



*Secure Provisioning of Cloud Services
based on SLA Management*

SPECS Project - Deliverable 2.3.3

Annex B – Negotiation API

Version no. 1.1
30 April 2016



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

Table of contents

Table of contents	2
Index of tables.....	3
1. Negotiation API Resources	4
2. Negotiation API Calls.....	5
3. Negotiation API Data Models	15
4.1. Service Description Term data model.....	15
4.2. Service Level Objective data model	16
4.3. SLA Template data model.....	18
References.....	20

Index of tables

Table 1. Negotiation API Resources..... 4

Table 2. *Service Description Term* data model.....16

Table 3. *Service Level Objective* data model18

Table 4. *SLA Template* data model.....19

In this annex, we illustrate the Negotiation API provided by the Negotiation module¹. In particular, the Negotiation API is offered by the SLO Manager component, which is the only one accessible from the outside. The interfaces of the components that are accessible only internally are not presented in this deliverable.

1. Negotiation API Resources

The Negotiation API supports the resources described in Table 1. Negotiation API Resources with the following structure, adopted in all annexes:

- The *Resource* column simply reports a significant name for the resources.
- The *Type* column describes the type of resource: allowed types are collections (managed according to the guidelines) or objects.
- The *Description* column offers a brief description of the resource, while the *Base URI* column reports the URI identifier used to access the resource.

Resource	Type	Description	Base URI
Service Description Terms	Collection	A list of available Service Description Terms (SDTs)	/sla-negotiation/sdts
Service Description Term	Object	An SDT associated to a certain identifier.	/sla-negotiation/sdts/{id}
Service Level Objectives	Collection	A list of service level objectives (SLOs) related to a capability, or to a list of capabilities.	/sla-negotiation/slos
Service Level Objective	Object	A service level objective (SLO) related to an id.	/sla-negotiation/slos/{id}
SLA Templates	Collection	A list of SLA templates WSAG compliant.	/sla-negotiation/sla-templates/{id}
SLA Template	Object	An SLA Template WSAG compliant associated to a certain identifier.	/sla-negotiation/sla-templates/{id}
SLA Offers	Collection	A list of SLA Offers based on a SLATemplate.	/sla-negotiation/sla-templates/{template_id}/slaoffers/
SLA Offer	Object	The SLA Offer associated to a certain identifier.	/sla-negotiation/sla-templates/{template_id}/slaoffers/{id}/

Table 1. Negotiation API Resources

¹ This annex updates Annex D of deliverable D1.3.

2. Negotiation API Calls

In the following tables, we present the set of REST calls associated to the Negotiation API. For each call, we report the resource URI (i.e., the call path) and the applicable HTTP methods, along with other relevant information (including the request and response bodies and the semantics of response codes).

Resource URI	/sla-negotiation/sdts/	
GET	<i>Description</i>	It retrieves all the SDTs available in the repository.
	<i>Query string</i>	Not supported
	<i>Request body</i>	Empty
	<i>Response body</i>	Media Type: ✓ application/xml http://www.specs-project.eu/resources/schemas/xml/sdt.xsd (SDT XML description)
	<i>Response code semantics</i>	✓ <i>200 Created</i> : the available collection of SDT resources has been successfully returned.
	<i>Notes</i>	
POST	<i>Description</i>	It creates an SDT.
	<i>Query string</i>	Not supported
	<i>Request body</i>	Media Type: ✓ application/xml http://www.specs-project.eu/resources/schemas/xml/sdt.xsd (SDT XML description)
	<i>Response body</i>	Media Type: ✓ text/plain Resource URI of the created SDT Resource.
	<i>Response code semantics</i>	✓ <i>201 Created</i> : the SDT resource has been correctly created. ✓ <i>435 Invalid Input</i> : request body not compliant with the defined schema.
	<i>Notes</i>	

