such as AAA, DBB, and E2EE using Chef recipes. Chef server is highly scalable in this regard; a single instance of the server is capable of handling requests for thousands of nodes [14]. To achieve greater scalability, it is possible to expand into a *tiered front-end/back-end architecture with horizontally scaled front-ends* to reduce the load on bottlenecks.

⁹ 12TB supported on specific OEM certified platforms, otherwise the maximum is 6TB.
SPECS Project – Deliverable 5.4

	vSphere 5.0	vSphere 5.1	vSphere 5.5	vSphere 6.0
Hosts per vCenter	1000	1000	1000	1000
Hosts per datacenter	500	500	500	500
Hosts per cluster	32	32	32	64
VMs per cluster	3000	4000	4000	8000
Powered on VMs	10000	10000	10000	10000
Registered VMs	15000	15000	15000	15000
Linked vCenter Servers	10	10	10	10

Table 8. vCenter Server Performance

In the remainder of this section, we present performance tests defined and executed for the AAAaaS application, and we discuss results. Note that performance and scalability analysis of DBB and E2EE mechanisms, integrated into the application, is available in deliverable D5.2.2.

The methodology for the performance and scalability analysis is detailed in deliverable D1.5.2 and is adopted on the project level. Here we present user profiles defined for the application and present results obtained with the performance analysis tool Gatling [15].

We made a single user profile, associated to the Wizard, which orchestrates all requests to the platform, so summarizing the overall behaviour during the real application execution. Performance figures reported in the tables present all calls invoked by the application wizard.

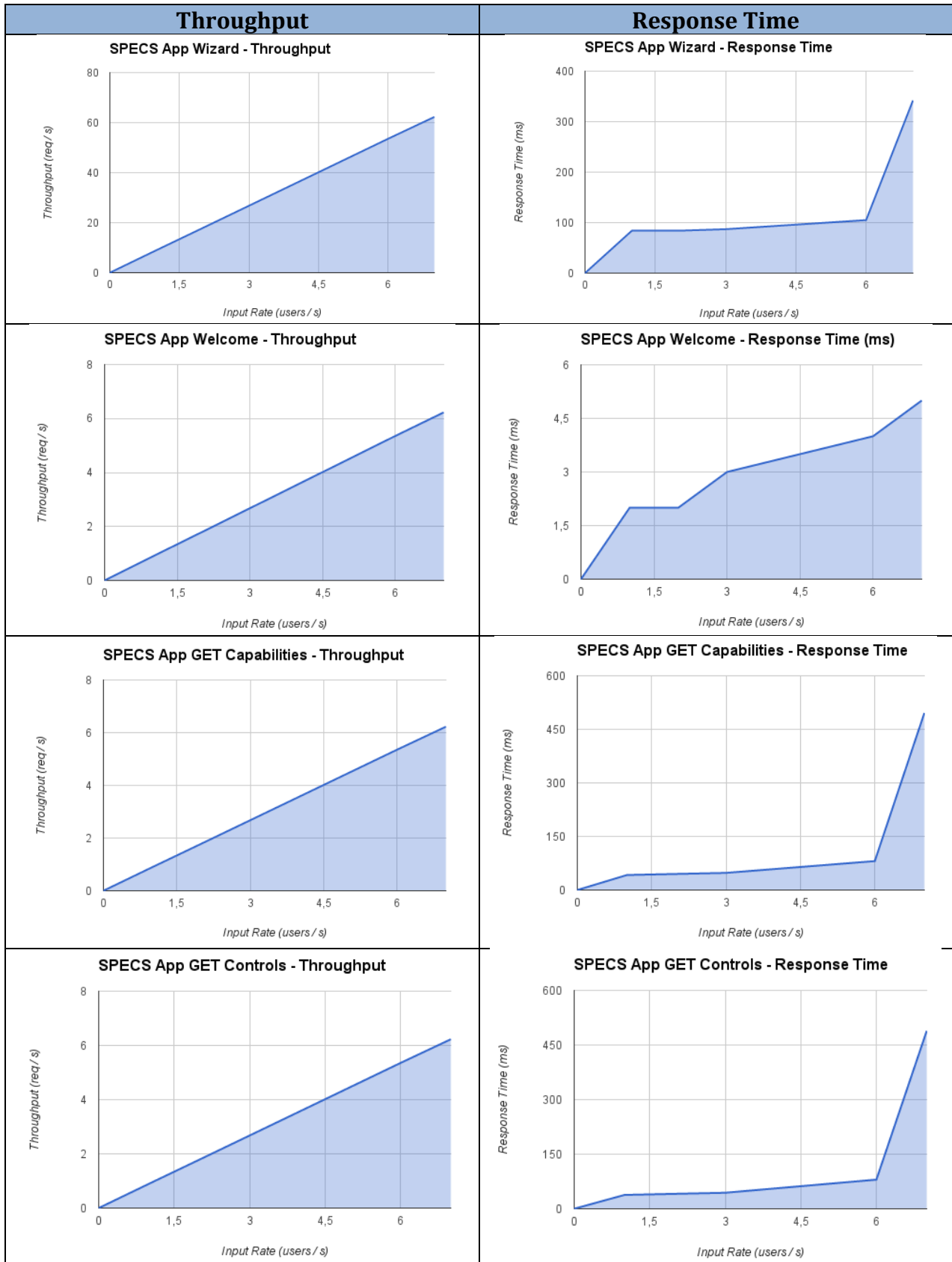
User Profile	Description	Scripts
Wizard	Perform the full negotiation process, selecting all the proposed capabilities	<ul style="list-style-type: none"> WebAppSign.scala

Table 9. AAAaaS application user profiles for performance tests

In the table below we present performance results for the application. In the left column we illustrate the *throughput* granted by the SPECS application, and in the right column we present the *response time* of the associated tests. In the graphs below, we report the values for each action of the application wizard. The application wizard implies the execution of all the services offered by the SPECS Platform, according to the flow discussed in deliverable D1.3.

It is worth noticing that the application can manage about 7 users per second (i.e. about 420 users per minute). The wizard results in the acquisition of a varying number of VMs between 3 and 5, as a consequence 420 user per minute, implies about 2100 VMs delivered per minute (EMC’s testbed can deliver at most 15000 VMs in total).

According to such result, even the minimal configuration proposed (all components hosted in the same VM, excluded only the Chef server) with only 1 GB of RAM memory enables to manage a little datacenter (few hundreds of VMs available).



