EUROPEAN COMMISSION

DIGIT
Connecting Europe Facility

Service Metadata Publishing (SMP) Version 0.12 - Interface Control Document

Service Offering Description

Date: 28/04/2016

Document Approver(s):

Approver Name	Role
Adrien FERIAL	Architect

Document Reviewers:

Reviewer Name	Role
Adrien FERIAL	Architect
João CUNHA	Architect

Summary of Changes:

Version	Date	Created by	Short Description of Changes
0.1	12/01/2016	Yves ADAM	Initial version
0.2	22/01/2016	Yves ADAM	Additions and corrections based on discussion with Adrien Ferial
0.3	26/01/2016	Yves ADAM	Additions and corrections based on SMP/SML task force meeting
0.4	28/01/2016	Yves ADAM	Additions and corrections based on discussion with Adrien Ferial
0.5	29/01/2016	Yves ADAM	Additions and corrections based on discussion with Adrien Ferial
0.6	04/02/2016	Yves ADAM	Additions and corrections based on feedback from João Cunha
0.7	17/02/2016	Yves ADAM	More additions and corrections based on feedback from João Cunha
0.8	23/02/2016	Yves ADAM	Update deriving from task force discussions of Feb 17th
0.9	29/02/2016	Yves ADAM	Corrections based on discussion with Adrien Ferial
0.10	24/03/2016	Yves ADAM	Corrections based on feedback from João Cunha and discussions with Adrien Ferial
0.11	20/04/2016	Yves ADAM	Corrections based on feedback from João RODRIGUES FRADE
0.12	26/04/2016	Yves ADAM	More additions and corrections based on feedback from João Cunha

Table of Contents

1. INTRODUCTION

1.1. Background

The eDelivery building block helps public administrations to exchange electronic data and documents with other public administrations, businesses and citizens, in an interoperable, secure, reliable and trusted way. Through the use of this building block, every participant becomes a node in the network using standard transport protocols and security policies. eDelivery is based on a distributed model, allowing direct communication between participants without the need to set up bilateral channels.

1.2. Purpose of the Interface Control Document

This document will univocally define the participant's interface to the SMP component of the eDelivery building block as it will extend and evolve in sight of its usage in the framework of eHealth and its additional requirements.

This use case / interface control document will be used as reference for mutual understanding of eHealth requirements on the one hand and the future service delivered by CEF on the other hand.

1.3. Scope of the document

This document is a high-level functional definition of the services provided by the SMP. This document will be later extended with additional document that further detail the services with technical information intended for the development of eHealth client solutions implementation.

1.4. Audience

This document is intended to:

- The architect and development teams of CEF for committing on future service delivery of SMP
- The architects and functional analysts of the eHealth team for validating the intended service against their requirements.

1.5. Definitions

All the concepts used throughout this document have been defined in the following documents:

- [REF4]
- [REF5]
- [REF8]

1.6. References

#	Document	Contents outline
[REF1]	Introduction to the Connecting Europe Facility - eDelivery building block	Overview of eDelivery
[REF2]	Using HTTP Methods for RESTful Services	Short description of HTTP Methods for RESTful Services
[REF3]	CEN/BII specifications for business documents	The **CEN Business Interoperability Interfaces** on public procurement in Europe (CEN/BII) initiative, established as a workshop under CEN, was initially launched in May 2007. Its aim was to help achieve the Digital Single Market by fostering implementation of **e-procurement** and **e-invoicing** in Europe, and especially in the **European public sector**.
[REF4]	***OpenPEPPOL AISBL - Policy for use of Identifiers***	
[REF5]	***OASIS - Service Metadata Publishing (SMP) Version 1.0 - Committee Specification 01***	This document describes a protocol for publishing service metadata within a 4-corner network.
[REF6]	eSens Building Blocks - ABB - Capability Lookup - 1.6.0	Capability Lookup is a technical service to accommodate a dynamic and flexible interoperability community. A capability lookup can provide metadata about the communication partner's interoperability capabilities on all levels defined in the European Interoperability Framework.
[REF7]	***eSens Building Blocks - PR - SMP***	e-SENS will use the SMP (Simple Metadata Publisher) specification originally developed by PEPPOL and generalized and standardized by OASIS. The SMP specification usually complements the Location LookUp ABB.
[REF8]	PEPPOL Transport Infrastructure Service Metadata Publishing (SMP)	This document describes the REST (Representational State Transfer) interface for Service Metadata Publication within the Business Document Exchange Network (BUSDOX).
[REF9]	SML/SMP/eDelivery PKI Impact Assessment for the CEF eHealth DSI	Objectives: 1) Assess the impact of migrating the "Configuration Server" of epSOS to the "SML/SMP" architecture of the eDelivery DSI; 2) Assess the impact of migrating the trust model of epSOS to the eDelivery dedicated PKI; 3) Assess the impacts of the replacement of the VPN network with TESTA services from a technical viewpoint.
[REF10]	Business Document Exchange Network - Common Definitions,	This document contains the definitions and terms that are common between the Business

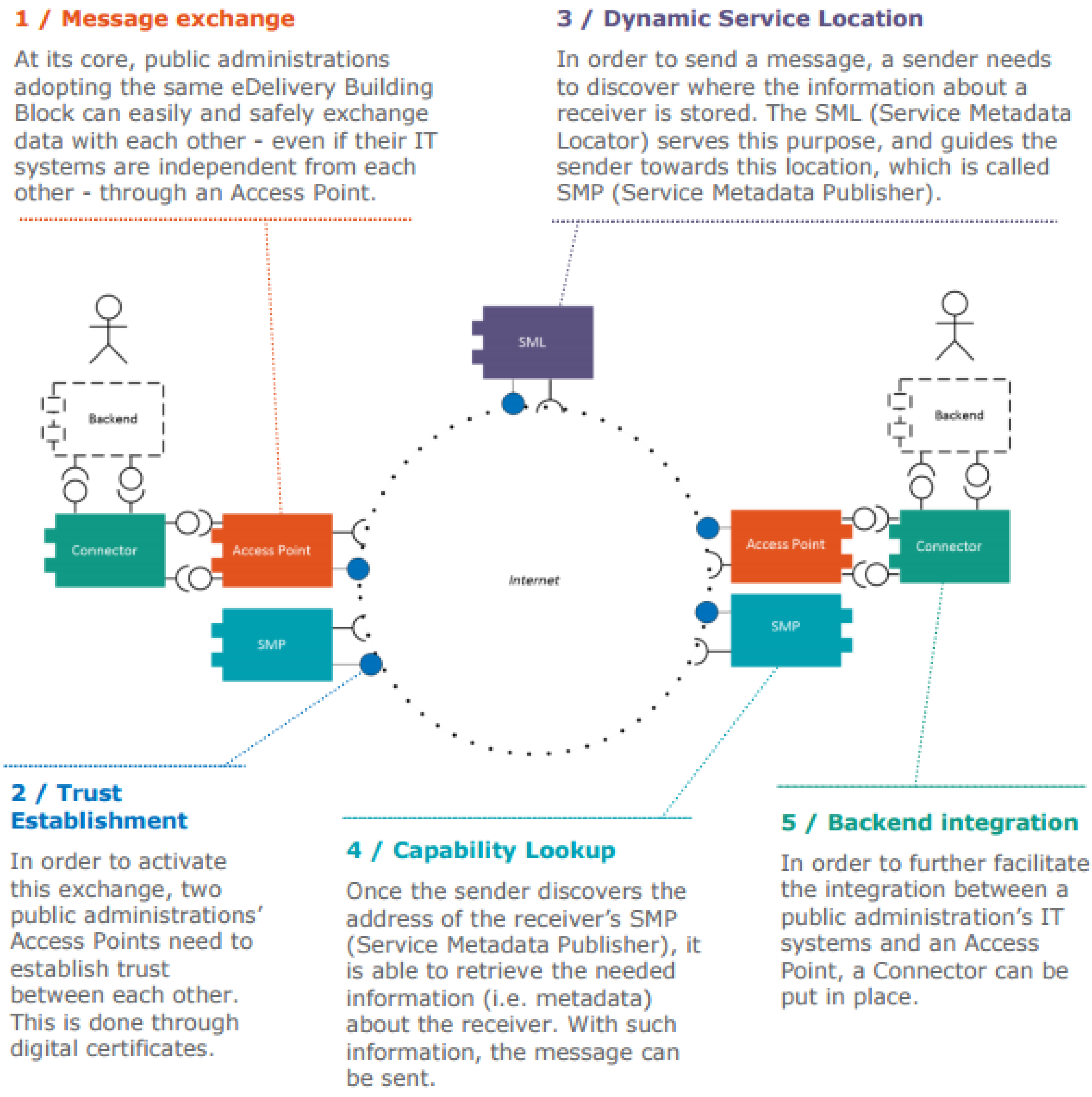
#	Document	Contents outline
	CommonDefinitions.pdf	Document Exchange Network (BUSDOX) service metadata and transport specifications. These are: 1° The START and LIME transport specifications; 2° The SML (Service Metadata Locator) and SMP (Service Metadata Publishing) specifications; 3° A scheme for process identifiers. This scheme is identified by the string —cenbii_procid_pia.
[REF11]	Change requests for the OASIS - Service Metadata Publishing (SMP) Version 1.0	DIGIT has identified a number of change requests that could, according to DIGIT, improve the robustness, the reusability and the genericity of the SMP standard. This document lists all these change requests.
[REF12]	Business Document Metadata Service Location - Software Architecture Document	This document is the Software Architecture document of the CIPA eDelivery Business Document Metadata Service Location application (BDMSL) sample implementation. It intends to provide detailed information about the project: 1) An overview of the solution 2) The different layers 3) The principles governing its software architecture.
[REF13]	PEPPOL Transport Infrastructure Service Metadata Locator (SML)	This document defines the profiles for the discovery and management interfaces for the Business Document Exchange Network (BUSDOX) Service Metadata Locator service.
[REF14]	Change requests for the OASIS Service Metadata Publishing (SMP) Version 1.0	As a result of multiple discussions with the different stakeholders, DIGIT has identified a number of change requests that could, according to DIGIT, improve the robustness, the reusability and the genericity of the SMP standard. This document lists all these change requests.

Important note : documents **listed in *bold italic red*** in the above list are to be considered for the detailed designed and the implementation of the SMP as this one must be fully compliant to those specifications.

2. INTERFACE DEFINITION

2.1. Positioning SMP in eDelivery

2.1.1. eDelivery in a nutshell



1 / Message exchange

At its core, public administrations adopting the same eDelivery Building Block can easily and safely exchange data with each other - even if their IT systems are independent from each other - through an Access Point.

3 / Dynamic Service Location

In order to send a message, a sender needs to discover where the information about a receiver is stored. The SML (Service Metadata Locator) serves this purpose, and guides the sender towards this location, which is called SMP (Service Metadata Publisher).

2 / Trust Establishment

In order to activate this exchange, two public administrations' Access Points need to establish trust between each other. This is done through digital certificates.

4 / Capability Lookup

Once the sender discovers the address of the receiver's SMP (Service Metadata Publisher), it is able to retrieve the needed information (i.e. metadata) about the receiver. With such information, the message can be sent.

5 / Backend integration

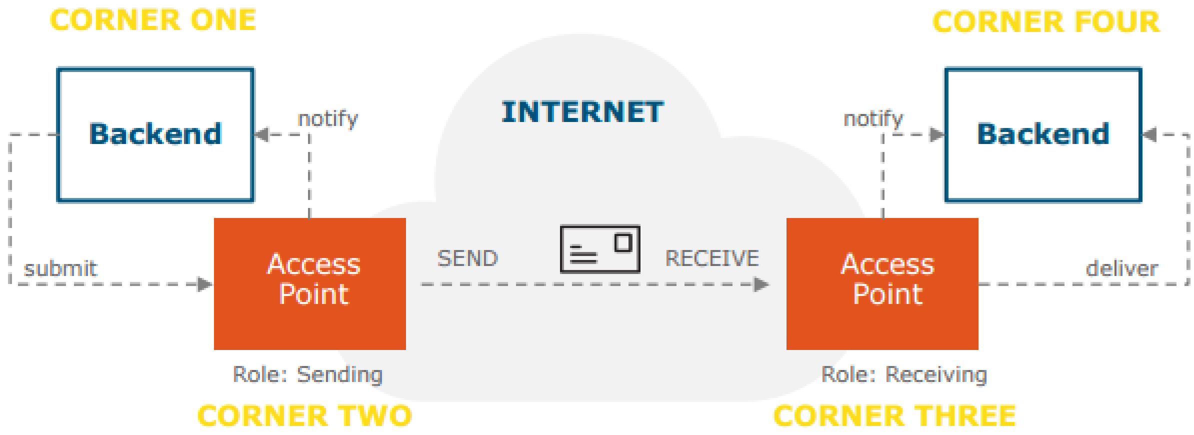
In order to further facilitate the integration between a public administration's IT systems and an Access Point, a Connector can be put in place.