Resource URI	/sla-negotiation/sdts/{id}	
GET	<i>Description</i>	It returns an SDT with the associated identifier id.
	<i>Query string</i>	Not supported
	<i>Request body</i>	Empty
	<i>Response body</i>	Media Type: ✓ Application/xml http://www.specs-project.eu/resources/schemas/xml/sdt.xsd (SDT XML description)
	<i>Response code semantics</i>	✓ 200 OK: The SDT with identifier id has successfully been returned. ✓ 404 Not Found: The SDT with the requested id was not found.
	<i>Notes</i>	
DELETE	<i>Description</i>	It deletes the SDT with the associated identifier id.
	<i>Query string</i>	Not supported
	<i>Request body</i>	Empty
	<i>Response body</i>	Empty
	<i>Response code semantics</i>	✓ 204 No content: The SDT with identifier id has successfully been deleted. ✓ 404 Not Found: The SDT with the requested id was not found.
	<i>Notes</i>	
PUT	<i>Description</i>	It updates the SDT with the associated id.
	<i>Query string</i>	Not supported

<i>Request body</i>	Media Type: ✓ application/xml http://www.specs-project.eu/resources/schemas/xml/sdt.xsd (SDT XML description)
<i>Response body</i>	Empty
<i>Response code semantics</i>	• 200 OK: The SDT with identifier id has successfully been updated. • 409 Conflict: Request body not compliant with the defined schema.
<i>Notes</i>	

Resource URI	/sla-negotiation/slos	
GET	<i>Description</i>	It returns a list of SLOs related to a capability specified in the query string parameter. The “items”, “page” and “length” query parameters are optional and can be used to narrow the number of items to return.
	<i>Query string</i>	• capability_id • items={num_items} • page={first_item}&length={len}
	<i>Request body</i>	Empty
	<i>Response body</i>	Media type: ✓ application/xml http://www.specs-project.eu/resources/schemas/xml/collections.xsd (Collection XML description) ✓ application/json http://www.specs-project.eu/resources/schemas/json/collections.json (Collection JSON description)
	<i>Response code semantics</i>	200 OK: The list of SLOs has been returned. 435 Invalid Input: Request body not compliant with the defined schema.
	<i>Notes</i>	The query string can be a list of capabilities separated by comma. Example: <code>/sla-negotiation/slos?capability_id="TLS,WebPool"</code>

POST	<i>Description</i>	Creates a new SLO.
	<i>Query string</i>	Not supported
	<i>Request body</i>	Media Type: ✓ application/xml ○ http://www.specs-project.eu/resources/schemas/xml/slo.xsd ○ (SLO XML description)
	<i>Response body</i>	Media Type: ✓ text/plain Resource URI of the created SLO.
	<i>Response code semantics</i>	✓ <i>201 Created</i>: The SLOs has been successfully created. ✓ <i>435 Invalid Input</i>: request body not compliant with the defined schema.
	<i>Notes</i>	

Resource URI	/sla-negotiation/slos/{id}	
GET	<i>Description</i>	It returns the SLO with the associated id.
	<i>Query string</i>	Not supported
	<i>Request body</i>	Empty
	<i>Response body</i>	Media Type: ✓ application/xml ○ http://www.specs-project.eu/resources/schemas/xml/slo.xsd ○ (SLO XML description)
	<i>Response code semantics</i>	✓ <i>200 OK</i>: The SLOs associated with the identifier id has successfully been returned. ✓ <i>404 Not Found</i>: The SLOs with the requested id was not found.
	<i>Notes</i>	
DELETE	<i>Description</i>	It deletes the SLO with the associated identifier id.
	<i>Query string</i>	Not supported

	<i>Request body</i>	Empty
	<i>Response body</i>	Empty
	<i>Response code semantics</i>	✓ <i>204 No content</i>: The SLO with identifier id has successfully been deleted. ✓ <i>404 Not Found</i>: The SLO with the requested id was not found.
	<i>Notes</i>	
PUT	<i>Description</i>	It updates the SLO with the associated id.
	<i>Query string</i>	Not supported
	<i>Request body</i>	Media Type: ✓ application/xml http://www.specs-project.eu/resources/schemas/xml/slo.xsd (SLO XML description)
	<i>Response body</i>	Empty
	<i>Response code semantics</i>	<i>200 OK</i>: The SLO with identifier id has successfully been updated. <i>409 Conflict</i>: Request body not compliant with the defined schema.
	<i>Notes</i>	