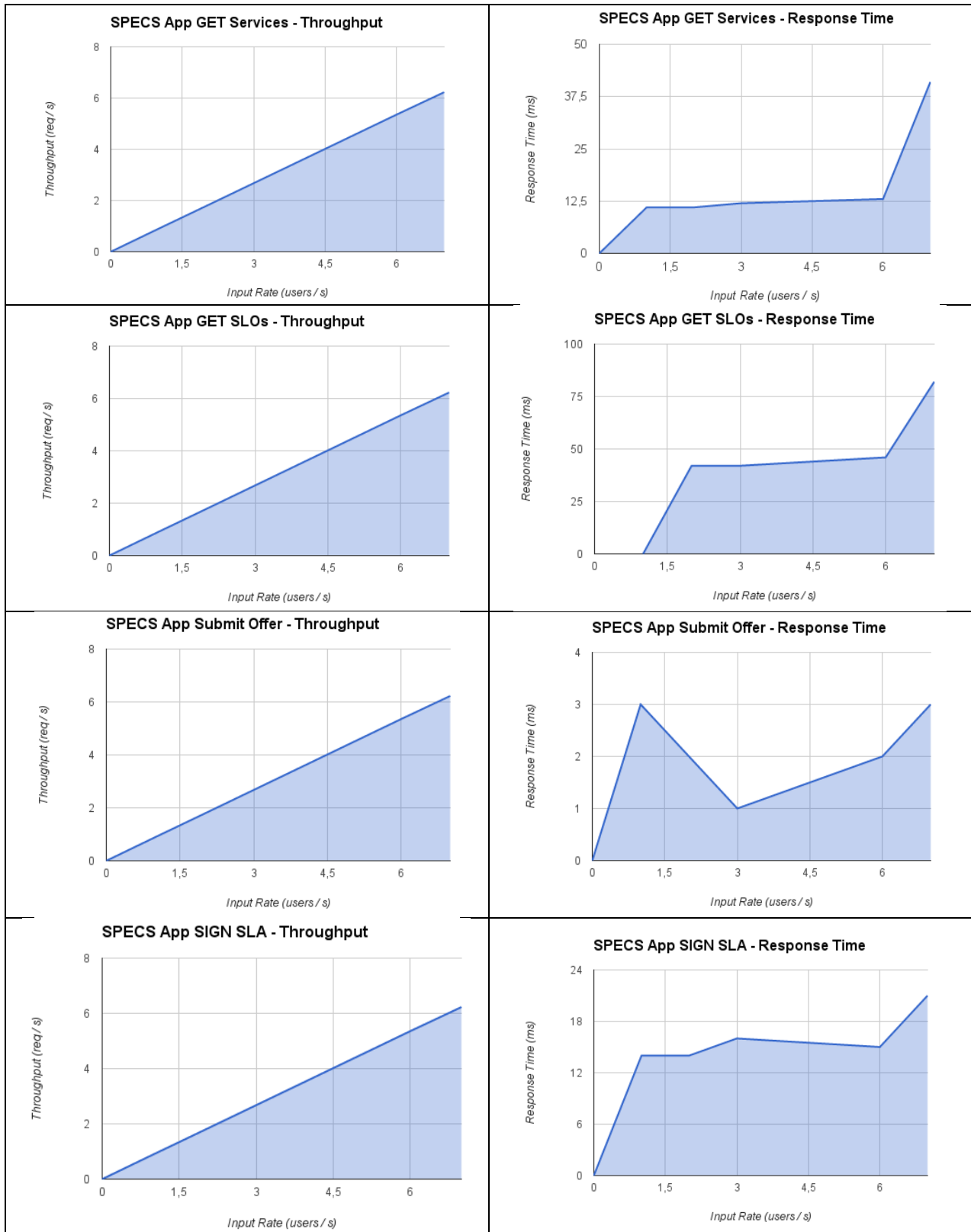


Table 10. Performance results for the AAAaaS application on EMC's testbed

7. Installation

In order to replicate the EMC testbed, the system architecture and configuration of components described in deliverable D5.3 is required. As illustrated in Section 5.1, at a minimum the system requires three hosts (two running ESXi, one running the management server in Windows), a storage array (such as VMAX) and a fibre-channel switch with the appropriate zoning in place so that all components can communication with each other. On the software side, vCenter is required to manage the datacenter's compute and network resources while ViPR must be installed and configured to discover and manage the connected storage array. Finally, a standalone Chef server is required to deploy and configure VMs running services.

While this deliverable (and D5.3) focuses on the ViPR product as the primary virtual storage solution, the SPECS application is designed to be platform agnostic and supports platforms such as Amazon EC, Openstack and vSphere. Irrespective of the storage services, it is possible to still negotiate, enforce and monitor an SLA for the AAA and E2EE capabilities.

The EMC testbed contains the following VMs required for the demo:

- **Chef server:** A full installation of the Chef server which should have, at least, an administrator account enabled. This server is a central repository for registers nodes and cookbooks (policies applied to nodes). The cookbooks are available from the following SPECS repositories:
 - <https://bitbucket.org/specs-team/specs-core-external-repository>
 - https://bitbucket.org/specs-team/specs-core-enabling_platform-repository
 - <https://bitbucket.org/specs-team/specs-core-enforcement-repository>
- **Chef workstation:** Central workstation configured to interact with the Chef Server and nodes, from which users can perform tasks such as:
 - Develop cookbooks and recipes
 - Manage nodes, roles, resources, etc.
 - Configure polices
- **ViPR vApp:** A virtual application installed on a number of VMs (various standard configurations available) that hosts the ViPR controller (see D5.3 for further detail).
- **SPECS Enabling Platform:** Hosts the SPECS application.

To install the SPECS platform on the VM with already installed and configured Chef Client, the following recipes should be executed:

- enabling-platform::apache-tomcat-v7
- monitoring::event-hub
- monitoring::event-archiver
- monitoring::ctp-server
- monitoring::ctp-adaptor
- monitoring::wui
- enforcement::implementation
- sla-negotiation::slo-manager
- enforcement::diagnosis
- enforcement::rds
- enforcement::planning
- monitoring::monipoli
- enforcement::aaa

- sla-negotiation::security-reasoner
- applications::web-container-app-rev2
- sla-platform::metric-catalogue
- sla-platform::sla-manager
- sla-platform
- sla-platform::service-manager

Changes in the configurations which define the base address are needed to allow it to work (*config.js* and *sws_conf.properties*).

Note that installation guides for core components of the SPECS architecture are available in dedicated deliverables (for the SLA Platform in D1.4.2, for the Negotiation module in D2.3.2, for the Monitoring module in deliverables D3.4.1 and D3.4.2, and for the Enforcement module in deliverables D4.3.2 and D4.3.3). Installation guides for the AAA mechanism is available in deliverable D4.3.3, and guides for DBB and E2EE mechanisms are available in deliverable D5.2.2.

8. Usage

The SPECS AAAaaS application is accessible to End-users via a web browser. Like other SPECS applications, it presents the negotiation wizard, used by End-users to negotiate required capabilities and obtain an SLA to sign. As shown in Figure 10, the AAAaaS application negotiation wizard enables the selection of two different services, namely the *Secure Storage AAA* and *Secure Storage with EMC ViPR*. Indeed, as pointed out in Section 5, through the AAAaaS application the End-user first acquires the virtual storage resources from the EMC datacentre, and then deploys the secure storage services (including DBB, E2EE and AAA mechanisms) on top of them. Both processes entail a negotiation since, as discussed in D5.3, it is possible to negotiate performance and security features related to virtual storage resources through the integration of ViPR and SPECS services. Hence, the flow for the deployment of the AAAaaS application is as follows:

1. Acquire virtual storage resources
2. Deploy secure storage service

We skip the first step as it has been extensively discussed in deliverable D5.3. In the remainder we report the second step.

The screenshot displays the 'SPECS Application Negotiation Wizard' interface. At the top, a progress bar shows six steps: 'Start', 'Select Service' (highlighted in blue), 'Select Capabilities', 'Select Security Controls', 'Define Agreement Terms', and 'SLA Overview'. Below the progress bar are 'Previous' and 'Next' navigation buttons. The main content area is titled 'Service Selection' and contains a text box with instructions: 'Select the type of service that you want to acquire. The needed resources will be acquired from an External CSP. Note that, in this version of the application, the selection of the provider is not enabled, and the resources will be acquired from Amazon WS.' Below this, there is a section titled 'Type of service' with two radio button options: 'SECURE_STORAGE_AAA' (selected) and 'Secure Storage with EMC ViPR'. Each option has a 'SHOW TEMPLATE' button next to it.