The technical architecture of eDelivery is based on a conceptual model called **'four-corner model'.** This means that Backend systems (corners one and four) do not exchange messages directly with each other but via Access Points (corners two and three) that, in any given exchange, play the sender or receiver role.

The Access Points of eDelivery are not operated centrally, instead they are deployed in the Member States under the responsibility of a public or private sector service provider.

The users of the Access Points are the Backend systems that need to exchange information with other administrations or businesses across borders.

During the exchange, the data and documents are secured by eDelivery's trust establishment mechanisms. This implies a choice of trust establishment model.



The four-corner model

2.1.2. SMP role

The role of the SMP in the 'four corner model' is

- on the one hand to allow servers (*receivers*) to publish the definition of the services they provide; i.e. the documents they are able to receive and the means through which they can receive them, and on the other hand,
- to allow clients (*senders*) to find out the definitions of those services.

In that purpose, the SMP will provide services respectively to register services definitions (like "put metadata") by the receiver's administrator and to consult those definitions ("Retrieve Metadata") by the sender.

2.1.3. SMP / SML interactions

In order for the complete process to be consistent, the SMP must propagate some information to the SML:

- The location information of the SMP itself for allowing the senders to discover the SMP
- The location information of all access points providing access to the declared ServiceGroups the participants the SMP is managing.

In that purpose, the SML exposes several management services that allow the SMP to declare new location information or changes upon existing one. These management interfaces are introduced in [REF13], and are listed below:

- ***“Manage participant identifiers”*** interface. This is the interface for Service Metadata publishers for managing the metadata relating to specific participant identifiers that they make available.
- ***“Manage service metadata”*** interface. This is the interface for Service Metadata publishers for managing the metadata about their services, e.g. binding, interface profile and key information.

These interfaces will not be detailed here but the document will refer to these when they are invoked from the SMP REST services. Refer to the “*Execution*” sections of the REST Services definitions below for further details on these interactions.

In addition, the SML exposes the Service Metadata discovery interface. This is the lookup interface which enables senders to discover service metadata about specific target participants. As it is out of the scope of this document this service is not further discussed in the present document.

This functionality is currently not addressed but should be in a future release. The following use cases should then be foreseen:

- UC08 - Register SMP
- UC09 - Change SMP Location
- UC10 - Unregister SMP
- UC11 - Migrate Metadata SMP

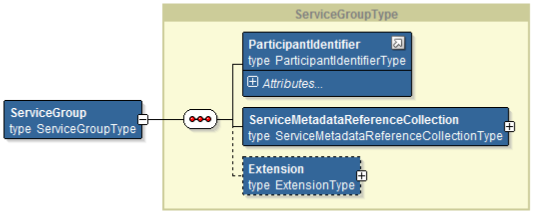
2.2. Data model

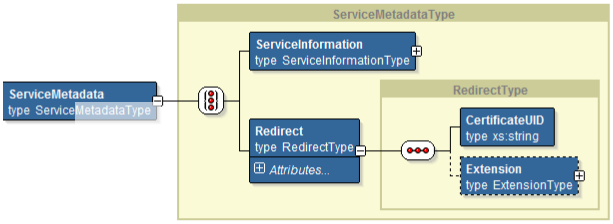
The SMP interface is built around the data it is intended to manage. Therefore, this documents starts by defining the data itself.

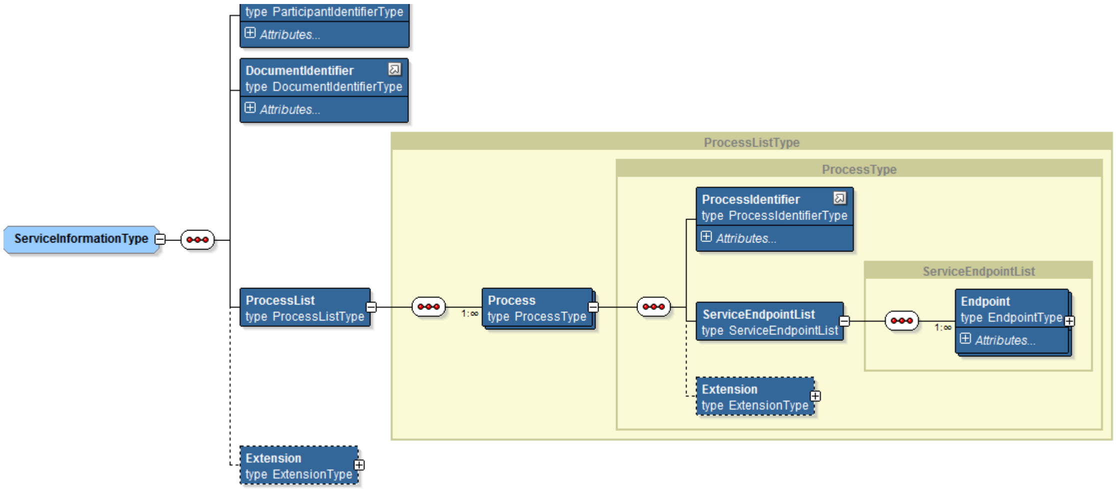
2.2.1. Logical data model

The diagram below depicts the major parts of the data model describing the configuration held by the SMP and managed through the interface described in this document. This model is another view of the XSD definition that can be found in annex 4.1 – "XSD files

Original official OASIS SMP XSD":







2.2.1.1. ServiceGroup

A service group is defined as "*structure that represents a set of services associated with a specific **Participant identifier** that is handled by a specific Service Metadata Publisher. The ServiceGroup structure holds a list of references to ServiceMetadata resources in the ServiceList structure*". (cf. [REF8], Data model)

Refer to [REF5] § 2.4 "Identifiers" for more details and additional references about identifiers of participants (/businesses), documents and processes.

2.2.1.2. ServiceMetadata

ServiceMetadata is defined as "*a structure that represents Metadata about a specific electronic service. The role of the ServiceMetadata structure is to associate a participant identifier with the ability to receive a specific document type over a specific transport […]*"

Refer to [REF5] § 2.4 "Identifiers" for more details and additional references about identifiers of participants (/businesses), documents and processes.

2.2.1.3. Process

As stated above, a ServiceMetadata is defined as "*a structure that represents Metadata about a specific electronic service. The role of the ServiceMetadata structure is to associate a participant identifier with the ability to receive a specific document type over a specific transport.*"

But…

"It also describes which business processes a document can participate […] " (cf. [REF8], "*Data model*")

… and it is the purpose of this intermediate entity (*Process*) to hold the process-related information (i.e. its identifier and scheme), and to allow a participant to use a document type to participate in multiple business processes (when applicable).

Refer to [REF5] § 2.4 "Identifiers" for more details and additional references about identifiers of participants (/businesses), documents and processes.

2.2.1.4. Endpoint

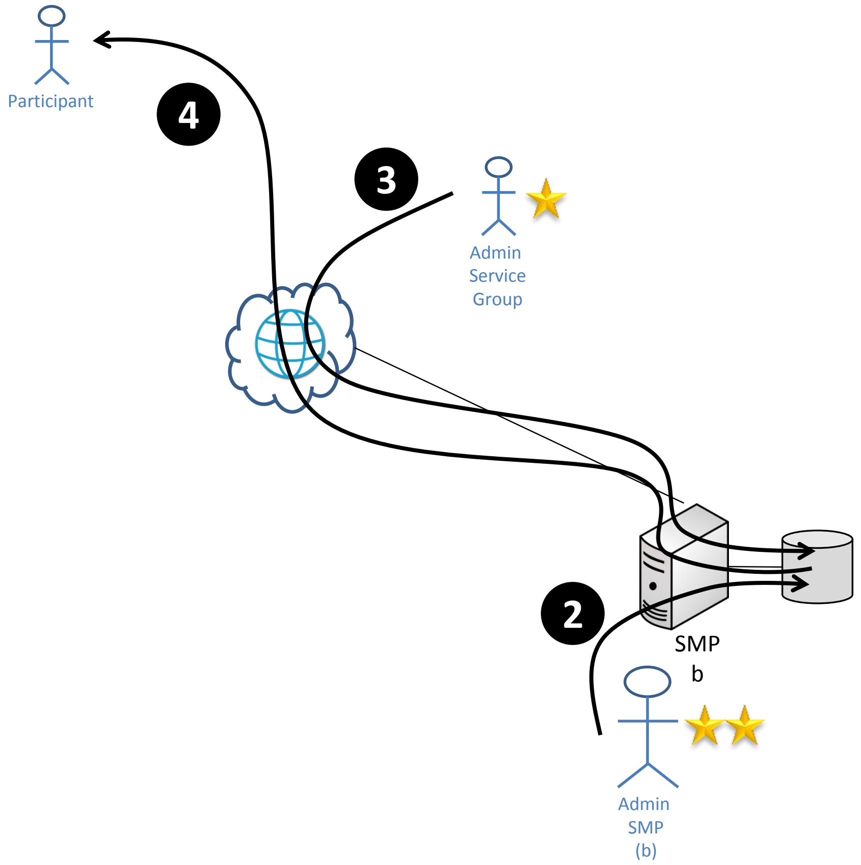
The endpoint is the ultimate entity, holding all the necessary information for all services of the ServiceGroup to be accessed by the sender in order to send document(s) to the receiver (cf. § 2.3.4.4 "Description of the individual fields (elements and attributes)" of [REF5])

XSD element	Description
endpointURI Element : /ServiceEndpointList/ Endpoint/EndpointURI	The address of an endpoint, as a URL
transportProfile Element : ServiceInformation/ ProcessList/../Endpoint/ @transportProfile	Indicates the type of transport method that is being used between access points
requireBusinessLevelSignature Element : ServiceInformation/ ProcessList/../Endpoint/ RequireBusinessLevelSignature	Set to "true" if the recipient requires business-level signatures for the message, meaning a signature applied to the business message before the message is put on the transport. This is independent of the transport-level signatures that a specific transport profile might mandate. This flag does not indicate which type of business-level signature might be required. Setting or consuming business-level signatures would typically be the responsibility of the final senders and receivers of messages, rather than a set of gateways.
minimumAuthenticationLevel Element : ServiceInformation/ ProcessList/../Endpoint/ MinimumAuthenticationLevel	Indicates the minimum authentication level that recipient requires. The specific semantics of this field is defined in a specific instance of a 4-corner infrastructure.
serviceActivationDate Element :	Activation date of the service. Senders should ignore services that are not yet activated. Format of ServiceActivationDate date is xs: dateTime.

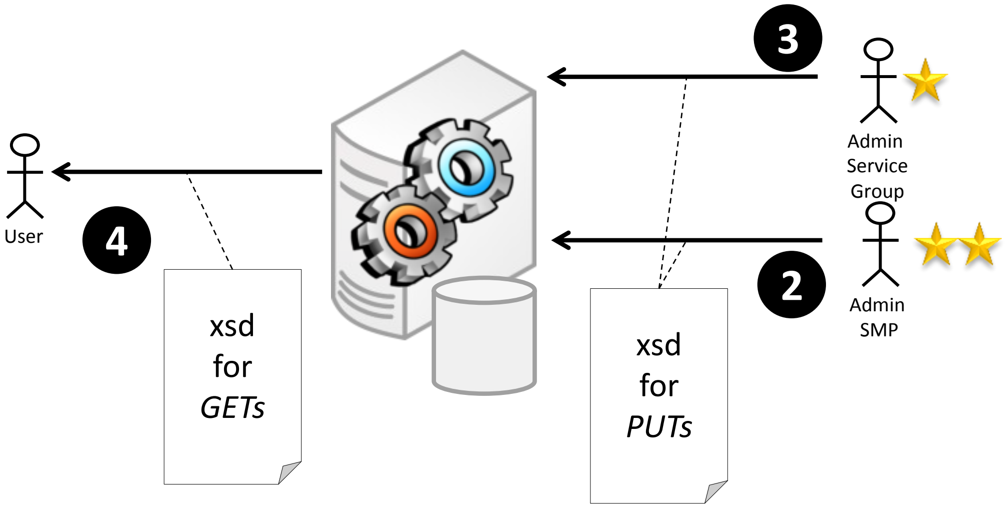
XSD element	Description
ServiceInformation/ ProcessList/../Endpoint/ ServiceActivationDate	
serviceExpirationDate Element : /ProcessList/../Endpoint/ ServiceExpirationDate	Expiration date of the service. Senders should ignore services that are expired. Format of ServiceExpirationDate date is xs:dateTime.
certificate Element : /ProcessList/../Endpoint/ Certificate	Holds the complete **[X509v3]** signing certificate of the recipient gateway, as a PEM base 64 encoded DER formatted value.
serviceDescription Element : /ProcessList/../Endpoint/ ServiceDescription	A human readable description of the service
technicalContactUrl Element : /ProcessList/../Endpoint/ TechnicalContactUrl	Represents a link to human readable contact information. This might also be an email address.
technicalInformationUrl Element : /ProcessList/../Endpoint/ TechnicalInformationUrl	A URL to human readable documentation of the service format. This could for example be a web site containing links to XML Schemas, WSDLs, Schematrons and other relevant resources.
extension Element : /Process/Extension	The extension element may contain any XML element. Clients MAY ignore this element. It can be used to add extension metadata to the process metadata block as a whole.
extension Element : /ServiceInformation/ Extension	The extension element may contain any XML element. Clients MAY ignore this element. It can be used to add extension metadata to the service metadata.