Resource URI	/sla-negotiation/sla-templates	
GET	<i>Description</i>	It returns the available collection of SLA Templates if no query has been specified. Otherwise, the “items”, “page” and “length” query parameters are used to narrow the number of items to return.
	<i>Query string</i>	• Not present • items={num_items} • page={first_item}&length={len}
	<i>Request body</i>	Empty
	<i>Response</i>	Media type:

	<i>body</i>	✓ <code>application/xml</code> http://www.specs-project.eu/resources/schemas/xml/collections.xsd (Collection XML description) ✓ <code>Application/json</code> http://www.specs-project.eu/resources/schemas/json/collections.json (Collection JSON description)
	<i>Response code semantics</i>	• <i>200 OK</i>: The SLATemplate list has successfully been returned.
	<i>Notes</i>	
POST	<i>Description</i>	Add an SLATemplate, provided in the XML format, to the collection of templates.
	<i>Query string</i>	Not supported
	<i>Request body</i>	Media Type: • <code>application/xml</code> http://www.specs-project.eu/resources/schemas/xml/SLAtemplate.xsd (SLA Template XML description)
	<i>Response body</i>	Media Type: • <code>text/plain</code> Returns the URI of the created resource.
	<i>Response code semantics</i>	• <i>201 Created</i>: The SLA Template has been correctly created. • <i>435 Invalid Input</i>: Request body not compliant with the defined schema.
	<i>Notes</i>	

Resource URI	/sla-negotiation/sla-templates/{id}	
GET	<i>Description</i>	It returns the SLA Template with the identifier id.
	<i>Query string</i>	Not supported
	<i>Request body</i>	Empty

	<i>Response body</i>	Media Type: ✓ application/xml http://www.specs-project.eu/resources/schemas/xml/SLAtemplate.xsd (SLA Template XML description)
	<i>Response code semantics</i>	• <i>200 OK</i>: The SLA Template with identifier id has successfully been returned. • <i>404 Not Found</i>: The SLA Template with the requested id not found.
	<i>Notes</i>	
DELETE	<i>Description</i>	It deletes the SLA Template with the associated id.
	<i>Query string</i>	Not supported
	<i>Request body</i>	Empty
	<i>Response body</i>	Empty
	<i>Response code semantics</i>	• <i>204 No content</i>: The SLO with identifier id has successfully been deleted. • <i>404 Not Found</i>: The SLO with the requested id was not found.
	<i>Notes</i>	
PUT	<i>Description</i>	It updates the SLA Template with the associated id.
	<i>Query string</i>	Not supported
	<i>Request body</i>	Media Type: ✓ application/xml http://www.specs-project.eu/resources/schemas/xml/SLAtemplate.xsd (SLA Template XML description)
	<i>Response body</i>	Empty
	<i>Response code semantics</i>	✓ <i>200 OK</i>: The SLO with identifier id has successfully been updated. ✓ <i>409 Conflict</i>: Request body not compliant with the defined schema.
	<i>Notes</i>	

Resource URI	/sla-negotiation/sla-templates/{template_id}/slaoffers/	
GET	<i>Description</i>	It creates and returns the collection of SLAOffers created for the SLATemplate with id <i>template_id</i> , if no query string is specified. Otherwise, the “items”, “page” and “length” query parameters are used to narrow the number of items to return.
	<i>Query string</i>	Not present items={num_items} page={first_item}&length={len}
	<i>Request body</i>	Empty
	<i>Response body</i>	Media type: ✓ Application/xml http://www.specs-project.eu/resources/schemas/xml/collections.xsd (Collection XML description) ✓ Application/json http://www.specs-project.eu/resources/schemas/json/collections.json (Collection JSON description)
	<i>Response code semantics</i>	200 OK: The SLAOffers with the associated <i>template_id</i> has successfully been returned. 404 Not Found: No SLAOffers were created for this requested <i>template_id</i>.
	<i>Notes</i>	A SLATemplate is used to retrieve the information about the constraints. A call to the Supply Chain Manager is made to collect the Supply Chain(s) needed to implement the SLA(s). The SLAOffers will be accessed based on this template id.
POST	<i>Description</i>	It creates and returns the collection of SLAOffers created for the SLATemplate with id <i>template_id</i> .
	<i>Query string</i>	Not supported
	<i>Request body</i>	Empty
	<i>Response body</i>	Media Type: ✓ application/xml http://www.specs-project.eu/resources/schemas/xml/collections.xsd