Figure 10. SPECS negotiation (service selection)

Once the storage resources have been acquired and related capabilities have been enforced according to the SLA, the next step is to launch the *SECURE_STORAGE_AAA* negotiation process. By selecting this offer, the End-user will be prompted with the security capabilities available under the *SECURE_STORAGE_AAA* service. These capabilities, shown in Figure 11, cover three core areas: database and backup as-a-Service, end-to-end encryption, and authentication, authorization, and auditing.

SPECS Application Negotiation Wizard

Start > Select Service > Select Capabilities > Select Security Controls > Define Agreement Terms > SLA Overview

Previous Next

Security Capabilities

Select one or more security capabilities that you want to add to the service.
A security capability is a collection of security controls, i.e. safeguards and countermeasures, that can be applied over your services.
In case you do not select any capability, the service will be delivered with no additional security features or guarantees.

Capability Name	Description
<input checked="" type="checkbox"/> Database and Backup as-a-service	Capability of surviving to incidents that compromise the availability and/or integrity of data stored remotely by providing backup service and the detection of WS and RF violations
<input checked="" type="checkbox"/> End-to-end Encryption	Capability of providing client-side encryption enforcing confidentiality.
<input checked="" type="checkbox"/> Authentication, Authorization, and Auditing	Capability of providing identity management and access control functionalities

Figure 11. SPECS negotiation (capability selection)

After the selection of the requested capabilities, the SPECS application presents the End-user with the available controls (as shown in Figure 12 and Figure 13). Figure 12 highlights the controls available under the *Database and Backup as-a-Service* capability, while Figure 13 highlights the controls available under the *End-to-end Encryption* and *Authentication, Authorization, and Auditing* capability.

SPECS Application Negotiation Wizard

Start Select Service Select Capabilities Select Security Controls Define Agreement Terms SLA Overview

Previous Next

Security Controls

For each selected capability, choose the security controls you are interested in and assign them a score according to the importance you give to them. The listed controls belong to specific security frameworks, such as NIST Security Control Framework and CSA Cloud Control Matrix. Each control is associated with a set of security metrics, on top of which, in the next step, you will be able to define SLOs that will be inserted in a Security SLA. Note that, in case you do not select any control, you will not be able to define SLOs in the next step.

Database and Backup as-a-service

CCM Control framework v3.0
 Select All

<input type="checkbox"/> Infrastructure and Virtualization Security - Change Detection Importance weight: <input type="text" value="MEDIUM"/>	The provider shall ensure the integrity of all virtual machine images at all times. Any changes made to virtual machine images must be logged and an alert raised regardless of their running state (e.g. dormant, off, or running).
<input type="checkbox"/> Business Continuity Mgmt and Op Resilience - Business Continuity Planning Importance weight: <input type="text" value="MEDIUM"/>	A consistent unified framework for business continuity planning and plan development shall be established, documented and adopted to ensure all business continuity plans are consistent in addressing priorities for testing, maintenance, and information security requirements.
<input type="checkbox"/> Business Continuity Mgmt and Op Resilience - Retention Policy Importance weight: <input type="text" value="MEDIUM"/>	Policies and procedures shall be established, for defining and adhering to the retention period of any critical asset as per established policies and procedures, as well as applicable legal, statutory, or regulatory compliance obligations. Backup and recovery measures shall be incorporated as part of business continuity planning and tested accordingly for effectiveness.
<input type="checkbox"/> Application and Interface Security - Data Integrity Importance weight: <input type="text" value="MEDIUM"/>	Data input and output integrity routines (i.e., reconciliation and edit checks) shall be implemented for application interfaces and databases to prevent manual or systematic processing errors, corruption of data, or misuse.

Figure 12. SPECS negotiation (security controls selection) (1/2)

End-to-end Encryption

CCM Control framework v3.0

Select All

<input checked="" type="checkbox"/> Encryption and Key Management - Entitlement Importance weight: MEDIUM	Keys must have identifiable owners (binding keys to identities) and there shall be key management policies.
<input checked="" type="checkbox"/> Encryption and Key Management - Sensitive Data Protection Importance weight: MEDIUM	Policies and procedures shall be established, for the use of encryption protocols for protection of sensitive data in storage (e.g., file servers, databases, and end-user workstations), data in use (memory), and data in transmission (e.g., system interfaces, over public networks, and electronic messaging).

Authentication, Authorization, and Auditing

CCM Control framework v3.0

Select All

<input checked="" type="checkbox"/> Identity and Access Management - Credential Lifecycle/Provision Management Importance weight: MEDIUM	"User access policies and procedures shall be established, and supporting business processes and technical measures implemented, for ensuring appropriate identity, entitlement, and access management for all internal corporate and customer (tenant) users with access to data and organizationally-owned or managed (physical and virtual) application interfaces and infrastructure network and systems components...
<input checked="" type="checkbox"/> Identity and Access Management - Policies and Procedures Importance weight: MEDIUM	Policies and procedures shall be established to store and manage identity information about every person who accesses IT infrastructure and to determine their level of access. Policies shall also be developed to control access to network resources based on user identity.

Figure 13. SPECS negotiation (security controls selection) (2/2)

Each of these controls is mapped to a specific security metric. For example, the End-to-end Encryption controls Encryption and Key Management – Entitlement/Sensitive Data Protection map to the E2EE – Confidentiality SLO. In the next step of the negotiation process, the End-user defines SLOs (i.e. selects preferred metrics, specifies their preferred values, and assigns to them *importance weights*), as shown in Figure 14.

SPECS Application Negotiation Wizard

Start ➤ Select Service ➤ Select Capabilities ➤ Select Security Controls ➤ Define Agreement Terms ➤ SLA Overview

Previous
Next

Service Level Objectives

Select the security metrics that you want to monitor and specify, for each of them, your importance score and the SLO you want to achieve. The SLO is represented by a comparison expression over a metric. If you are not a security expert just adopt the default values and go ahead with the next step. Note that, in case you do not select any metric, you will not be able to obtain any guarantee over implemented security features.

E2EE

Select All

Confidentiality
The existence of certification.

Importance: MEDIUM

Expression: equal yes

[Show metric details](#)

DBB

Select All

Write-Serializability
The activation of write serializability

Importance: MEDIUM

Expression: equal yes

[Show metric details](#)

Read-Freshness
The activation of read freshness

Importance: MEDIUM

Expression: equal yes

[Show metric details](#)

Integrity
The activation of integrity.

Importance: MEDIUM

Expression: equal yes

[Show metric details](#)

AAA

Select All

AAA Log Completeness
The level of completeness of report

Importance: MEDIUM

Expression: equal MEDIUM

[Show metric details](#)

Figure 14. SPECS negotiation (definition of SLOs)

Once the End-user has specified her/his requirements, the SPECS portal will allow the End-user to *Request Offers* (as shown in Figure 15), which returns all available SLA Offers meeting the End-user requirements (as shown in Figure 16).