2.2.2. XSD files

Two XSDs are used to support the overall processes as defined in §2.6.1.1 - "Administration process":



1. The first one is the 'standard' one as published by OASIS which defines the interface for the retrieval of participant's information, adapted according to change requests (cf. §4.1.2 - Extended SMP XSD for standard services (GET)): ❹
2. The second one, defined in this document (cf. 4.1.3 – "Extended SMP XSD for ADMIN services (PUT)") defines the interface for administering the information about participant's: ❷ and ❸ .



2.2.2.1. OASIS' XSD for GET operations

The structure of this model is described as an XSD in the following document https://docs.oasis-open.org/bdxr/bdx-smp/v1.0/cs01/schemas/bdx-smp-201407.xsd and is included as annex 4.1 – “XSD files

Original official OASIS SMP XSD” to this document.

Please note that the XSD provided in annex 4.1.2 - “Extended SMP XSD”, is the one that is intended to be used in the future but it is not the original one from OASIS mentioned just above (see change details in annex).

This XSD structure describes the returned text for the ‘GET’ services (UC06 and UC07).

2.2.2.2. New XSD for PUT operations (Admin)

This XSD is extensively based on the previous one. It has been adapted to support the required input text for the admin services (UC02 to UC05). It also defines error codes to facilitate error management.

2.3. Use cases summary

2.3.1. Actors

Actor	Definition
System Admin ★★★	A user granted rights to administer the Admin SMP type of users. This role is symbolized by 3 stars (it has the highest authority)
Admin SMP ★★	A user granted rights to administer the participants (or ServiceGroups) This role is symbolized by 2 stars (it has the authority to create Admin ServiceGroups users)
Admin ServiceGroup ★	A user granted rights to administer the national access points (i.e. one or more ServiceGroups); i.e. to define the access points services metadata This role is symbolized by 1 single stars (it has the authority to define service groups, but not to create other users)
User	Any participant sending documents to any other receiver participant and consulting the SMP in that purpose This role is symbolized by no single star since he has only public read accesses

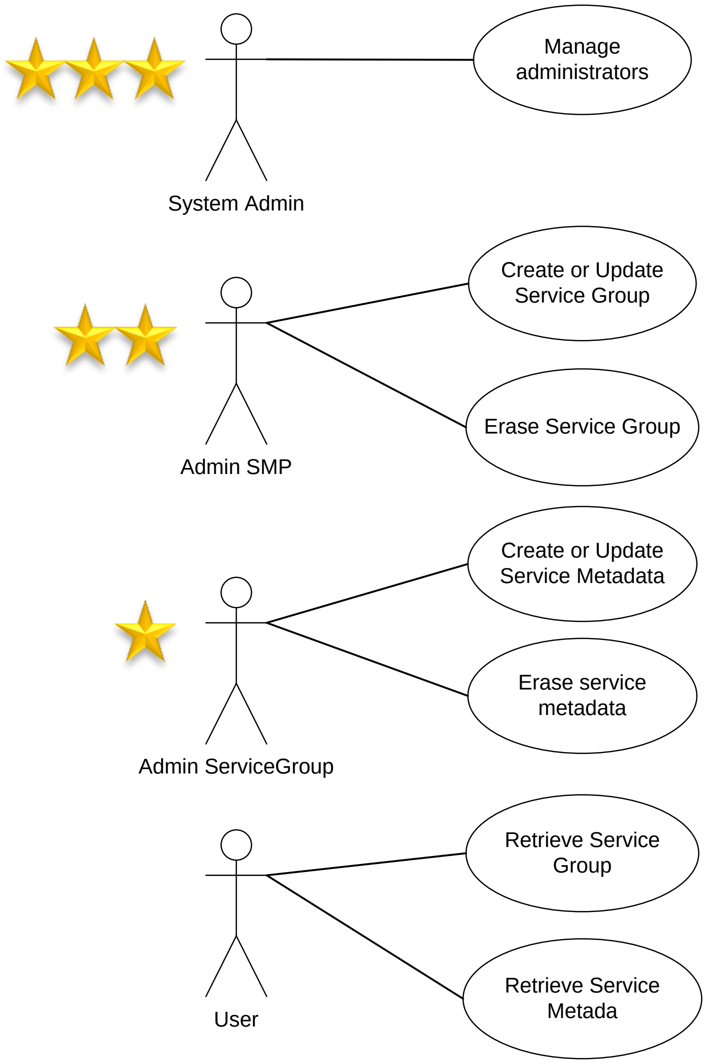
In addition to the role described above, the two additional terms will be used:

- ***Sender***: to refer to an actor who uses the system (the SMP) on the left hand side of the 'four corner model' introduced in 2.1.1 – "eDelivery in a nutshell". In the present use cases, the sender will only behave as a 'User' as described above in the roles list.

- ***Receiver***: to refer to an actor who uses the system (the SMP) on the right hand side of the same model. In the present use cases, the receiver will behave either as "Admin SMP" or "Admin ServiceGroup" roles.

The "System Admin" being neither on the left nor on the right of that model, but rather on top of it, he will never be referred to as 'sender' nor 'receiver'.

2.3.2. Use cases diagram



2.3.3. Use case list

ID	Actor	UC	Short description	Oper.	Data
UC01	System Admin	**Manage Administrators**	Create and modify user information in SMP table 'User'	n/a	User (table)
UC02	Admin SMP	**Create or Update Service Group**	Create a new ServiceGroup for a new receiver participant. This service stores the *Service Group* and links it to the specified duplet participantIdentifier + participantIndentifierScheme. Information is store into ServiceGroup table. This same service is used to create and update a ServiceGroup.	PUT	ServiceGroup
UC03	Admin SMP	**Erase Service Group**	Erases the service group definition AND the list of services and for the specified receiver participant.	DELETE	ServiceGroup
UC04	Admin ServiceGroup	**Create or Update Service Metadata**	Publish detailed information about one specific document service (multiple processes and endpoints). This same service is used to create and update ServiceMetaData.	PUT	ServiceMetadata
UC05	Admin ServiceGroup	**Erase Service Metadata**	Remove all information about one specific service (i.e. all related processes and endpoints definitions)	DELETE	ServiceMetadata
UC06	User	**Retrieve Service Group**	Obtain the list of services provided by a specific receiver participant (collection of references to the ServiceMetaData's) This service provides the information related to the *Service Group* according to the input duplet participantIdentifier + participantIndentifierScheme. Returns information from the ServiceMetadata table only (references to actual MetaData).	GET	ServiceGroup

ID	Actor	UC	Short description	Oper.	Data
UC07	User	**Retrieve Service Metadata**	Obtain detailed definition about one specific service of a specific participant for all supported transport. This service retrieves the SignedServiceMetadata according to the input quadruplet participantIdentifier+participantIndentifierScheme+documentIdentifier+documentIdentifierScheme. Returns information from the Endpoint table.	GET	SignedServiceMetadata

2.3.4. Story

The following “story” shows a typical example of successive usage of the use cases (when applicable) as it might happen in real life. Each step of this story is prefixed with the use case identifier if the SMP (the System) is involved. If ‘N/A’ is mentioned, some action part of the ‘story’ happens without any involvement of the SMP.

- UC01: As System Admin, I create a new 'Admin SMP' to allow the creation and the management of a new ServiceGroup for a participant.
- UC02: As Admin SMP, I create a new ServiceGroup and define the related administrator “Admin ServiceGroup” to allow the management of ServiceMetadata for the related participant.
- UC04: As Admin ServiceGroup, I define ALL the ServiceMetadata for the participant that I administer.
- N/A: As User, I ask the DNS to resolve the address of the SMP hosting the receiver's metadata.
- UC07: As User, I retrieve the definition of the service (metadata) I need to invoke to send a document to the receiver.
- N/A: As User, I send the document to the receiver.

2.4. Administration use cases

Paragraphs 2.4 and 2.5 define the use cases listed above with more detail.

The following use cases (of this paragraph 2.4) are intended for the different types of administrators in order to define all services (*ServiceGroup* and *ServiceMetada*).

They are based on a specific administration XSD, based on the standard OASIS' one (cf. 4.1.3 – "Extended SMP XSD for ADMIN services (PUT)")

2.4.1. *UC01 - Manage Administrators*

This use case introduce the foundation for an administration console: creating an 'Admin SMP' user is the task of superuser, and no REST service shall consequently support that functionality. As this is a necessary functionality, this one should be included into the administration console.

2.4.1.1. Use Case

Brief description

Create and modify administrator information in SMP table 'Administrator'.

Note: this temporary solution will later be replaced by functionality in a user friendly administration console.

Actors

System Admin

Preconditions

The actor (system admin) has all access rights to modify content of SMP configuration tables

Basic flow event

Step	
1	System admin creates a new administrator in table 'Administrator'

2 Use case ends with success

Alternative flows

1a **Administrator must be removed**

1a1 System admin removes all ServiceGroup definitions linked to that administrator by calling "DeleteServiceGroup" SMP service for all ServiceGroups this administrator is linked to (as defined by the "ownership" relationship).

1a2 System admin removes the administrator from table 'Administrator'

1b **New administrator must take over administration of some participant(s)**

1b1 After creating the new user (step 1), the system admin reassigns specific ServiceGroup's to that user by changing the 'username' foreign key in table Ownership.

1b2 Use case ends

1c **Administrator already exists and must be modified**

1c1 System admin modifies some data (role, password) of the user in table 'User'

1c2 Use case ends

Post conditions

Successful conditions

Administrator definition has been modified

Failure conditions

N/A

2.4.1.2. REST Service: None

This functionality should be implemented into the administrator’s console of the SMP which is not further detailed it the present document.

2.4.2. UC02 - Create or Update Service Group

2.4.2.1. Use case

Brief description

Create a new ServiceGroup for a new receiver participant.
This service stores the Service Group and links it to the specified duplet participantIdentifier + participantIndentifierScheme.
Information is store into ServiceGroup table.
This same service is used to create and update a ServiceGroup.

Actors

Admin SMP

Preconditions

The authenticated user as the role of "Admin SMP"

Identifier and scheme of the service group provided in the request must comply to the policy defined in [REF4]

Basic flow event

Step

1 The receiver declares its service group and the related Administrator (Admin ServiceGroup) to the SMP

2 The SMP authenticates the user, validates the request, and add or replace the information into its configuration database (into table ServiceGroup).

3 The receiver receives the confirmation that the definitions were stored properly with HTTP response "201 Created".

4 Use case ends with success

Alternative flows

3a **ServiceGroup already exists**

3a1 The receiver receives the confirmation that the definitions were updated properly with HTTP response "200 OK".

3a2 Use case ends with success

Exception flows

1a **SMP is not reachable**

1a1 The user receives a network connection error

1a2 Use case ends

2a **Authentication / authorization fails**

2a1 The SMP replies with HTTP error "401 Unauthorized"

2a2 The receiver receives the error message

2a3 Use case ends

2b **Request is not well formed (or any other business/technical error)**

2b1 The SMP replies with HTTP error "500 Server Internal Error" with details on the error allowing to identify the error in the request (cf. "Error codes" table below)

2b2 The receiver receives the error message

2b3 Use case ends

Post conditions

Successful conditions

ServiceGroup is either created or updated, and the corresponding "Admin ServiceGroup" is defined.