	<i>Response code semantics</i>	✓ 200 OK: The Collection SLAOffer with the associated <i>template_id</i> has successfully been returned. ✓ 404 Not Found: No SLAOffer were created for this requested <i>template_id</i>.
	<i>Notes</i>	An SLATemplate is used to retrieve the information about the constraints. A call to the Supply Chain Manager is made to collect the Supply Chain(s) needed to implement the SLA(s). The SLAOffer will be accessed based on this template id.

Resource URI	/sla-negotiation/sla-templates/{template_id}/slaoffers/{id}	
GET	<i>Description</i>	It returns the SLAOffer with the associated id created for the template associated with <i>template_id</i> .
	<i>Query string</i>	Not supported
	<i>Request body</i>	Empty
	<i>Response body</i>	Media Type: ✓ application/xml http://www.specs-project.eu/resources/schemas/xml/SLATemplate.xsd (SLA Template XML description)
	<i>Response code semantics</i>	✓ 200 OK: The SLAOffer for the requested <i>template_id</i> and SLAOffer id have been returned. ✓ 404 Not Found: The resource could not be found.
	<i>Notes</i>	
Resource URI	/sla-negotiation/sla-templates/{template_id}/slaoffers/current	
PUT	<i>Description</i>	It updates the accepted SLAOffer for the template associated with <i>template_id</i> .
	<i>Query string</i>	Not supported
	<i>Request body</i>	Media Type: ✓ application/xml http://www.specs-project.eu/resources/schemas/xml/SLATemplate.xsd
	<i>Response body</i>	Empty

	<i>Response code semantics</i>	• <i>200 OK</i>: The SLAOffer for the requested template_id and SLAOffer id have been returned. ✓ <i>404 Not Found</i>: The resource could not be found.
	<i>Notes</i>	This method shall be used to accept an SLAOffer for a negotiated template.
DELETE	<i>Description</i>	It deletes the Offer with the associated id.
	<i>Query string</i>	Not supported
	<i>Request body</i>	Empty
	<i>Response body</i>	Empty
	<i>Response code semantics</i>	• <i>204 No content</i>: The Offer with identifier id has successfully been deleted. • <i>404 Not Found</i>: The Offer with the requested id was not found.
	<i>Notes</i>	

3. Negotiation API Data Models

This section provides the data models adopted by the Negotiation API and specified in the previous section as part of the documentation of each Negotiation API call. Apart from the data model of collections described in Section 5.4 of deliverable D1.3, the Negotiation API calls rely upon the following data models:

- SDT
- SLO
- SLA Template.

4.1. Service Description Term data model

The Service Description Term (SDT) data model represents a description of the activity related to building a valid SDT according to the provided identifier. Since an SDT is part of an SLA, written in XML, it is only provided the XML format for SDT data model in Table 2. The schema is also available at [\[xml 1\]](#).