The screenshot shows the 'SPECS Application Negotiation Wizard' interface. At the top, a progress bar indicates the current step is 'SLA Overview', with previous steps being 'Start', 'Select Service', 'Select Capabilities', 'Select Security Controls', and 'Define Agreement Terms'. Below the progress bar is a 'Previous' button. The main content area is titled 'Negotiated SLA' and contains a text box with the following message: 'A Security SLA has been built according to your choices during the negotiation process. Check it and submit it in order to have it stored in your collection of negotiated SLAs. The next steps of the SPECS Application will enable you to sign and implement the SLA.' Below this text box is a blue button labeled 'REQUEST OFFERS'.

Figure 15. SPECS negotiation (request offers)

The screenshot shows the 'SPECS Application Negotiation Wizard' interface, similar to Figure 15. The progress bar is the same, but the 'SLA Overview' step is now highlighted in blue. Below the progress bar is a 'Previous' button. The main content area is titled 'Negotiated SLA' and contains the same text box as in Figure 15. Below the text box is a blue button labeled 'REQUEST OFFERS'. Underneath this button is a form field labeled 'SLA Offer' containing the text '570d05600cf2a3af1956d7e1_offer1' and a radio button. At the bottom of the form are two blue buttons: 'SUBMIT OFFER' and 'SHOW OFFER'.

Figure 16. SPECS negotiation (received SLA Offers)

The End-user can then choose each SLA and select the option *Show Offer* (see Figure 17) which will display an overview of the SLA Offer (in XML format). This form displays the negotiated capabilities, controls, metrics, their associated values, and the importance weight associated with them. The *importance* is additionally specified, selected by the user that enables SPECS to make more informed decisions about the service a provider is offering.

SPECS Application Negotiation Wizard

Start > Select Service > Select Capabilities > Select Security Controls > Define Agreement Terms > SLA Overview

Previous

Negotiated SLA

A Security SLA has been built according to your choices during the negotiation process. Check it and submit it in order to have it stored in your collection of negotiated SLAs. The next steps of the SPECS Application will enable you to sign and implement the SLA.

REQUEST OFFERS

SLA Offer

570d05600cf2a3af1956d7e1_offer1

SUBMIT OFFER
SHOW OFFER

```

<wsag:AgreementOffer xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:specs="http://www.specs-project.eu/resources/schemas/xml/SLAtemplate" xmlns:wsag="http://schemas.ggf.org/graap/2007/03/ws-agreement" xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:nist="http://www.specs-project.eu/resources/schemas/xml/control_frameworks/nist" xmlns:ccm="http://www.specs-project.eu/resources/schemas/xml/control_frameworks/ccm">
  <wsag:Name>
    570d05600cf2a3af1956d7e1_offer1
  </wsag:Name>
  <wsag:Context>
    <wsag:AgreementInitiator>
      specs-customer-2
    </wsag:AgreementInitiator>
    <wsag:AgreementResponder>
      $SPECS-APPLICATION
    </wsag:AgreementResponder>
    <wsag:ServiceProvider>
      AgreementResponder
    </wsag:ServiceProvider>
  </wsag:Context>
</wsag:AgreementOffer>

```

Figure 17. SPECS negotiation (reviewing SLA Offer)

On submitting the chosen SLA offer, the End-user is asked to sign it. As can be seen in Figure 18, the End-user can choose to *Sign the SLA*, display the *Template*, or display the *Synthetic* view. The *Synthetic* view gives the End-user a simplified, tabular view of the capabilities captured by the SLA.

Sign SLA

This page shows the list of negotiated SLAs.
Click on the "Sign SLA" button to sign the selected SLA.

SLA ID
570D05600CF2A3AF1956D7E1

Metric Name	Operation	Value	Importance
specs_DBB_M17	EQUAL	yes	MEDIUM
specs_DBB_M18	EQUAL	yes	MEDIUM
specs_DBB_M25	EQUAL	yes	MEDIUM
specs_e2ee_M19	EQUAL	yes	MEDIUM
specs_aaa_m3	EQUAL	MEDIUM	MEDIUM

Figure 18. SPECS negotiation (sign SLA)

Now that the SLA has been signed, the End-user must select *Implement SLA* (Figure 19) to complete the process and enforce the signed SLA.

Implement SLA

This page shows the list of signed SLAs.
Click on the "Implement SLA" button to start the implementation process.

SLA ID
570D05600CF2A3AF1956D7E1

Figure 19. SPECS Negotiation (implement SLA)

Implementing the SLA triggers the deployment of the secure storage service according to the signed SLA. SPECS provides a view to the End-user to monitor the status of the deployment, as can be seen in Figure 20. Options to view the SLA in its entirety or open the *Synthetic* view are also available so that End-users can easily review and monitor their currently enforced SLAs.

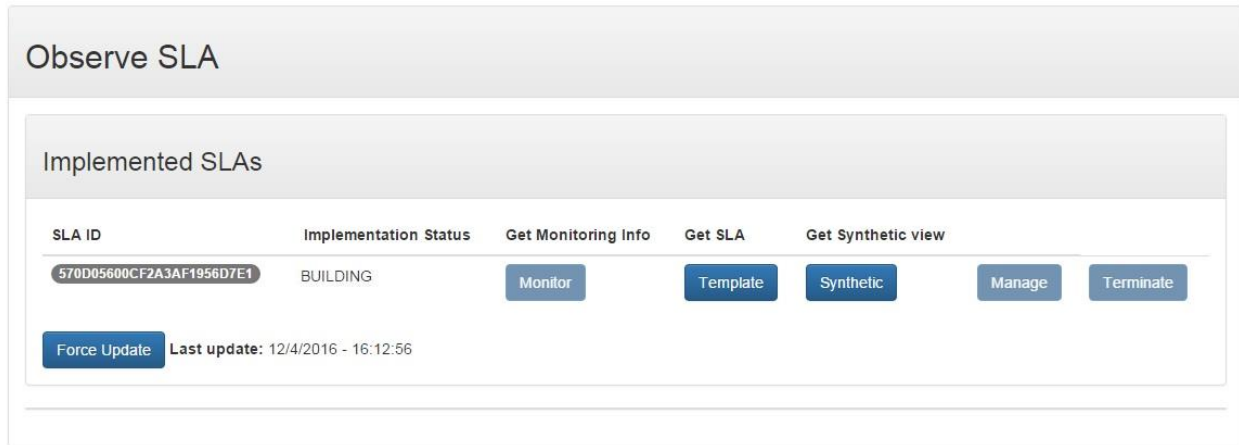


Figure 20. SPECS Negotiation (observe SLA)

For the usage of the E2EE, DBB, and AAA components please see deliverables D5.2.2 and D4.3.3.

9. Testing

The scenario presented in this deliverable is composed of the SPECS framework, SPECS mechanisms, and ViPR solution. The SPECS framework and the security mechanisms have been developed and tested in other tasks of the project, thus we refer the interested reader to deliverables D1.4.2, D2.3.2, D3.4.2, and D4.5.3 for tests associated to the core modules, deliverable D1.5.2 for the integration tests associated to the SPECS core components, deliverable D4.5.3 for tests associated to the AAA mechanism and deliverable D5.2.2 for tests associated to the DBB and E2EE mechanisms. Since the ViPR application has been developed in task T5.3, we refer the reader to see associated tests in deliverable D5.3.

In the following tables we present all details for the integration test aimed at evaluating the behaviour of the AAAaaS application.