Failure conditions

In case of error, no change occurs into the configuration database and the response gives technical details on the exception condition

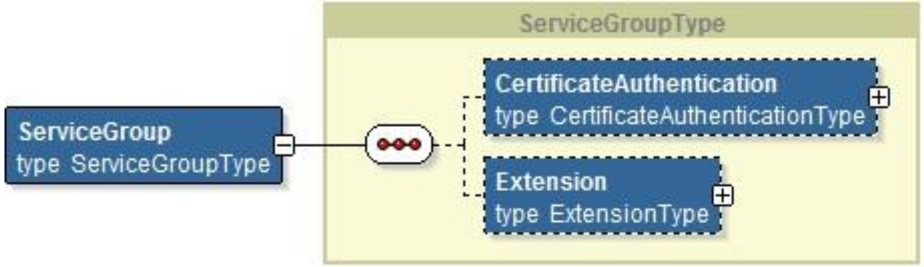
2.4.2.2. REST Service: PutServiceGroup

Input:

- The participant's identifier and identifier's scheme (ParticipantIdentifier) **in the HTTP header**

- **In the TEXT:** a modified version of the ServiceGroup type of original OASIS XSD (cf. 4.1.3 – "Extended SMP XSD for ADMIN services (PUT)") containing:

    - The Participant's identifier and scheme that uniquely identifies this service group;
    - the Certificate information required for authenticating the user as "Admin Service Group" for this service group; required only if the system of the "Admin Service Group" is not on the same (V)LAN than the SMP – cf. 2.6.4 -"Reverse proxy"),
    - Optionally, the Extension information in the HTTP TEXT
    - Even though the ServiceGroup may contain no element, ServiceGroup element itself must be present.



Details on the structure Certificate identifier:

- The following attributes of the certificate will be used in this order:
    - CN,
    - O and
    - C.

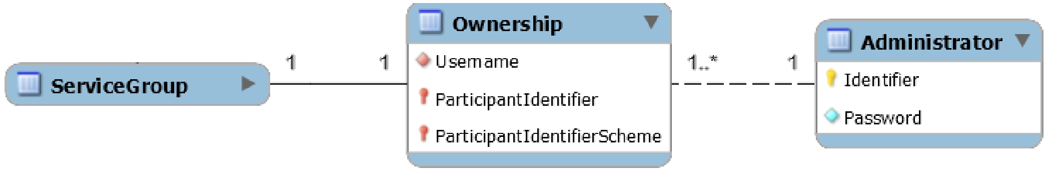
- As an example, the following certificate sno=0001&subject=EMAILADDRESS=receiver@test.be, CN=SMP_receiverCN, OU=B4, O=DIGIT, L=Brussels, ST=BE, C=BE&validfrom=Jun 1 10:37:53 2015 CEST&validto=Jun 1 10:37:53 2035 CEST&issuer=EMAILADDRESS=root@test.be,CN=rootCN,OU=B4,O=DIGIT,L=Brussels,ST=BE,C=BE

will be provided as such, and in that order:

<CertificateIdentifer>sno=0001,CN=SMP_receiverCN, O=DIGIT,C=BE</CertificateIdentifer>

Execution:

- Start a new transaction.
- Create or update (overwrites) the corresponding rows in the configuration, ownership and ServiceGroup identified by the participant's identifier and identifier's scheme keys:



  - If element "CertificateAuthentication" is present in the text, then use this as information to store as "Identifier"
  - If not, store instead the basic authentication information provided in the header.
- If it is a newly created ServiceGroup, invoke SML service "Create Business Identifier".
- If SML service invocation succeeded, commit the transaction.
- If SML service invocation failed:
  - rollback the transaction;
  - Response to this service is "failure".

Output:

Return a response confirming the success (or eventually the failure) of the operation.

Sample Request

HTTP Header

```
PUT http://130.206.118.4:8080/cipa-smp-full-webapp/iso6523-actorid-upis::0088:5798000000112 HTTP/1.1
Accept-Encoding: gzip,deflate
Content-Type: text/xml;charset=UTF-8
Authorization: Basic dGVzdGVyOnRlc3Q=
Content-Length: 658
Host: 130.206.118.4:8080
Connection: Keep-Alive
User-Agent: Apache-HttpClient/4.1.1 (java 1.5)
```

Text (Basic Authentication)

```
<ServiceGroup
  xmlns="http://docs.oasis-open.org/bdxr/ns/SMP/2014/07"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="E:\eDelivery\SMP\CID\untitled7-ADMIN.xsd">

  <ParticipantIdentifier scheme="ehealth-participantid-qns">urn:germany:ncpb
  </ParticipantIdentifier>


  <Extension><something>text</something></Extension>

</ServiceGroup>
```

Text (Certificate-based Authentication)

```
<ServiceGroup
  xmlns="http://docs.oasis-open.org/bdxr/ns/SMP/2014/07"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="E:\eDelivery\SMP\CID\untitled7-ADMIN.xsd">

  <ParticipantIdentifier scheme="ehealth-participantid-qns">urn:germany:ncpb
  </ParticipantIdentifier>

  <CertificateAuthentication>
     <CertificateIdentifier>CN=SMP_1000000181,O=myDearCompany,C=DK:406b2abf0bd1d46ac4292efee597d414</CertificateIdentifier>
  </CertificateAuthentication>

   <Extension><something>text</something></Extension>
</ServiceGroup>
```

Sample Response

HTTP header

```
HTTP/1.1 201 Created
Server: Apache-Coyote/1.1
Pragma: No-cache
Expires: Thu, 01 Jan 1970 01:00:00 CET
Content-Length: 0
Date: Wed, 27 Jan 2016 10:32:40 GMT
Cache-Control: no-cache, proxy-revalidate
Connection: Keep-Alive
```

NB: if the ServiceGroup previously existed, "200 OK" will be returned as HTTP response instead of "201 Created" as show in the above example.

Text

N/A.

Error codes

HTTP code	HTTP Message	Business code	Meaning
201	Created	n/a	The PUT operation completed successfully
400	Bad Request	n/a	There is a format error in the request
401	Unauthorized	n/a	The user is not granted the right to issue this request
500	Server Internal Error	TECHNICAL	Some unexpected technical error occurred (detailed information is available in the response)
500	Server Internal Error	XSD_INVALID	The XML included in the request is not validate against the XSD defining the input structure
500	Server Internal Error	MISSING_FIELD	Some field that is optional in the XSD but mandatory for this invocation is missing (missing field's name in description)
500	Server Internal Error	WRONG_FIELD	Some field is valid against XSD definition, but the more specific content is invalid (erroneous field's name in description)
500	Server Internal Error	OUT_OF_RANGE	Some numeric (or date field) is out of the valid range (erroneous field's name in description)
500	Server Internal Error	UNAUTHOR_FIELD	Some field that is optional in the XSD but forbidden for this invocation is present (unauthorized field's name in description)
500	Server Internal Error	FORMAT_ERROR	Some field is expected to have a specific format is not valid (erroneous field's name in description)

NB: for HTTP error 500, the business code and the description are in the response and compliant to the ErrorResponseType as described in cf. 1.1 - "

Detailed Errors’ structure").

Audit

The following information must be audited for this service (more details under $2.6.5 – 'Auditing'):

- AdministratorIdentifier
- AccessTime
- Operation
- ParticipantIdentifier
- ParticipantIdentifierScheme
- IpAddress
- RequestHeader
- RequestText
- ResponseHeader
- HTTP code
- Business code
- ErrorDescription

2.4.3. UC03 - Erase Service Group

2.4.3.1. Use case

Brief description

Erases the service group definition AND the list of services for the specified receiver participant.

Actors

Admin SMP

Preconditions

The authenticated user has the role of "Admin SMP"

Referenced service group was previously defined

Basic flow event

Step

1 The receiver request its service group to be removed from the SMP

2 The SMP authenticates the user, validates the request, and remove all the information from its configuration database (from table ServiceGroup and all the children tables).

3 The receiver receives the confirmation that the definitions were removed properly with HTTP response "200 OK".

4 Use case ends with success

Exception flows

1a **SMP is not reachable**

1a1 The user receives a network connection error

1a2 Use case ends

2a **Authentication / authorization fails**

2a1 The SMP replies with HTTP error "401 Unauthorized"

2a2 The receiver receives the error message

2a3 Use case ends

2b **Request is not well formed (or any other business/technical error)**

2b1 The SMP replies with HTTP error "500 Server Internal Error" with details on the error allowing to identify the error in the request (cf. "Error codes" table below)

2b2 The receiver receives the error message

2b3 Use case ends

2c **ServiceGroup is not defined**

2c1 The SMP replies with HTTP error "404 Resource not found"

2c2 The receiver receives the error message

2c3 Use case ends

Post conditions

Successful conditions

The specified service group is removed with all its related information

Failure conditions

In case of error, no change occurs into the configuration database and the response gives technical details on the exception condition

2.4.3.2. REST Service: DeleteServiceGroup

Input: ServiceGroup identifier: ParticipantIdentifier, ParticipantIdentifierScheme in the HTTP header

Execution:

The username in the request is verified to be the owner of the specified Service Group. If not, the operation is rejected.

Start a new transaction.

Delete ALL information related to that service group in tables: Endpoint, Process, ServiceMetadata and finally the ServiceGroup itself where the *ParticipantIdentifiers* match the specified *ServiceGoup* identifier.

Invoke SML service "Delete Business Identifier".

If SML service invocation succeeded, commit the transaction.
If SML service invocation failed:

- rollback the transaction;
- Response to this service is "failure".

Output: HTTP 200 if done, 404 if the specified service group does not exist and 500 if any error occurred.

Sample Request

HTTP Header

```
DELETE http://130.206.118.4:8080/cipa-smp-full-webapp/iso6523-actorid-upis::0088:5798000000112 HTTP/1.1
Accept-Encoding: gzip,deflate
Authorization: Basic dGVzdGVyOnRlc3Q=
Host: 130.206.118.4:8080
Connection: Keep-Alive
User-Agent: Apache-HttpClient/4.1.1 (java 1.5)
```

Text

N/A

Sample Response

HTTP header

```
HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Pragma: No-cache
Expires: Thu, 01 Jan 1970 01:00:00 CET
Content-Length: 0
Date: Thu, 21 Jan 2016 16:11:47 GMT
Cache-Control: no-cache, proxy-revalidate
Connection: Keep-Alive
```

Text

N/A

Error codes

HTTP code	HTTP Message	Business code	Meaning
200	OK	n/a	The request was completed successfully
400	Bad Request	n/a	There is a format error in the request
401	Unauthorized	n/a	The user is not granted the right to issue this request
404	Resource not found	n/a	The requested information was not found

HTTP code	HTTP Message	Business code	Meaning
500	Server Internal Error	TECHNICAL	Some unexpected technical error occurred (detailed information is available in the response)

Audit

The following information must be audited for this service (more details under $2.6.5 – 'Auditing'):

- AdministratorIdentifier
- AccessTime
- Operation
- ParticipantIdentifier
- ParticipantIdentifierScheme
- IpAddress
- RequestHeader
- ResponseHeader
- HTTP code
- Business code
- ErrorDescription

2.4.4. UC04 - Create or Update Service Metadata

2.4.4.1. Use case

Brief description

Publish detailed information about one specific document service (multiple processes and endpoints).
This same service is used to create and update ServiceMetaData.
(Cf. [REF8]§2.1*) A sender* (ed. "user") *may want to discover what document types can be handled by a specific participant identifier. Such discovery is relevant for applications supporting several equivalent business processes. Knowing the capabilities of the recipient is valuable information to a sender application and ultimately to an end user. E.g. the end user may be presented with a choice between a "simple" and a "rich" business process.*
This is enabled by a pattern where the sender first retrieves the ServiceGroup entity, which holds a list of references to the ServiceMetadata resources associated with it. The ServiceMetadata in turn holds the metadata information that describes the capabilities associated with the recipient participant identifier

Actors

Admin ServiceGroup

Preconditions

The authenticated user has the role of "Admin ServiceGroup"

Admin ServiceGroup user initiating the request is linked to the specified ServiceGroup

The certificate of the "Admin ServiceGroup" is valid

The certificate information of the "Admin ServiceGroup" was previously stored in the configuration

Identifier and scheme of the service group and documents provided in the request must comply to the policy defined in [REF4]

Basic flow event

Step	
1	The receiver request its service metadata to be put into the SMP
2	The SMP verifies the certificate of the "Admin ServiceGroup" against its information in the database, validates the request, and either create or update all the information into its configuration database (into table ServiceMetadata and all its children tables).
3	The receiver receives the confirmation that the definitions were created properly with HTTP response "201 Created".
4	Use case ends

Alternative flows

3a	**ServiceMetadata already exists**
3a1	The receiver receives the confirmation that the definitions were updated properly with HTTP response "200 OK".