Format	Data model
XML	<pre> <?xml version="1.0" encoding="UTF-8"?> <!-- This schema defines all the elements of the SLA Template that are related to service description terms Elements defined in this file are part of the main SPECS target namespace: http://www.specs-project.eu/resources/schemas/xml/SLAtemplate whose alias is "specs". This file is INCLUDED in SLAtemplate.xsd and INCLUDES the schema for security capabilities defined in capability.xsd and the schema for security metrics defined in security_metric.xsd --> <xs:schema targetNamespace="http://www.specs- project.eu/resources/schemas/xml/SLAtemplate" xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:specs="http://www.specs-project.eu/resources/schemas/xml/SLAtemplate" elementFormDefault="qualified"> <!-- includes capability.xsd, where other elements belonging to the "specs" namespace are defined --> <xs:include schemaLocation="capability.xsd"/> <!-- includes security_metric.xsd, where other elements belonging to the "specs" namespace are defined --> <xs:include schemaLocation="security_metric.xsd"/> <!-- ***** serviceDescription: serviceResources, capabilities and metrics --> <xs:complexType name="serviceDescriptionType" > <xs:sequence> <!-- serviceResources --> <xs:element name="serviceResources" minOccurs="0" maxOccurs="unbounded"> <xs:complexType> <xs:sequence> <xs:element name="resourcesProvider" minOccurs="1" maxOccurs="unbounded" > <xs:complexType> <xs:sequence> <xs:element name="VM" minOccurs="0" maxOccurs="unbounded"> <xs:complexType> <xs:attribute name="appliance" type="xs:string"/> </pre>

```

        <xs:attribute name="hardware" type="xs:string"/>
        <xs:attribute name="description" type="xs:string"/>
      </xs:complexType>
    </xs:element>
  </xs:sequence>

  <xs:attribute name="id" type="xs:string" use="required"/>
  <xs:attribute name="name" type="xs:string" use="required"/>
  <xs:attribute name="zone" type="xs:string" use="required"/>
  <xs:attribute name="description" type="xs:string" use="required"/>
  <xs:attribute name="maxAllowedVMs" type="xs:integer"
use="optional"/><!-- set by each provider -->
  <xs:attribute name="minRequiredVMs" type="xs:integer"
use="optional"/><!-- filled in an offer -->
  <xs:attribute name="label" type="xs:string" use="required"/>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>

<!-- capabilities -->
<xs:element name="capabilities" minOccurs="1" maxOccurs="1">
  <xs:complexType>
    <xs:sequence>
      <!-- defined in capability.xsd -->
      <xs:element name="capability" type="specs:capabilityType" minOccurs="1"
maxOccurs="unbounded" />
    </xs:sequence>
  </xs:complexType>
</xs:element>

<!-- metrics -->
<xs:element name="security_metrics" minOccurs="0" maxOccurs="1">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Metric" type="specs:MetricType" minOccurs="1"
maxOccurs="unbounded" />
    </xs:sequence>
  </xs:complexType>
</xs:element>

</xs:sequence>
</xs:complexType>
<!-- end of serviceDescription -->

</xs:schema>

```

Table 2. Service Description Term data model

4.2. Service Level Objective data model

The Service Level Objective (SLO) data model represents a description of the activity related to building a valid SLO according to the provided identifier. Since an SLO is part of an SLA document, which is written in XML, we only provide the XML format for SLO data model in Table 3. *Service Level Objective* data model. The schema is also available at [\[xml 2\]](#).

Format	Data model
XML	<pre> <?xml version="1.0" encoding="UTF-8"?> <!-- This schema defines all the elements of the SLA Template that are related to security SLOs. </pre>

```
Elements defined in this file are part of the main
SPECS target namespace:
http://www.specs-project.eu/resources/schemas/xml/SLAtemplate
whose alias is "specs".
This file is INCLUDED in SLAtemplate.xsd -->

<xs:schema
  targetNamespace="http://www.specs-
project.eu/resources/schemas/xml/SLAtemplate"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"

  xmlns:specs="http://www.specs-project.eu/resources/schemas/xml/SLAtemplate"
  elementFormDefault="qualified">

  <!-- SLO type -->
  <xs:complexType name="SLOType">
    <xs:sequence>
      <xs:element name="MetricREF" type="xs:string" minOccurs="1" maxOccurs="1"
/><!-- reference to metric definition -->
      <xs:element name="SLOexpression" type="specs:SLOexpressionType"
minOccurs="1" maxOccurs="1" />
      <xs:element name="importance_weight" minOccurs="1" maxOccurs="1" >
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:enumeration value="LOW"/>
            <xs:enumeration value="MEDIUM"/>
            <xs:enumeration value="HIGH"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
    </xs:sequence>
    <xs:attribute name="SLO_ID" type="xs:string" use="required"/>
  </xs:complexType>

  <xs:complexType name="SLOexpressionType">
    <xs:choice minOccurs="1" maxOccurs="1">
      <xs:element name="oneOpExpression">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="operator" type="specs:oneOpOperator" minOccurs="1"
maxOccurs="1"/>
            <xs:element name="operand" type="xs:anySimpleType" minOccurs="1"
maxOccurs="1"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="twoOpExpression">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="operator" type="specs:twoOpOperator" minOccurs="1"
maxOccurs="1"/>
            <xs:element name="operand1" type="xs:anySimpleType" minOccurs="1"
maxOccurs="1"/>
            <xs:element name="operand2" type="xs:anySimpleType" minOccurs="1"
maxOccurs="1"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:choice>
  </xs:complexType>

  <xs:simpleType name="oneOpOperator">
    <xs:restriction base="xs:string">
      <xs:enumeration value="eq"/>
      <xs:enumeration value="gt"/>
      <xs:enumeration value="lt"/>
      <xs:enumeration value="geq"/>
    </xs:restriction>
  </xs:simpleType>
```