Example ID	<i>App-D1</i>
Description	This scenario integrates the AAAaaS application with the AAA security mechanism.
Link	https://bitbucket.org/specs-team/specs-integration-test-appd
Core components	All
Security mechanisms	AAA, ViPR
SPECS applications	Default SPECS application (AAAaaS application)
Inputs	An SLA template with the AAA capabilities and corresponding SLOs, security mechanism AAA.
Expected results	(1) Correct capabilities, security controls and SLOs are presented by the SPECS application in the negotiation GUI, (2) SLA offers are created correctly corresponding to selected SLOs, (3) negotiation finishes successfully, (4) SLA is signed, (5) SLA is implemented successfully, and (6) VMs are provisioned successfully and correct AAA components are installed.
Outputs	All points defined above were successfully accomplished and the results were all as expected.
Comments	The Selenium ¹⁰ tool was used to automate interaction with SPECS web application.

Table 11. Integration test *App-D1* for the AAA-as-a-Service application

¹⁰ <http://www.seleniumhq.org/>
SPECS Project – Deliverable 5.4

Example ID	<i>App-D2</i>
Description	This scenario extends the <i>App-D1</i> integration scenario with the DBB and E2EE mechanisms.
Link	https://bitbucket.org/specs-team/specs-integration-test-appd
Core components	All
Security mechanisms	AAA, DBB, E2EE, ViPR
SPECS applications	Default SPECS application (AAAaaS application)
Inputs	An SLA template with AAA, DBB, and E2EE capabilities and corresponding SLOs, security mechanisms AAA, DBB, and E2EE.
Expected results	(1) Correct capabilities, security controls and SLOs are presented by the SPECS application in the negotiation GUI, (2) SLA offers are created correctly corresponding to selected SLOs, (3) negotiation finishes successfully, (4) SLA is signed, (5) SLA is implemented successfully, and (6) VMs are provisioned successfully and correct AAA, DBB, and E2EE components are installed.
Outputs	All points defined above were successfully accomplished and the results were all as expected.
Comments	The Selenium tool was used to automate interaction with SPECS web application.

Table 12. Integration test *App-D2* for the AAA-as-a-Service application

The integration of all components in EMC testbed highlights the low footprint and overhead introduced by SPECS and shows how the extreme automation introduced by SPECS could be used in a real-world environment with low effort from CSP side in the customization and integration of the SPECS platform.

Considering that, for example, the deployment of the AAA machine with the installation of Apache Tomcat and the AAA mechanism takes less than a couple of minutes in average. This performance could be improved by creating a high priority resource schedule for the machine or raising the number of CPUs associated and the RAM of the template. As in the real environments, this type of allocation and prioritization usually has a price that the End-user can decide to afford or not, depending on the requirements of her/his application.

10. Conclusions

By successfully integrating ViPR and E2EE mechanism using the SPECS AAA mechanism, we demonstrated that SPECS identity management capabilities provided by the AAA mechanism can be used as an identity management *as-a-service* component. Furthermore, we demonstrated that the SPECS AAA mechanism can be used as easily as any other external identity management provider, such as Google IAM. There is a notable distinction however, the SPECS AAA mechanism can be deployed automatically by the SPECS platform and is customizable through the SLA negotiation process (please see deliverable D4.3.3 for capabilities and security metrics offered by the AAA mechanism).

Additionally, we demonstrated that the E2EE mechanism can be plugged-in to systems other than what was initially defined (OpenStack, Amazon EC).

The development of the application led the deployment and validation of a novel approach to the instantiation of cloud services on cloud storage acquired through a broker. This deliverable has demonstrated that it is possible to give fine grained control to the End-user over not only the security mechanisms they wish to deploy but also over the hardware on which they are deployed. This enables further visibility over the entire End-user environment and guarantees protection of services and End-user applications/data through the use of security SLAs.

The core security mechanism deployed for the demonstration of this advance in cloud service brokering was the identity management functionalities (the AAA mechanism). The demonstrator successfully showed how such a mechanism could be automatically provisioned on a newly acquired secure storage service.

Building on the infrastructure developed and validated in D5.3, this deliverable has made further progress towards the realisation of the ngDC scenario. End-users now have the capability of defining (i) the storage on which their data will be securely stored (D5.3) and (ii) the compute resources which will run their services (D5.4). In addition, it is possible for End-users to deploy security services such as the AAA mechanism that is tailored to their security requirements. In particular, the AAA mechanism offers End-users increased control over the access control rights to their data and services. This, alongside other mechanisms (e.g. E2EE), increases the visibility and control over the security of End-user services and data. These benefits add value to cloud services by improving customer confidence that is required in order to achieve greater adoption levels of cloud services.

11. Bibliography

- [1] EMC Corporation, “*ViPR Controller*”, 2016. Available online, <http://www.emc.com/products/storage/software-defined-storage/vipr-controller.htm>, last accessed in April 2016.
- [2] Chef Software, “*Chef*”, 2008. Available online, <https://www.chef.io/>, last accessed in April 2016.
- [3] SPECS, “*SPECS Core Enforcement Repository*”, 2015. Available online, <https://bitbucket.org/specs-team/specs-core-enforcement-repository>, last accessed in April 2016.
- [4] National Institute of Standards and Technology (NIST), “*Security and privacy controls for federal information systems and organizations*”, NIST 800-53v4, 2013. Available online, <http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-53r4.pdf>, last accessed in February 2016.
- [5] Cloud Security Alliance, “*Cloud Controls Matrix Working Group*”, 2015. Available online, <https://cloudsecurityalliance.org/group/cloud-controls-matrix/>, last accessed in February 2016.
- [6] T. Wu, “*The Stanford SRP Homepage*”, 2016. Available online, <http://srp.stanford.edu/>, last accessed in April 2016.
- [7] SPECS, “*SPECS Enforcement Broker*”, 2015. Available online, <https://bitbucket.org/specs-team/specs-core-enforcement-broker>, last accessed in April 2016.
- [8] VMWare, “*VMWare ESXi*”, Available online, <https://www.vmware.com/products/esxi-and-esx/overview>, Last accessed in April 2016.
- [9] EMC, “*EMC Symmetrix VMAX 20K Specification*”, 2016. Available online, <http://www.emc.com/collateral/hardware/specification-sheet/h6176-symmetrix-vmax-20k-ss.pdf>, last accessed in April 2016.
- [10] EMC, “*CoprHD*”, 2016. Available online, <https://github.com/CoprHD>, Last accessed in April 2016.
- [11] VMWare, “*Configuration Maximums, vSphere 6.0*”, 2015. Available online, <https://www.vmware.com/pdf/vsphere6/r60/vsphere-60-configuration-maximums.pdf>, last accessed in April 2016.
- [12] Google, “*chrom-identity – launchWebAuthFlow*”, 2016. Available online, <https://developer.chrome.com/apps/identity#method-launchWebAuthFlow>, last accessed in April 2016.
- [13] Koofr d.o.o., “*Koofr*”, 2015. Available online, <http://koofr.eu/>, last accessed in April 2016.
- [14] Chef, “*About the Chef Server*”, 2008. Available online, https://docs.chef.io/server_components.html, last accessed in April 2016.
- [15] Gatling Corp, “*Gatling*”, 2015. Available online, <http://gatling.io/#/>, last accessed in March 2016.