3a2 Use case ends with success

Exception flows

1a **SMP is not reachable**

1a1 The user receives a network connection error

1a2 Use case ends with success

2a **Authentication / authorization fails**

2a1 The SMP replies with HTTP error "401 Unauthorized"

2a2 The receiver receives the error message

2a3 Use case ends

2b **Request is not well formed (or any other business/technical error)**

2b1 The SMP replies with HTTP error "500 Server Internal Error" with details on the error allowing to identify the error in the request (cf. "Error codes" table below)

2b2 The receiver receives the error message

2b3 Use case ends

2c **ServiceGroup is not defined**

2c1 The SMP replies with HTTP error "404 Resource not found"

2c2 The receiver receives the error message

2c3 Use case ends

Post conditions

Successful conditions

ServiceMetadata is defined

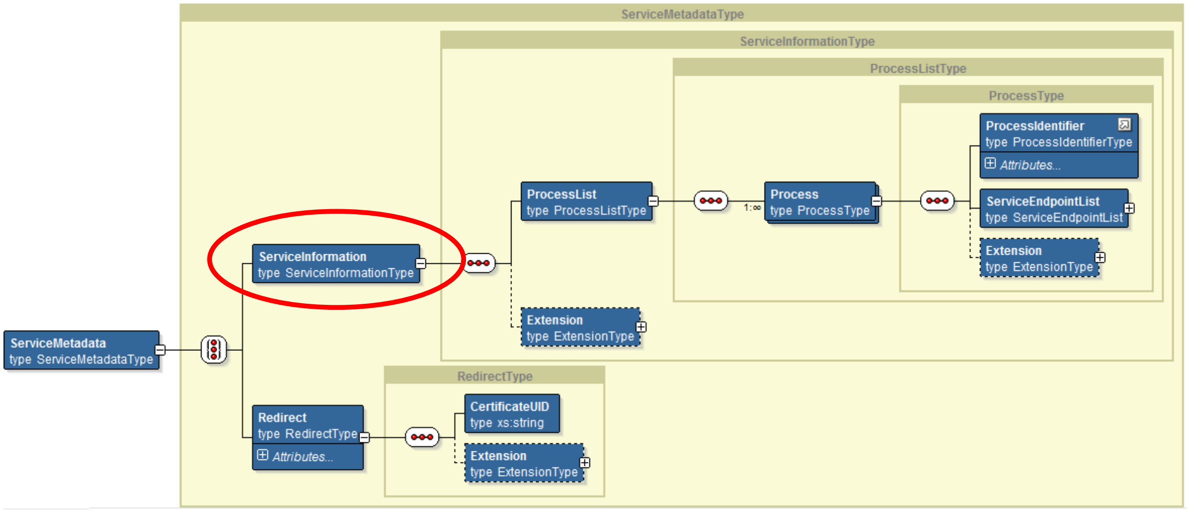
Failure conditions

In case of error, no change occurs into the configuration database and the response gives technical details on the exception condition

2.4.4.2. REST Service : PutServiceMetadata

Input:

- ServiceGroup and Document's identifiers in the URL and
- *ServiceMetadata* in the text



This input structure, from the *ServiceInformation* node down to the Process' leaves will fully define the content of the referenced service metadata as defined by the four identifiers of the participant AND related specific document.

This means that the configuration of a Service must be done with a single call (for all *Processes*) to this service and it can be considered that all previously existing information in ServiceInformation, Process and Endpoint tables are discarded (if they exist) and completely replaced by the newly provided information.

Execution:

Start a new transaction.

For each process in the input and in the database:

- If the process is in both input and already exists in the database : update the database information of the endpoint as specified in the input;
- If the process is in the input only : create an new endpoint in the database as specified in the input;
- If the process is in the database only : delete that endpoint from the database.

In case of error in one of the above three alternatives,

- abort the loop
- rollback the transaction
- Response to this service is "failure".

If no error occurred:

- Commit the transaction
- Response to this service is "success".

Authorization

The operation will be allowed if and only the authenticated user matches the "Admin ServiceGroup" user linked to the ServiceGroup or is "Admin SMP".

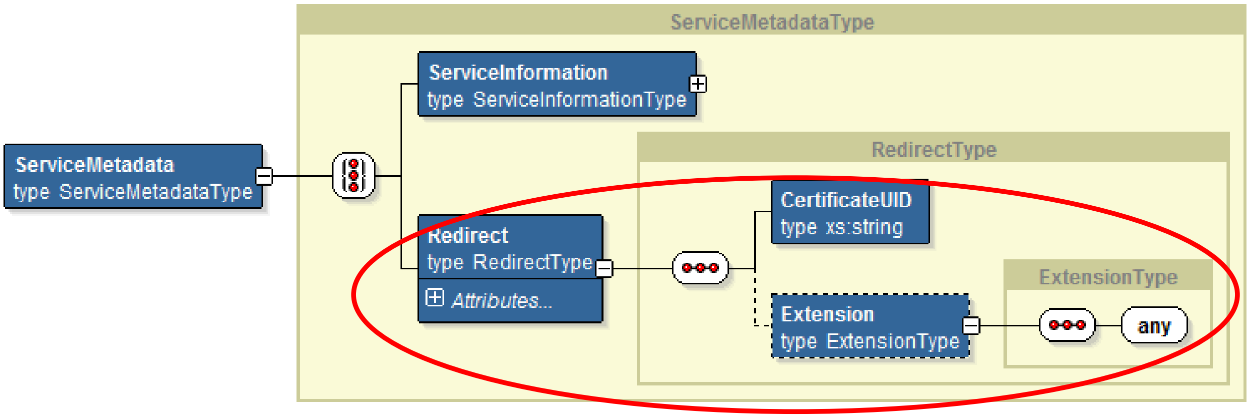
For this user to be the eligible "Admin ServiceGroup" it must have been referenced as such in the ServiceGroup definition (cf. PutServiceGroup) by an "Admin SMP" user (defined him by the "System Administrator") via service "PutServiceGroup".

All the provided information will either be created in the configuration (put = *create*) or be **overwritten** (put = *update*); i.e. this 'put' operation does both.

Redirection

As explained above, in some cases ServiceMetadata information can be stored in 'another SMP'; i.e. another SMP than the one that is queried by the user. In such case, 'redirect' information is provided to the user to allow him to query the appropriate SMP for obtaining the ServiceMetadata information from the relevant SMP.

For that to be possible, the receiver must eventually be able to store that redirect information. That is why this service provides this possibility, by allowing provision of "Redirect" information instead of the "ServiceInformation" itself:



The fields are in used as follows:

- *CertificateUID* : holds the Subject Unique Identifier of the certificate of the destination SMP. A client SHOULD validate that the Subject Unique Identifier of the certificate used to sign the resource at the destination SMP matches the Subject Unique Identifier published in the redirecting SMP
- *href* attribute of the Redirect element contains the full address of the destination SMP record that the client is redirected to.
- Extension : not defined and optional

Note about cascaded redirections:

In the case where a client encounters such a redirection element, the client MUST follow the first redirect reference to the alternative SMP. If the SignedServiceMetadata resource at the alternative SMP also contains a redirection element, the client SHOULD NOT follow that redirect. It is the responsibility of the client to enforce this constraint.

Output: HTTP response code 200 if ok, 401 if not allowed and 500 if any other error occurred. Details are available in the response text.

Sample Request 1

This example sends actual information of the service, and uses a certificate.

HTTP Header (with certificate)

```
PUT http://130.206.118.4:8080/cipa-smp-full-webapp/iso6523-actorid-upis::0088:5798000000112/services/busdox-docid-
qns::urn:oasis:names:specification:ubl:schema:xsd:Invoice-
12::Invoice%23%23urn:www.cenbii.eu:transaction:biicoretrdm010:ver1.0:%23urn:www.peppol.eu:bis:peppol4a:ver1.0::2.0 HTTP/1.1
Accept-Encoding: gzip,deflate
Content-Type: text/xml;charset=UTF-8
Client-Cert: sno=0001&subject=EMAILADDRESS=receiver@test.be, CN=SMP_receiverCN, OU=B4, O=DIGIT, L=Brussels, ST=BE,
C=BE&validfrom=Jun 1 10:37:53 2015 CEST&validto=Jun 1 10:37:53 2035
CEST&issuer=EMAILADDRESS=root@test.be,CN=rootCN,OU=B4,O=DIGIT,L=Brussels,ST=BE,C=BE

Content-Length: 2497
Host: 130.206.118.4:8080
Connection: Keep-Alive
User-Agent: Apache-HttpClient/4.1.1 (java 1.5)
```

NB: the "Client-Cert" value in the HTTP header above is only an example that is specific to production and acceptance environments at DIGIT and should not be considered as constraining.

Text (Information)

```
<ServiceMetadata xmlns="http://docs.oasis-open.org/bdxr/ns/SMP/2014/07" xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-
200401-wsswssecurity- utility-1.0.xsd"   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://docs.oasis-
open.org/bdxr/ns/SMP/2014/07 untitled7-ADMIN.xsd">
        <ServiceInformation>
                <ProcessList>
                        <Process>
                                <ProcessIdentifier scheme="cenbii-procid-
ubl">urn:www.cenbii.eu:profile:bii04:ver1.0</ProcessIdentifier>
                                <ServiceEndpointList>
                                        <Endpoint transportProfile="busdox-transport-as2-ver1p0">
                                                <EndpointURI>http://d02di1010873.net1.cec.eu.int:7080/cipa-
dispatcher/AS2Receiver</EndpointURI>
                                                <RequireBusinessLevelSignature>false</RequireBusinessLevelSignature>
                                                <ServiceActivationDate>2003-01-01T00:00:00</ServiceActivationDate>
                                                <ServiceExpirationDate>2020-05-01T00:00:00</ServiceExpirationDate>
                                                <Certificate>CERTIFICATEA</Certificate>
                                                <ServiceDescription>invoice service AS2</ServiceDescription>
                                                <TechnicalContactUrl>https://example.com</TechnicalContactUrl>
                                        </Endpoint>
                                        <Endpoint transportProfile="busdox-transport-as2-ver1p0">
                                                <EndpointURI>http://busdox.org/otherService/as2</EndpointURI>
                                                <RequireBusinessLevelSignature>false</RequireBusinessLevelSignature>
                                                <ServiceActivationDate>2009-05-01T09:00:00</ServiceActivationDate>
                                                <ServiceExpirationDate>2016-05-01T09:00:00</ServiceExpirationDate>
                                                <Certificate>CERTIFICATEA</Certificate>
                                                <ServiceDescription>invoice service</ServiceDescription>
                                                <TechnicalContactUrl>https://example.com</TechnicalContactUrl>
                                        </Endpoint>
                                </ServiceEndpointList>
                        </Process>
                </ProcessList>
        </ServiceInformation>
</ServiceMetadata>
```

Sample Request 2

This example sends redirect information, and uses basic authentication.

HTTP Header (basic authentication)

```
PUT http://130.206.118.4:8080/cipa-smp-full-webapp/iso6523-actorid-upis::0088:5798000000112/services/busdox-docid-
qns::urn:oasis:names:specification:ubl:schema:xsd:Invoice-
12::Invoice%23%23urn:www.cenbii.eu:transaction:biicoretrdm010:ver1.0:%23urn:www.peppol.eu:bis:peppol4a:ver1.0::2.0 HTTP/1.1
Accept-Encoding: gzip,deflate
Content-Type: text/xml;charset=UTF-8
Authorization: Basic dGVzdGVyOnRlc3Q=
Content-Length: 2497
Host: 130.206.118.4:8080
Connection: Keep-Alive
User-Agent: Apache-HttpClient/4.1.1 (java 1.5)
```

Text (Redirect)

```
<ServiceMetadata

  xmlns="http://docs.oasis-open.org/bdxr/ns/SMP/2014/07"
  xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wsswssecurity- utility-1.0.xsd"
  xmlns:ids="http://busdox.org/transport/identifiers/1.0/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" >

		<Redirect href="http://serviceMetadata2.eu/busdox-actorid-upis%3A%3A0010%3A5798000000001/services/bdx-docid-
qns%3A%3Aurn%3Aoasis%3Anames%3Aspecification%3Aubl%3Aschema%3Axsd%3AInvoice- 2%3A%3AInvoice%23%23UBL-2.0">
				<CertificateUID>PID:9208-2001-3-279815395</CertificateUID>
				<Extension>
						<ex:Test xmlns:ex="http://test.eu">Test</ex:Test>
				</Extension>
		</Redirect>

</ServiceMetadata>
```

Sample Response (applicable for both examples requests above)

HTTP header

```
HTTP/1.1 201 Created
Server: Apache-Coyote/1.1
Pragma: No-cache
Expires: Thu, 01 Jan 1970 01:00:00 CET
Content-Length: 0
Date: Fri, 22 Jan 2016 09:46:10 GMT
Cache-Control: no-cache, proxy-revalidate
Connection: Keep-Alive
```

NB: if the ServiceMetadata previously existed, "200 OK" will be returned as HTTP response instead of "201 Created" as show in the above example.