	<pre> <xs:enumeration value="leq"/> </xs:restriction> </xs:simpleType> <xs:simpleType name="twoOpOperator"> <xs:restriction base="xs:string"> <xs:enumeration value="in excluded"/> <!--boundaries of interval excluded --> </xs:restriction> </xs:simpleType> <xs:simpleType name="weightType"> <xs:restriction base="xs:string"> <xs:enumeration value="LOW"/> <xs:enumeration value="MEDIUM"/> <xs:enumeration value="HIGH"/> </xs:restriction> </xs:simpleType> </xs:schema> </pre>
--	--

Table 3. Service Level Objective data model

4.3. SLA Template data model

The *SLA Template* data model is discussed in details in D2.2.2. It extends the WSAG standard by introducing security-related attributes including security capabilities, security metrics and security SLOs, in addition to defining a custom representation of provided services and available resources. Table 4 reports the XML schema, which refers to many of the elements defined in previously discussed schemas (SDT and SLO schemas) and which is available at [xml 3]. The complete SPECS SLA XML Framework is available online².

Format	Data model
XML	<pre> <?xml version="1.0" encoding="UTF-8"?> <!-- This schema defines the elements of the SLA template and extends the WSAG schema. Elements defined in this schema belong to the SPECS target namespace http://www.specs-project.eu/resources/schemas/xml/SLAtemplate whose alias is "specs". This schema includes two sub-schemas: sdt.xsd and slo.sdt. They define other elements of the "specs" namespace --> <xs:schema xmlns="http://www.w3.org/2001/XMLSchema" xmlns:specs="http://www.specs-project.eu/resources/schemas/xml/SLAtemplate" targetNamespace="http://www.specs- project.eu/resources/schemas/xml/SLAtemplate" xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"> <!-- includes sdt.xsd, where other elements belonging to the "specs" namespace are defined --> <xs:include schemaLocation="sdt.xsd"/> <!-- includes slo.xsd, where other elements belonging to the "specs" namespace are defined --> <xs:include schemaLocation="slo.xsd"/> </pre>

² <https://bitbucket.org/specs-team/specs-utility-xml-sla-framework>

```
<!-- defined in sdt.xsd -->
<xs:element name="serviceDescription" type="specs:serviceDescriptionType"/>

<!-- endpoint (it is to be put in <wsag:ServiceReference>) -->
<xs:element name="endpoint" type="xs:string" />

<xs:element name="objectiveList">
  <xs:complexType>
    <xs:sequence>
      <!-- defined in slo.xsd -->
      <xs:element name="SLO" type="specs:SLOType" minOccurs="1"
maxOccurs="unbounded" />
    </xs:sequence>
  </xs:complexType>
</xs:element>

</xs:schema>
```

Table 4. SLA Template data model

References

- xml_1 <http://www.specs-project.eu/resources/schemas/xml/sdt.xsd>.
- xml_2 <http://www.specs-project.eu/resources/schemas/xml/slo.xsd>.
- xml_3 <http://www.specs-project.eu/resources/schemas/xml/SLAtemplate.xsd>.