Appendix 1. The AAAaaS SLA Template

In the following, we report the SLA Template created for the AAAaaS application with the default values for all associated security metrics (in the XML format). The important parts are highlighted in yellow. Note that the template only refers to the process described in Section 8 about the deployment of the secure storage service after the acquisition of the virtual storage resources.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<wsag:AgreementOffer
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:specs="http://www.specs-project.eu/resources/schemas/xml/SLAtemplate"
  xmlns:wsag="http://schemas.ggf.org/graap/2007/03/ws-agreement"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:nist="http://www.specs-project.eu/resources/schemas/xml/control_frameworks/nist"
  xmlns:ccm="http://www.specs-project.eu/resources/schemas/xml/control_frameworks/ccm">

  <wsag:Name>SECURE_STORAGE_AAA</wsag:Name>

  <wsag:Context>
    <wsag:AgreementInitiator>specs-customer-2</wsag:AgreementInitiator>
    <wsag:AgreementResponder>$SPECS-APPLICATION</wsag:AgreementResponder>
    <wsag:ServiceProvider>AgreementResponder</wsag:ServiceProvider>
    <wsag:ExpirationTime>2016-04-30T06:00:00+03:00</wsag:ExpirationTime>
    <wsag:TemplateName>SECURE_STORAGE_WITH_AAA</wsag:TemplateName>
  </wsag:Context>

  <wsag:Terms>
    <wsag:All>
      <wsag:ServiceDescriptionTerm
        wsag:Name="Secure Storage with AAA"
        wsag:ServiceName="SecureStorageAAA">
        <specs:serviceDescription>
          <specs:serviceResources>
            <specs:resourcesProvider
              id="aws-ec2"
              name="Amazon"
              zone="us-east-1"
              maxAllowedVMs="20"
              description=""
              label="">
            <specs:VM
              appliance="us-east-1/ami-ff0e0696"
              hardware="t1.micro"
              description="open suse 13.1 on amazon EC2"/>
            </specs:resourcesProvider>
          </specs:serviceResources>

          <specs:capabilities>
            <specs:capability
              id="DBB"
              name="Database and Backup as-a-service"
              description="Capability of surviving to incidents that
                compromise the availability and/or integrity of
                data stored remotely by providing backup
                service and the detection of WS and RF
                violations"
              mandatory="false">
            <specs:controlFramework
              id="CCM_v3.0"
              frameworkName="CCM Control framework v3.0">
          </specs:capabilities>
        </specs:serviceDescription>
      </wsag:ServiceDescriptionTerm>
    </wsag:All>
  </wsag:Terms>
</wsag:AgreementOffer>
```

```
<specs:CCMsecurityControl
  id="IVS-02"
  name="Infrastructure and Virtualization Security -
    Change Detection"
  control_domain="IVS">
  <ccm:description>The provider shall ensure the
    integrity of all virtual machine images at all times.
    Any changes made to virtual machine images must be
    logged and an alert raised regardless of their
    running state (e.g. dormant, off, or running).
  </ccm:description>
  <ccm:importance_weight>MEDIUM</ccm:importance_weight>
</specs:CCMsecurityControl>

<specs:CCMsecurityControl
  id="BCR-01"
  name="Business Continuity Mgmt and Op Resilience -
    Business Continuity Planning"
  control_domain="BCR">
  <ccm:description>A consistent unified framework for
    business continuity planning and plan development
    shall be established, documented and adopted to
    ensure all business continuity plans are consistent
    in addressing priorities for testing, maintenance,
    and information security requirements.
  </ccm:description>

  <ccm:importance_weight>MEDIUM</ccm:importance_weight>
</specs:CCMsecurityControl>

<specs:CCMsecurityControl
  id="BCR-11"
  name="Business Continuity Mgmt and Op Resilience -
    Retention Policy" control_domain="BCR">
  <ccm:description>Policies and procedures shall be
    established, for defining and adhering to the
    retention period of any critical asset as per
    established policies and procedures, as well as
    applicable legal, statutory, or regulatory compliance
    obligations. Backup and recovery measures shall be
    incorporated as part of business continuity planning
    and tested accordingly for effectiveness.
  </ccm:description>
  <ccm:importance_weight>MEDIUM</ccm:importance_weight>
</specs:CCMsecurityControl>

<specs:CCMsecurityControl
  id="AIS-03"
  name="Application and Interface Security - Data
    Integrity" control_domain="AIS">
  <ccm:description>Data input and output integrity
    routines (i.e., reconciliation and edit checks) shall
    be implemented for application interfaces and
    databases to prevent manual or systematic processing
    errors, corruption of data, or misuse.
  </ccm:description>
  <ccm:importance_weight>MEDIUM</ccm:importance_weight>
</specs:CCMsecurityControl>
</specs:controlFramework>
</specs:capability>

<specs:capability
  id="E2EE"
  name="End-to-end Encryption"
  description="Capability of providing client-side encryption
    enforcing confidentiality."
  mandatory="false">
```

```
<specs:controlFramework
  id="CCM_v3.0"
  frameworkName="CCM Control framework v3.0">
  <specs:CCMsecurityControl
    id="EKM-01"
    name="Encryption and Key Management - Entitlement"
    control_domain="EKM">
    <ccm:description>Keys must have identifiable owners
      (binding keys to identities) and there shall be key
      management policies.
    </ccm:description>
    <ccm:importance_weight>MEDIUM</ccm:importance_weight>
  </specs:CCMsecurityControl>

  <specs:CCMsecurityControl
    id="EKM-03"
    name="Encryption and Key Management - Sensitive Data
      Protection" control_domain="EKM">
    <ccm:description>Policies and procedures shall be
      established, for the use of encryption protocols for
      protection of sensitive data in storage (e.g., file
      servers, databases, and end-user workstations), data
      in use (memory), and data in transmission (e.g.,
      system interfaces, over public networks, and
      electronic messaging).
    </ccm:description>
    <ccm:importance_weight>MEDIUM</ccm:importance_weight>
  </specs:CCMsecurityControl>
</specs:controlFramework>
</specs:capability>

<specs:capability
  id="AAA"
  name="Authentication, Authorization, and Auditing"
  description="Capability of providing identity management and
    access control functionalities"
  mandatory="false">
  <specs:controlFramework
    id="CCM_v3.0"
    frameworkName="CCM Control framework v3.0">
    <specs:CCMsecurityControl
      id="IAM-02"
      name="Identity & Access Management - Credential
        Lifecycle/Provision Management"
      control_domain="IAM">
      <ccm:description>"User access policies and procedures
        shall be established, and supporting business
        processes and technical measures implemented, for
        ensuring appropriate identity, entitlement, and
        access management for all internal corporate and
        customer (tenant) users with access to data and
        organizationally-owned or managed (physical and
        virtual) application interfaces and infrastructure
        network and systems components..."
      </ccm:description>
      <ccm:importance_weight>MEDIUM</ccm:importance_weight>
    </specs:CCMsecurityControl>

    <specs:CCMsecurityControl
      id="IAM-04"
      name="Identity & Access Management - Policies and
        Procedures"
      control_domain="IAM">
      <ccm:description>Policies and procedures shall be
        established to store and manage identity information
        about every person who accesses IT infrastructure
        and to determine their level of access. Policies
        shall also be developed to control access to network
```



```
        resources based on user identity.
        </ccm:description>
        <ccm:importance_weight>MEDIUM</ccm:importance_weight>
    </specs:CCMsecurityControl>
</specs:controlFramework>
</specs:capability>