Text

N/A

Error codes

HTTP code	HTTP Message	Business code	Meaning
201	Created	n/a	The PUT operation completed successfully
400	Bad Request	n/a	There is a format error in the request
401	Unauthorized	n/a	The user is not granted the right to issue this request
500	Server Internal Error	TECHNICAL	Some unexpected technical error occurred.(detailed information available in the response)
500	Server Internal Error	XSD_INVALID	The XML included in the request is not validate against the XSD defining the input structure
500	Server Internal Error	MISSING_FIELD	Some field that is optional in the XSD but mandatory for this invocation is missing (missing field's name in description)
500	Server Internal Error	WRONG_FIELD	Some field is valid against XSD definition, but the more specific content is invalid (erroneous field's name in description)
500	Server Internal Error	OUT_OF_RANGE	Some numeric (or date field) is out of the valid range (erroneous field's name in description)
500	Server Internal Error	UNAUTHOR_FIELD	Some field that is optional in the XSD but forbidden for this invocation is present (unauthorized field's name in description)
500	Server Internal Error	FORMAT_ERROR	Some field is expected to have a specific format is not valid (erroneous field's name in description)

Audit

The following information must be audited for this service (more details under $2.6.5 – 'Auditing'):

- AdministratorIdentifier
- AccessTime
- Operation
- ParticipantIdentifier
- ParticipantIdentifierScheme
- DocumentIdentifier
- DocumentIdentifierScheme
- IpAddress
- RequestHeader
- RequestText
- ResponseHeader
- HTTP code
- Business code
- ErrorDescription

2.4.5. UC05 - Erase Service Metadata

2.4.5.1. Use case

Brief description

Remove all information about one specific service (i.e. all related processes and endpoints definitions)

Actors

Admin ServiceGroup

Preconditions

The user knows the address of the SMP.

Admin ServiceGroup administrator initiating the request is linked to the specified ServiceGroup

The authenticated user has the role of "Admin ServiceGroup"

The referenced ServiceMetadata exists

Basic flow event

Step

1 The receiver request its service metadata to be removed from the SMP

2 The SMP authenticates the user, validates the request, and delete any information from the referenced ServiceMetadata from its configuration database (from table ServiceMetadata and all its tables).

3 The receiver receives the confirmation that the definitions were removed properly with HTTP response "200 OK".

4 Use case ends with success

Exception flows

1a **SMP is not reachable**

1a1 The user receives a network connection error

1a2 Use case ends

2a **Authentication / authorization fails**

2a1 The SMP replies with HTTP error "401 Unauthorized"

2a2 The receiver receives the error message

2a3 Use case ends

2b **Request is not well formed (or any other business/technical error)**

2b1 The SMP replies with HTTP error "500 Server Internal Error" with details on the error allowing to identify the error in the request (cf. "Error codes" table below)

2b2 The receiver receives the error message

2b3 Use case ends

2c **ServiceGroup or ServiceMetadata is not defined**

2c1 The SMP replies with HTTP error "404 Resource not found"

2c2 The receiver receives the error message

2c3 Use case ends

Post conditions

Successful conditions

ServiceMetadata are absent

Failure conditions

In case of error, no change occurs into the configuration database and the response gives technical details on the exception condition

2.4.5.2. REST Service: DeleteServiceMetadata

Input: ServiceMetadata identifier in the HTTP header

Execution:

Authorization

The operation will be allowed if and only the authenticated user matches the “Admin ServiceGroup” user linked to the ServiceGroup or is "Admin SMP".

For this user to be the eligible “Admin ServiceGroup” it must have been referenced as such in the ServiceGroup definition (cf. PutServiceGroup) by an “Admin SMP” user (defined him by the “System Administrator”) via service “PutServiceGroup”.

Start a new transaction.

> NB:
>
> If no more ServiceMetadata information is available on the related ServiceGroup, the limited information on the ServiceGroup is nevertheless kept to allow keeping track of the previously defined administrator and the service group. Should it be deleted, it is the responsibility of the “Admin SMP” user to issue the required operation (DeleteServiceGroup) if necessary.

Delete in one single transaction any information related to that service in tables: Endpoint, Process and finally ServiceMetadata where participant and documents identifiers match the provided ServiceMetadata identifier:

For each endpoint in the database: invoke SML service “Delete Business Identifier” and delete that endpoint from the database.

In case of error in one of the above deletion, abort the loop and (try to) undo what was previously done:

- Rollback the transaction
- For each endpoint that was deleted, invoke SML service “Create Business Identifier” to re-create as specified in the database, those that were deleted;

  Continue even if an error occurs in order to rollback as much as possible.
  In such case though, inform the administrator with any possible technical mean (mail, log, …) that the configuration has become inconsistent with no possible automatic recovery by specifying which operation failed (don’t retry later the SML because it might occur after a successful completion 2nd call and corrupt the configuration after restoration).

- Response to this service is “failure”.

If no error occurred:

- Commit the transaction
- Response to this service is “success”.

Output: HTTP 200 if done, 404 if the service metadata or the service group does not exist and 500 if any error occurred.

Sample Request

HTTP Header

```
DELETE http://130.206.118.4:8080/cipa-smp-full-webapp/iso6523-actorid-upis::0088:5798000000112/services/busdox-docid-
qns::urn:oasis:names:specification:ubl:schema:xsd:Invoice-
12::Invoice%23%23urn:www.cenbii.eu:transaction:biicoretrdm010:ver1.0:%23urn:www.peppol.eu:bis:peppol4a:ver1.0::2.0 HTTP/1.1
```

```
Accept-Encoding: gzip,deflate
Authorization: Basic dGVzdGVyOnRlc3Q=
Host: 130.206.118.4:8080
Connection: Keep-Alive
User-Agent: Apache-HttpClient/4.1.1 (java 1.5)
```

Text

N/A

Sample Response

HTTP header

```
HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Pragma: No-cache
Expires: Thu, 01 Jan 1970 01:00:00 CET
Content-Length: 0
Date: Fri, 22 Jan 2016 09:47:52 GMT
Cache-Control: no-cache, proxy-revalidate
Connection: Keep-Alive
```

Text

N/A

Error codes

HTTP code	HTTP Message	Business code	Meaning
200	OK	n/a	The request was completed successfully
400	Bad Request	n/a	There is a format error in the request
401	Unauthorized	n/a	The user is not granted the right to issue this request
404	Resource not found	n/a	The requested information was not found
500	Server Internal Error	TECHNICAL	Some unexpected technical error occurred.(detailed information available in the response)

Audit

The following information must be audited for this service (more details under $2.6.5 – 'Auditing'):

- AdministratorIdentifier
- AccessTime
- Operation
- ParticipantIdentifier
- ParticipantIdentifierScheme
- DocumentIdentifier
- DocumentIdentifierScheme
- IpAddress
- RequestHeader
- ResponseHeader
- HTTP code
- Business code
- ErrorDescription

2.5. Information retrieval use cases

The following use cases are mainly intended for the sender participants' type of users in order for them to collect information on the target receivers. They are based on the ‘standard’ OASIS XSD (cf. 4.1.2 – “Extended SMP XSD for standard services (GET)”)

2.5.1. UC06 - Retrieve Service Group

2.5.1.1. Use case

Brief description

Obtain the list of services provided by a specific receiver participant (collection of references to the ServiceMetaData's)

This service provides the information related to the Service Group according to the input duplet participantIdentifier+participantIndentifierScheme.

Returns information from the ServiceMetadata table only (references to actual MetaData).

(Cf. [REF8]§2.1) *A sender* (ed. "user") *may want to discover what document types can be handled by a specific participant identifier.*

Such discovery is relevant for applications supporting several equivalent business processes.

This is enabled by a pattern where the sender first retrieves the ServiceGroup entity, which holds a list of references to the ServiceMetadata resources associated with it.

The ServiceMetadata in turn holds the metadata information that describes the capabilities associated with the recipient participant identifier

Actors

User

Preconditions

The requester application has previously resolved the address of the SMP from the DNS.

Referenced service group was previously defined by the receiver

Basic flow event

Step	
1	The user request one service group's references to the SMP
2	The SMP validates the request, and retrieve the information from its configuration database (into table ServiceGroup and Service Metadata tables).
3	The user receives the participant's service group information
4	Use case ends with success

Exception flows

1a **SMP is not reachable**

1a1 The user receives a network connection error

1a2 Use case ends

2a **Request is not well formed (or any other business/technical error)**

2a1 The SMP replies with HTTP error "500 Server Internal Error" with details on the error allowing to identify the error in the request (cf. "Error codes" table below)

2a2 The receiver receives the error message

2a3 Use case ends

2b **ServiceGroup is not defined**

2b1 The SMP replies with HTTP error "404 Resource not found"

2b2 The receiver receives the error message

2b3 Use case ends

Post conditions

Successful conditions

The user receives ServiceGroup information for the requested receiver participant.

Failure conditions

The user received no ServiceGroup information about the requested receiver participant.

2.5.1.2. REST Service: GetServiceGroup

Input: *ParticipantIdentifier*

Represents the business level endpoint key and key type, e.g. a DUNS or GLN number that is associated with a group of services. See the ParticipantIdentifier section of the ‘Common Definitions’ document [BDEN-CDEF] for information on this data type.

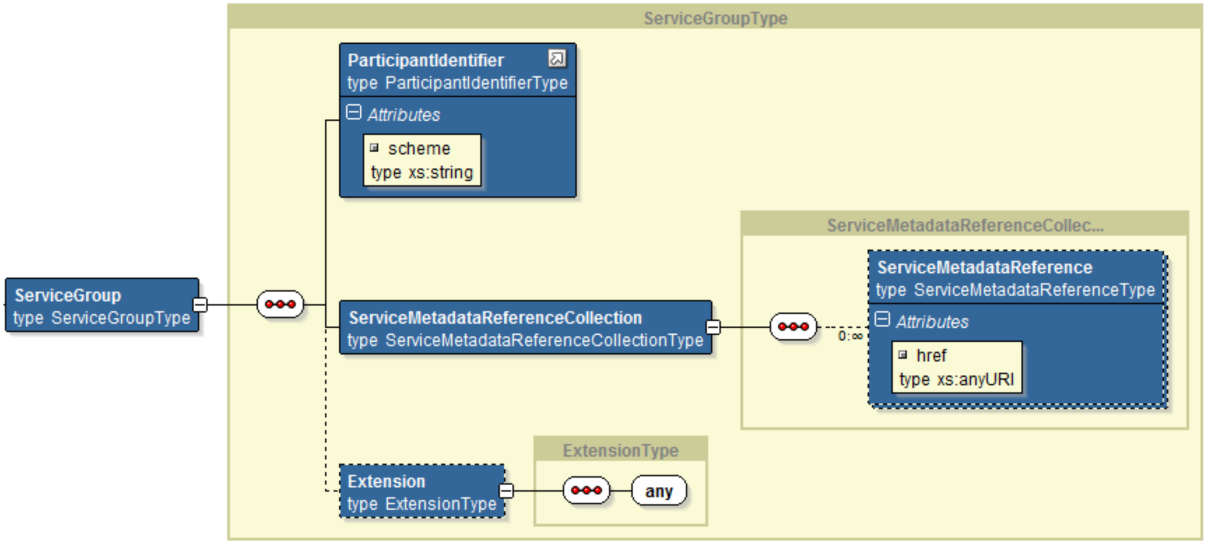
Execution:

Selects all service Metadata related to the ServiceGroup specified by the provided ParticipantIdentifier and build the corresponding URI from it.

NB: there is no interaction with the SML (from the SMP).