</specs:capabilities>

<specs:security_metrics>
  <specs:Metric
    name="Write-Serializability"
    referenceId="write_serializability_M17">
    <specs:MetricDefinition>
      <specs:unit name="">
        <specs:enumUnit>
          <specs:enumItemsType>boolean</specs:enumItemsType>
          <specs:enumItems>
            <specs:enumItem></specs:enumItem>
            <specs:enumItem></specs:enumItem>
          </specs:enumItems>
        </specs:enumUnit>
      </specs:unit>
      <specs:scale>
        <specs:Quantitative>Ratio</specs:Quantitative>
      </specs:scale>
      <specs:expression>The activation of write
        serializability</specs:expression>
      <specs:definition>This metric ensures the EU consistency
        among updates of the stored data. In case of WS
        violations, the EU will be notified and the system will
        be restored to the state of the last completed
        update.</specs:definition>
      <specs:note></specs:note>
    </specs:MetricDefinition>
  </specs:Metric>

  <specs:Metric
    name="Read-Freshness"
    referenceId="read_freshness_M18">
    <specs:MetricDefinition>
      <specs:unit name="">
        <specs:enumUnit>
          <specs:enumItemsType>boolean</specs:enumItemsType>
          <specs:enumItems>
            <specs:enumItem></specs:enumItem>
            <specs:enumItem></specs:enumItem>
          </specs:enumItems>
        </specs:enumUnit>
      </specs:unit>
      <specs:scale>
        <specs:Quantitative>Ratio</specs:Quantitative>
      </specs:scale>
      <specs:expression>The activation of read
        freshness</specs:expression>
      <specs:definition>This metric ensures the EU that the
        requested data will always be fresh as of the last
        update. In case of RF violations, the EU will be notified
        and the system will be restored to the state of the last
        completed backup.</specs:definition>
      <specs:note></specs:note>
    </specs:MetricDefinition>
  </specs:Metric>

  <specs:Metric
    name="Integrity"
    referenceId="integrity_M25">
```

```
<specs:MetricDefinition>
  <specs:unit name="">
    <specs:enumUnit>
      <specs:enumItemsType>boolean</specs:enumItemsType>
      <specs:enumItems>
        <specs:enumItem></specs:enumItem>
        <specs:enumItem></specs:enumItem>
      </specs:enumItems>
    </specs:enumUnit>
  </specs:unit>
  <specs:scale>
    <specs:Quantitative>Ratio</specs:Quantitative>
  </specs:scale>
  <specs:expression>The activation of
  integrity.</specs:expression>
  <specs:definition>This metric ensures the EU
  integrity of the stored data.</specs:definition>
  <specs:note></specs:note>
</specs:MetricDefinition>
</specs:Metric>

<specs:Metric
  name="Confidentiality"
  referenceId="confidentiality_M19">
  <specs:MetricDefinition>
    <specs:unit name="">
      <specs:enumUnit>
        <specs:enumItemsType>boolean</specs:enumItemsType>
        <specs:enumItems>
          <specs:enumItem></specs:enumItem>
          <specs:enumItem></specs:enumItem>
        </specs:enumItems>
      </specs:enumUnit>
    </specs:unit>
    <specs:scale>
      <specs:Quantitative>Ratio</specs:Quantitative>
    </specs:scale>
    <specs:expression>The existence of
    certification.</specs:expression>
    <specs:definition>This metric ensures the EU
    confidentiality of the stored data. Confidentiality is
    enforced with end-2-end encryption provided by the Client
    component. We guarantee that the Client component is
    audited and thus (used as is) grants security of
    encryption.</specs:definition>
    <specs:note></specs:note>
  </specs:MetricDefinition>
</specs:Metric>

<specs:Metric
  name="Secure delegated Access"
  referenceId="secure_delegated_access">
  <specs:MetricDefinition>
    <specs:unit name="">
      <specs:enumUnit>
        <specs:enumItemsType>boolean</specs:enumItemsType>
        <specs:enumItems>
          <specs:enumItem>yes</specs:enumItem>
          <specs:enumItem>no</specs:enumItem>
        </specs:enumItems>
      </specs:enumUnit>
    </specs:unit>
    <specs:scale>
      <specs:Quantitative>Ratio</specs:Quantitative>
    </specs:scale>
    <specs:expression>The installation of an OAuth
    server</specs:expression>
```

```

        <specs:definition>This metric ensures that an OAuth
        Server is configured to ensure authentication and
        authorization of users and secure delegated
        access to the users' resources to registered
        clients.</specs:definition>
        <specs:note></specs:note>
    </specs:MetricDefinition>
</specs:Metric>

<specs:Metric
    name="Access report generation frequency "
    referenceId="access_report_generation_freq">
    <specs:MetricDefinition>
        <specs:unit name="hours">
            <specs:intervalUnit>

                <specs:intervalItemsType>integer</specs:intervalItemsType>

                <specs:intervalItemStart>1</specs:intervalItemStart>

                <specs:intervalItemStop></specs:intervalItemStop>

                <specs:intervalItemStep></specs:intervalItemStep>
                </specs:intervalUnit>
            </specs:unit>
        <specs:scale>
            <specs:Quantitative>Ratio</specs:Quantitative>
        </specs:scale>
        <specs:expression>The frequency of report
        generation</specs:expression>
        <specs:definition>This metric sets the frequency of access
        reports generation.For example, for
        access_report_gen_frequency=12, SPECS ensures that a
        report is generated at least once every 12
        hours.</specs:definition>
        <specs:note></specs:note>
    </specs:MetricDefinition>
</specs:Metric>

<specs:Metric
    name="AAA Log Completeness "
    referenceId="aaa_log_completeness">
    <specs:MetricDefinition>
        <specs:enumUnit>
            <specs:enumItemsType>string</specs:enumItemsType>
            <specs:enumItems>
                <specs:enumItem>LOW</specs:enumItem>
                <specs:enumItem>MEDIUM</specs:enumItem>

                <specs:enumItem>HIGH</specs:enumItem>
            </specs:enumItems>
        </specs:enumUnit>
        </specs:unit>
        <specs:scale>
            <specs:Quantitative>Ratio</specs:Quantitative>
        </specs:scale>
        <specs:expression>The level of completeness of
        report</specs:expression>
        <specs:definition>This metric represents how detailed the
        access reports must be.</specs:definition>
        <specs:note></specs:note>
    </specs:MetricDefinition>
</specs:Metric>
</specs:security_metrics>
</specs:serviceDescription>
</wsag:ServiceDescriptionTerm>

<wsag:ServiceProperties

```

```
wsag:Name="//specs:capability[@id='DBB']"
wsag:ServiceName="SecureStorageAAA">
<wsag:VariableSet>
  <wsag:Variable
    wsag:Name="specs_DBB_M17"
    wsag:Metric="write_serializability_M17">
    <wsag:Location>
      //specs:CCMsecurityControl[@id='IVS-02'] |
      //specs:CCMsecurityControl[@id='BCR_01'] |
      //specs:CCMsecurityControl[@id='BCR_11']
    </wsag:Location>
  </wsag:Variable>

  <wsag:Variable
    wsag:Name="specs_DBB_M18"
    wsag:Metric="read_freshness_M18">
    <wsag:Location>
      //specs:CCMsecurityControl[@id='AIS-03'] |
      //specs:CCMsecurityControl[@id='BCR_01'] |
      //specs:CCMsecurityControl[@id='BCR_11']
    </wsag:Location>
  </wsag:Variable>