Output: *ServiceGroup*

This SMP service will return the reference URI for the user that will enable him to retrieve all information about the services that a participant (receiver) participates in; i.e. all service’s metadata of the specified participant. To obtain the details on those services, the ServiceMetadata can be obtained from the SMP using the references provided.



Sample Request

HTTP Header

```
GET http://130.206.118.4:8080/cipa-smp-full-webapp/iso6523-actorid-upis::0088:5798000000112 HTTP/1.1
Accept-Encoding: gzip,deflate
Host: 130.206.118.4:8080
Connection: Keep-Alive
User-Agent: Apache-HttpClient/4.1.1 (java 1.5)
```

Text

```
N/A
```

Sample Response

HTTP header

```
HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Content-Type: text/xml
Content-Length: 959
Date: Thu, 21 Jan 2016 08:38:33 GMT
Cache-Control: proxy-revalidate
Connection: Keep-Alive
```

Text

```
<ServiceGroup xmlns="http://docs.oasis-open.org/bdxr/ns/SMP/2014/07">
  <ParticipantIdentifier scheme="busdox-actorid-upis">
    0010:5798000000001
  </ParticipantIdentifier>
  <ServiceMetadataReferenceCollection>
    <ServiceMetadataReference href="http://serviceMetadata.eu/busdox-actorid-upis%3A%3A0010%3A5798000000001/services/bdx-docid-
qns%3A%3Aurn%3Aoasis%3Anames%3Aspecification%3Aubl%3Aschema%3Axsd%3AInvoice- 2%3A%3AInvoice%23%23UBL-2.0" />
  </ServiceMetadataReferenceCollection>
  <Extension>
    <ex:Test xmlns:ex="http://test.eu">Test</ex:Test>
  </Extension>
 </ServiceGroup>
```

Error codes

HTTP code	HTTP Message	Business code	Meaning
200	OK	n/a	The request was completed successfully
404	Resource not found	n/a	The requested information was not found
500	Server Internal Error	TECHNICAL	Some unexpected technical error occurred (detailed information is available in the response)

Audit

The following information must be audited for this service (more details under $2.6.5 – 'Auditing'):

- AccessTime
- Operation
- ParticipantIdentifier
- ParticipantIdentifierScheme
- IpAddress
- RequestHeader
- ResponseHeader
- ResponseText
- HTTP code

2.5.2. UC07 - Retrieve Service Metadata

2.5.2.1. Use case

Brief description

Obtain detailed definition about one specific service of a specific participant for all supported transport.
This service retrieves the SignedServiceMetadata according to the input quadruplet participantIdentifier+participantIndentifierScheme+documentIdentifier+documentIdentifierScheme.
Returns information from the Endpoint table.

Actors

User

Preconditions

The user application has previously resolved the address of the SMP from the DNS.

Referenced service group and required Service Meta data were previously defined by the receiver.

Basic flow event

Step

1 The user requests the detailed information of a receiver's service to the SMP

2 The SMP validates the request, retrieves the information from its configuration database and sends its as response to the user

3 The user receives the participant's service detailed information

4 Use case ends with success

Alternative flows

3a **Redirect**

3a1 The configuration refers to another SMP. The SMP returns the redirection information to the user

3a2

The user reinitiate the same request to that other SMP : restart use case at step 1

3a3 Use case ends

Exception flows

1a **SMP is not reachable**

1a1 The user receives a network connection error

1a2 Use case ends

2a **Request is not well formed (or any other business/technical error)**

2a1 The SMP replies with HTTP error "500 Server Internal Error" with details on the error allowing to identify the error in the request (cf. "Error codes" table below)

2a2 The receiver receives the error message

2a3 Use case ends

2b **ServiceGroup or ServiceMetadata is not defined**

2b1 The SMP replies with HTTP error "404 Resource not found"

2b2 The receiver receives the error message

2b3 Use case ends

2a2a **Multiple redirect**

2a2a1 The client receives redirect information for the 2nd time (and must ignore it)

2a2a2 Use case ends

Post conditions

Successful conditions

The user receives ServiceMetaData information for the requested receiver participant.

Failure conditions

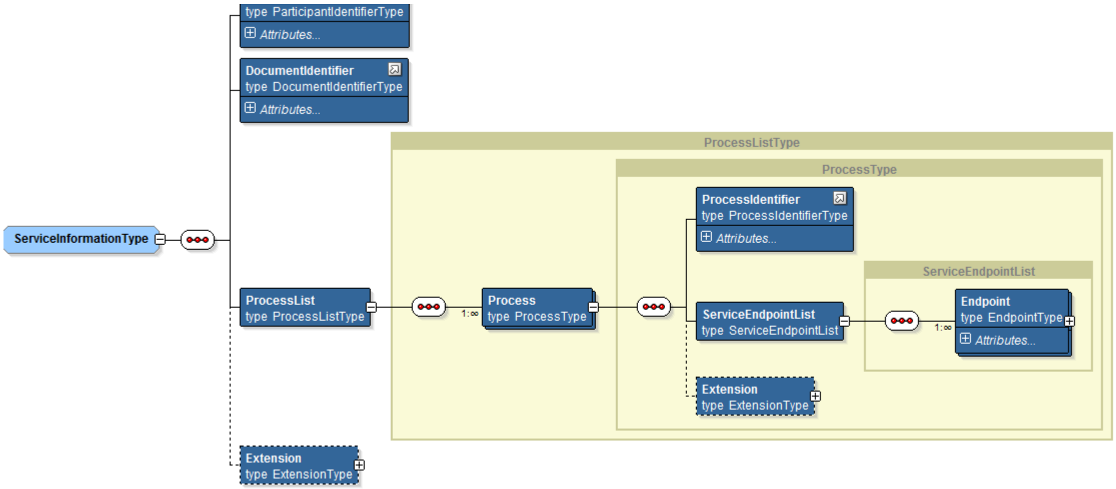
The user received no Metadata information about the requested receiver participant.

2.5.2.2. REST Service: GetSignedServiceMetadata

Input: *ServiceMetadataReference;* i.e. the PK made of 4 fields that uniquely identify the ServiceMetadata entry in the SMP configuration.

Execution:

This service will return necessary information for the user to send documents to the receiver, this information is held in the *ServiceInformation* structure; i.e. the information stored in tables Process and Endpoint (related to the requested service metadata and highlighted into red squares below):

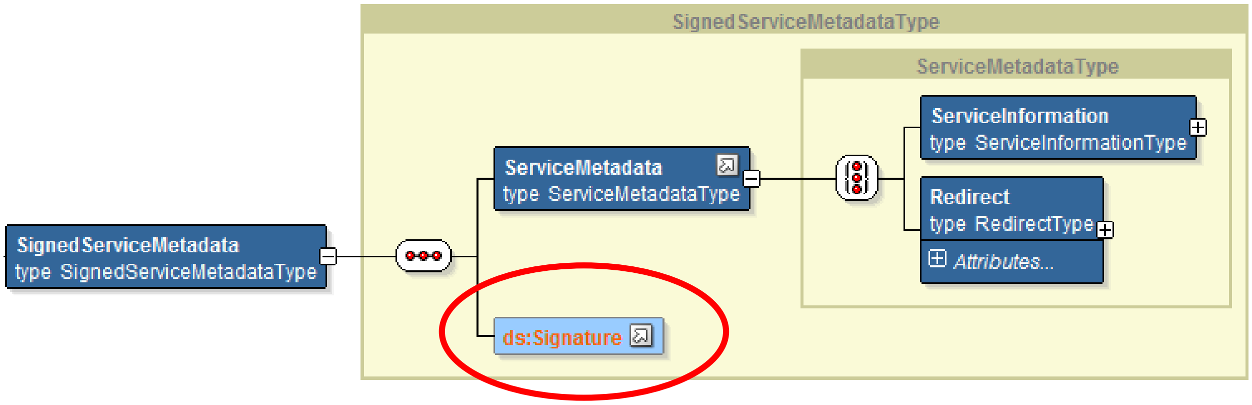


NB: there is no interaction with the SML.

Output: *SignedServiceMetadata*

Cf. [REF8], §4.3 : this data structure represents Metadata about a specific electronic service. The role of the ServiceMetadata structure is to associate a participant identifier with the ability to receive a specific document type over a specific transport. It also describes which business processes a document can participate in, and various operational data such as service activation and expiration times. The ServiceMetadata resource contains all the metadata about a service that a user Access Point needs to know in order to send a message to that service.

The SignedServiceMetadata structure holds both a *ServiceMetadata* structure and the corresponding signature by the receiver to allow the user (or any other user) verifying the authenticity of the information provided by the SMP by using the public key of the receiver before sending him any document.



NOTE: [REF11] / CR007: discusses the mandatory qualification of the Signature field.

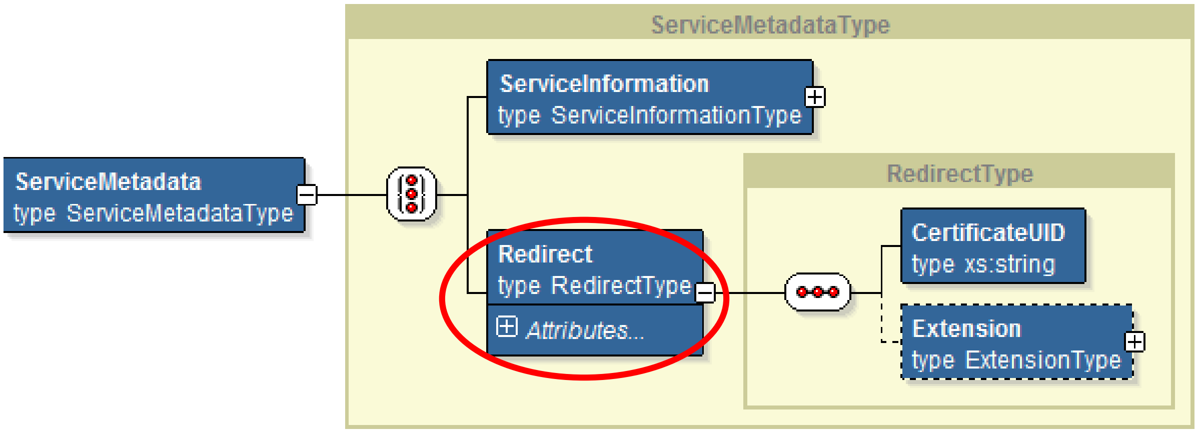
Output (alternative): Redirection (supports the alternative flow ‘a’ in the use case)

Eventually, this service will return *redirect* information instead of the *SignedServiceMetadata* information itself, when it is held by another SMP.

Redirection is exhaustively explained in [REF8] §4.3 ServiceMetadata and in [REF5] §2.1.3 Service Metadata Publisher Redirection.

In such a case, the information returned is the reference to the SMP that holds the corresponding "*ServiceMetadata*"; i.e. in the "Redirect" structure containing the target URI.

The queried SMP has in fact no information about the participant services (there is no related Process entry for that participant), instead, he has the target URI of the other SMP in the 'Redirect' column of the ServiceMetadata row for that receiver.