  <wsag:Variable
    wsag:Name="specs_DBB_M25"
    wsag:Metric="integrity_M25">
    <wsag:Location>
      //specs:CCMsecurityControl[@id='AIS-03']
    </wsag:Location>
  </wsag:Variable>
</wsag:VariableSet>
</wsag:ServiceProperties>

<wsag:ServiceProperties
  wsag:Name="//specs:capability[@id='E2EE']"
  wsag:ServiceName="SecureStorageAAA">
  <wsag:VariableSet>
    <wsag:Variable
      wsag:Name="specs_e2ee_M19"
      wsag:Metric="confidentiality_M19">
      <wsag:Location>
        //specs:CCMsecurityControl[@ccm:id='EKM-01'] |
        //specs:CCMsecurityControl[@ccm:id='EKM-03']
      </wsag:Location>
    </wsag:Variable>
  </wsag:VariableSet>
</wsag:ServiceProperties>

  <wsag:ServiceProperties
  wsag:Name="//specs:capability[@id='AAA']"
  wsag:ServiceName="SecureStorageAAA">
  <wsag:VariableSet>
    <wsag:Variable
      wsag:Name="specs_aaa_m1"
      wsag:Metric="secure_delegated_access">
      <wsag:Location>
        //specs:CCMsecurityControl[@ccm:id='IAM-02'] |
        //specs:CCMsecurityControl[@ccm:id='IAM-04']
      </wsag:Location>
    </wsag:Variable>
  </wsag:VariableSet>

    <wsag:VariableSet>
    <wsag:Variable
      wsag:Name="specs_aaa_m2"
      wsag:Metric="access_report_generation_freq">
      <wsag:Location>
        //specs:CCMsecurityControl[@ccm:id='IAM-02'] |
```

```

        //specs:CCMsecurityControl[@ccm:id='IAM-04']
    </wsag:Location>
</wsag:Variable>
</wsag:VariableSet>

    <wsag:VariableSet>
    <wsag:Variable
        wsag:Name="specs_aaa_m3"
        wsag:Metric="aaa_log_completeness">
    <wsag:Location>
        //specs:CCMsecurityControl[@ccm:id='IAM-02'] |
        //specs:CCMsecurityControl[@ccm:id='IAM-04']
    </wsag:Location>
    </wsag:Variable>
</wsag:VariableSet>

</wsag:ServiceProperties>

<wsag:GuaranteeTerm
    wsag:Name="//specs:capability[@id='DBB']"
    wsag:Obligated="ServiceProvider">
    <wsag:ServiceScope wsag:ServiceName="SecureStorageAAA"/>
    <wsag:QualifyingCondition>false</wsag:QualifyingCondition>
    <wsag:ServiceLevelObjective>
    <wsag:CustomServiceLevel>
    <specs:objectiveList>
    <specs:SLO SLO_ID="DBB_slo1">
    <specs:MetricREF>specs_DBB_M17</specs:MetricREF>
    <specs:SLOexpression>
    <specs:oneOpExpression>
    <specs:operator>eq</specs:operator>
    <specs:operand>yes</specs:operand>
    </specs:oneOpExpression>
    </specs:SLOexpression>
    <specs:importance_weight>MEDIUM</specs:importance_weight>
    </specs:SLO>

    <specs:SLO SLO_ID="DBB_slo2">
    <specs:MetricREF>specs_DBB_M18</specs:MetricREF>
    <specs:SLOexpression>
    <specs:oneOpExpression>
    <specs:operator>eq</specs:operator>
    <specs:operand>yes</specs:operand>
    </specs:oneOpExpression>
    </specs:SLOexpression>
    <specs:importance_weight>MEDIUM</specs:importance_weight>
    </specs:SLO>

    <specs:SLO SLO_ID="DBB_slo3">
    <specs:MetricREF>specs_DBB_M25</specs:MetricREF>
    <specs:SLOexpression>
    <specs:oneOpExpression>
    <specs:operator>eq</specs:operator>
    <specs:operand>yes</specs:operand>
    </specs:oneOpExpression>
    </specs:SLOexpression>
    <specs:importance_weight>MEDIUM</specs:importance_weight>
    </specs:SLO>
    </specs:objectiveList>
    </wsag:CustomServiceLevel>
    </wsag:ServiceLevelObjective>
    <wsag:BusinessValueList></wsag:BusinessValueList>
</wsag:GuaranteeTerm>

<wsag:GuaranteeTerm
    wsag:Name="//specs:capability[@id='E2EE']"
    wsag:Obligated="ServiceProvider">
    <wsag:ServiceScope wsag:ServiceName="SecureStorageAAA"/>

```

```

<wsag:QualifyingCondition>false</wsag:QualifyingCondition>
<wsag:ServiceLevelObjective>
  <wsag:CustomServiceLevel>
    <specs:objectiveList>
      <specs:SLO SLO_ID="e2ee slo1">
        <specs:MetricREF>specs_e2ee_M19</specs:MetricREF>
        <specs:SLOexpression>
          <specs:oneOpExpression>
            <specs:operator>eq</specs:operator>
            <specs:operand>yes</specs:operand>
          </specs:oneOpExpression>
        </specs:SLOexpression>
        <specs:importance_weight>MEDIUM</specs:importance_weight>
      </specs:SLO>
    </specs:objectiveList>
  </wsag:CustomServiceLevel>
</wsag:ServiceLevelObjective>
<wsag:BusinessValueList></wsag:BusinessValueList>
</wsag:GuaranteeTerm>

  <wsag:GuaranteeTerm
wsag:Name="//specs:capability[@id='AAA']"
wsag:Obligated="ServiceProvider">
  <wsag:ServiceScope wsag:ServiceName="SecureStorageAAA"/>
  <wsag:QualifyingCondition>false</wsag:QualifyingCondition>
  <wsag:ServiceLevelObjective>
    <wsag:CustomServiceLevel>
      <specs:objectiveList>
        <specs:SLO SLO_ID="aaa slo1">
          <specs:MetricREF>specs_aaa_m1</specs:MetricREF>
          <specs:SLOexpression>
            <specs:oneOpExpression>
              <specs:operator>eq</specs:operator>
              <specs:operand>yes</specs:operand>
            </specs:oneOpExpression>
          </specs:SLOexpression>
          <specs:importance_weight>MEDIUM</specs:importance_weight>
        </specs:SLO>
        <specs:SLO SLO_ID="aaa slo2">
          <specs:MetricREF>specs_aaa_m2</specs:MetricREF>
          <specs:SLOexpression>
            <specs:oneOpExpression>
              <specs:operator>leq</specs:operator>
              <specs:operand>24</specs:operand>
            </specs:oneOpExpression>
          </specs:SLOexpression>
          <specs:importance_weight>MEDIUM</specs:importance_weight>
        </specs:SLO>
        <specs:SLO SLO_ID="aaa slo3">
          <specs:MetricREF>specs_aaa_m3</specs:MetricREF>
          <specs:SLOexpression>
            <specs:oneOpExpression>
              <specs:operator>eq</specs:operator>
              <specs:operand>MEDIUM</specs:operand>
            </specs:oneOpExpression>
          </specs:SLOexpression>
          <specs:importance_weight>MEDIUM</specs:importance_weight>
        </specs:SLO>
      </specs:objectiveList>
    </wsag:CustomServiceLevel>
  </wsag:ServiceLevelObjective>
  <wsag:BusinessValueList></wsag:BusinessValueList>
</wsag:GuaranteeTerm>
</wsag:All>
</wsag:Terms>
</wsag:AgreementOffer>

```