Sample Request

HTTP Header

```
GET http://130.206.118.4:8080/cipa-smp-full-webapp/iso6523-actorid-upis::0088:5798000000112/services/busdox-docid-qns::urn:oasis:names:specification:ubl:schema:xsd:Invoice-12::Invoice%23%23urn:www.cenbii.eu:transaction:biicoretrdm010:ver1.0:%23urn:www.peppol.eu:bis:peppol4a:ver1.0::2.0 HTTP/1.1
Accept-Encoding: gzip,deflate
Host: 130.206.118.4:8080
Connection: Keep-Alive
User-Agent: Apache-HttpClient/4.1.1 (java 1.5)
```

Text

N/A

Sample Response

HTTP header

```
HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Content-Type: text/xml
Transfer-Encoding: chunked
Date: Thu, 21 Jan 2016 10:22:38 GMT
Cache-Control: proxy-revalidate
Connection: Keep-Alive
```

Text

```
<SignedServiceMetadata xmlns="http://docs.oasis-open.org/bdxr/ns/SMP/2014/07">
  <ServiceMetadata
   xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
   <ServiceInformation>
     <ParticipantIdentifier scheme="busdox-actorid-upis">
      0010:5798000000001
     </ParticipantIdentifier>
     <DocumentIdentifier scheme="bdx-docid-qns">
      urn:oasis:names:specification:ubl:schema:xsd:Invoice-2::Invoice##UBL-2.02
     </DocumentIdentifier>
     <ProcessList>
      <Process>
        <ProcessIdentifier scheme="cenbii-procid-ubl">BII04
```

```
          </ProcessIdentifier>
          <ServiceEndpointList>
            <Endpoint transportProfile="busdox-transport-start">
              <EndpointURI>http://busdox.org/sampleService/</EndpointURI>
              <RequireBusinessLevelSignature>false
              </RequireBusinessLevelSignature>
              <MinimumAuthenticationLevel>2</MinimumAuthenticationLevel>
              <ServiceActivationDate>2009-05-01T09:00:00
              </ServiceActivationDate>
              <ServiceExpirationDate>2016-05-01T09:00:00
              </ServiceExpirationDate>
              <Certificate>AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA</Certificate>
              <ServiceDescription>invoice service</ServiceDescription>
              <TechnicalContactUrl>https://example.com
              </TechnicalContactUrl>
              <TechnicalInformationUrl>http://example.com/info
              </TechnicalInformationUrl>
            </Endpoint>
          </ServiceEndpointList>
        </Process>
        <Process>
          <ProcessIdentifier scheme="cenbii-procid-ubl">BII07
          </ProcessIdentifier>
          <ServiceEndpointList>
            <Endpoint transportProfile="busdox-transport-start">
              <EndpointURI>http://busdox.org/sampleService/</EndpointURI>
              <RequireBusinessLevelSignature>true
              </RequireBusinessLevelSignature>
              <MinimumAuthenticationLevel>1</MinimumAuthenticationLevel>
              <ServiceActivationDate>2009-05-01T09:00:00
              </ServiceActivationDate>
              <ServiceExpirationDate>2016-05-01T09:00:00
              </ServiceExpirationDate>
              <Certificate>AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA</Certificate>
              <ServiceDescription>invoice service</ServiceDescription>
              <TechnicalContactUrl>https://example.com
              </TechnicalContactUrl>
              <TechnicalInformationUrl>http://example.com/info
              </TechnicalInformationUrl>
              <Extension>
                <ex:Test xmlns:ex="http://test.eu">Test</ex:Test>
              </Extension>
            </Endpoint>
          </ServiceEndpointList>
          <Extension>
            <ex:Test xmlns:ex="http://test.eu">Test</ex:Test>
          </Extension>
        </Process>
      </ProcessList>
      <Extension>
        <ex:Test xmlns:ex="http://test.eu">Test</ex:Test>
       </Extension>
    </ServiceInformation>
  </ServiceMetadata>

   <Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
       <SignedInfo>
         <CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
         <SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
         <Reference URI="">
           <Transforms>
             <Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"/>
           </Transforms>
           <DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
           <DigestValue>6r3W426Gx5foBPtasSdIEj6JvAY=</DigestValue>
         </Reference>
       </SignedInfo>

<SignatureValue>2NJB0Pv3ORL+EpPYLCl/InXI+mDbUsV8CrWzRVJvEJMnnyuI2bPMe6k4MJwp9A4bTkzjvkMPARYAhyVNm6MNNlJRAFL4qdds
RrWa4Jgf/QF0zQgpJ7ZUPdVQ8L8A54FiPZWltOIgZCfO7sDbEcB00V4gKmzVPBsVu6BIBOws/UY=</SignatureValue>
       <KeyInfo>
         <X509Data>
```

```
<X509SubjectName>1.2.840.113549.1.9.1=#160e73656e64657240746573742e6265,CN=senderCN,OU=B4,O=DIGIT,L=Brussels,ST=BE,C=BE</
X509SubjectName>

<X509Certificate>MIICpTCCAg6gAwIBAgIBATANBgkqhkiG9w0BAQUFADB4MQswCQYDVQQGEwJCRTELMAkGA1UECAwCQkUxETAPBgNVBAc
MCEJydXNzZWxzMQ4wDAYDVQQKDAVESUdJVDELMAkGA1UECwwCQjQxDzANBgNVBAMMBnJvb3RDTjEbMBkGCSqGSIb3DQEJARYMcm9vd
EB0ZXN0LmJlMB4XDTE1MDMxNzE2MTkwN1oXDTI1MDMxNDE2MTkwN1owfDELMAkGA1UEBhMCQkUxCzAJBgNVBAgMAkJFMREwDwYDVQ
QHDAhCcnVzc2VsczEOMAwGA1UECgwFRElHSVQxCzAJBgNVBAsMAkI0MREwDwYDVQQDDAhzZW5kZXJDTjEdMBsGCSqGSIb3DQEJARYOc2Vu
ZGVyQHRlc3QuYmUwgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBANxLUPjIn7R0CsHf86kIwNzCu+6AdmWM8fBLUHL+VXT6ayr1kwgGbFM
b/vUUX6a46jRCiZBM+9IK1Hpjg9QX/QIQiWtvD+yDr6jUxahZ/w13kqFG/K81IVu9DwLBoiNwDvQ6l6UbvMvV+1nWy3gjRcKlFs/C+E2uybgJxSM/s
MkbAgMBAAGjOzA5MB8GA1UdIwQYMBaAFHCVSh4WnWR8MGBGedr+bJH96tc4MAkGA1UdEwQCMAAwCwYDVR0PBAQDAgTwMA0GCSqG
SIb3DQEBBQUAA4GBAK6idNRxyeBmqPoSKxq7Ck3ej6R2QPyWbwZ+6/S7iCRt8PfgOu++Yu5YEjlUX1hlkbQKF/JuKTLqxNnKIE6Ef65+JP2ZaI9O2w
dzpRclAhAd00XbNKpyipr4jMdWmu2U8vyBBwn/utG1ZrLhAUiqnPvmaQrResiGHM2xzCmVwtse</X509Certificate>
          </X509Data>
        </KeyInfo>
      </Signature>

 </SignedServiceMetadata>
```

Sample Response (redirect alternative)

HTTP header

```
HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Content-Type: text/xml
Transfer-Encoding: chunked
Date: Thu, 21 Jan 2016 10:22:38 GMT
Cache-Control: proxy-revalidate
Connection: Keep-Alive
```

Text

```
<?xml version="1.0" encoding="utf-8" ?>

<SignedServiceMetadata
      xmlns="http://docs.oasis-open.org/bdxr/ns/SMP/2014/07">
      <ServiceMetadata>
        <Redirect
          href="http://serviceMetadata2.eu/busdox-actorid-upis%3A%3A0010%3A5798000000001/services/bdx-docid-
         qns%3A%3Aurn%3Aoasis%3Anames%3Aspecification%3Aubl%3Aschema%3Axsd%3AInvoice- 2%3A%3AInvoice%23%23UBL-2.0">
          <CertificateUID>PID:9208-2001-3-279815395</CertificateUID>
          <Extension>
            <ex:Test xmlns:ex="http://test.eu">Test</ex:Test>
          </Extension>
        </Redirect>
      </ServiceMetadata>

    <Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
        <SignedInfo>
          <CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
          <SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
          <Reference URI="">
            <Transforms>
              <Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"/>
            </Transforms>
            <DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
            <DigestValue>6r3W426Gx5foBPtasSdIEj6JvAY=</DigestValue>
          </Reference>
        </SignedInfo>

        <SignatureValue>2NJB0Pv3ORL+EpPYLCl/InXI+mDbUsV8CrWzRVJvEJMnnyuI2bPMe6k4MJwp9A4bTkzjvkMPARYAhyVNm6MNNlJRAFL4qdd
        sRrWa4Jgf/QF0zQgpJ7ZUPdVQ8L8A54FiPZWltOIgZCfO7sDbEcB00V4gKmzVPBsVu6BIBOws/UY=</SignatureValue>
        <KeyInfo>
          <X509Data>

        <X509SubjectName>1.2.840.113549.1.9.1=#160e73656e64657240746573742e6265,CN=senderCN,OU=B4,O=DIGIT,L=Brussels,ST=BE,C=B
        E</X509SubjectName>

        <X509Certificate>MIICpTCCAg6gAwIBAgIBATANBgkqhkiG9w0BAQUFADB4MQswCQYDVQQGEwJCRTELMAkGA1UECAwCQkUxETAPBgNVB
    AcMCEJydXNzZWxzMQ4wDAYDVQQKDAVESUdJVDELMAkGA1UECwwCQjQxDzANBgNVBAMMBnJvb3RDTjEbMBkGCSqGSIb3DQEJARYMcm
    9vdEB0ZXN0LmJlMB4XDTE1MDMxNzE2MTkwN1oXDTI1MDMxNDE2MTkwN1owfDELMAkGA1UEBhMCQkUxCzAJBgNVBAgMAkJFMREwDw
    YDVQQHDAhCcnVzc2VsczEOMAwGA1UECgwFRElHSVQxCzAJBgNVBAsMAkI0MREwDwYDVQQDDAhzZW5kZXJDTjEdMBsGCSqGSIb3DQEJAR
    YOc2VuZGVyQHRlc3QuYmUwgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBANxLUPjIn7R0CsHf86kIwNzCu+6AdmWM8fBLUHL+VXT6ayr1
    kwgGbFMb/vUUX6a46jRCiZBM+9IK1Hpjg9QX/QIQiWtvD+yDr6jUxahZ/w13kqFG/K81IVu9DwLBoiNwDvQ6l6UbvMvV+1nWy3gjRcKlFs/C+E2
    uybgJxSM/sMkbAgMBAAGjOzA5MB8GA1UdIwQYMBaAFHCVSh4WnWR8MGBGedr+bJH96tc4MAkGA1UdEwQCMAAwCwYDVR0PBAQDAg
    TwMA0GCSqGSIb3DQEBBQUAA4GBAK6idNRxyeBmqPoSKxq7Ck3ej6R2QPyWbwZ+6/S7iCRt8PfgOu++Yu5YEjlUX1hlkbQKF/JuKTLqxNnKIE6E
    f65+JP2ZaI9O2wdzpRclAhAd00XbNKpyipr4jMdWmu2U8vyBBwn/utG1ZrLhAUiqnPvmaQrResiGHM2xzCmVwtse</X509Certificate>
      </X509Data>
    </KeyInfo>
  </Signature>

</SignedServiceMetadata>
```

Error codes

HTTP code	HTTP Message	Business code	Meaning
200	OK	n/a	The request was completed successfully
404	Resource not found	n/a	The requested information was not found
500	Server Internal Error	TECHNICAL	Some unexpected technical error occurred (detailed information is available in the response)

Audit

The following information must be audited for this service (more details under $2.6.5 – 'Auditing'):

- AccessTime
- Operation
- ParticipantIdentifier
- ParticipantIdentifierScheme
- DocumentIdentifier
- DocumentIdentifierScheme
- IpAddress
- RequestHeader
- ResponseHeader
- ResponseText
- HTTP code

2.6. Security

2.6.1. User management

2.6.1.1. Administration process

As described in §2.3.1 – “Actors”, there will be 3 types of users accessing the SMP. Among them, only “Admin ServiceGroup” and “Admin SMP” types of users will be registered into the configuration of the SMP.

This paragraph summarizes the process for defining the users who are responsible for managing the overall configuration of SMPs.

1. **Creation of an SMP Admin**

   The "System Admin" creates an "Admin SMP" user in the "User" table of the SMP. The password is stored along the username (cf. §2.6.1.4 – "Security tables").

   In the picture below, "System Admin b" creates one user "Admin SMP b" that will manage the service groups on this SMP's.

2. **Creation of a ServiceGroup administrator (for one Participant)**

   The "Admin SMP" deploys the certificates that will be used to access the SMP for a new participant's administration (if certificates are used).
   The "Admin SMP" user accesses the SMP via http with basic authentication with the previously assigned username and password by the "System Admin".
   He uses "UC02 - Create or Update Service Group" (cf. §2.4.2) to define new service groups.

   When doing so, the "Admin SMP" provides either
   - A "*CertificateIdentifier*"; i.e. some pieces of the Participant's certificates that will be used to identify the "Admin ServiceGroup" user accessing the SMP for configuration purposes; (mostly for distributed SMP model)
   - Nothing: in that case, the basic authentication information of the "Admin SMP" (in the HTTP header) will be stored as identifier, and will be himself the administrator of this ServiceGroup (cf. Step 1 of UC02 - Create or Update Service Group).

   Later, he will be able to remove that Service Group via the same access method using "UC03 - Erase Service Group" (cf. §2.4.3).

   In the picture below, "*Admin SMP b*" creates one user "*Admin ServiceGroup D,E,F*" that will manage parties D,E and F.

3. **Creation of ServiceMetadata**

   The "Admin ServiceGroup" accesses the SMP using its certificate.

   He defines some new services using "UC04 - Create or Update Service Metadata" (cf. §2.4.4).

   He later will be able to remove deprecated services similarly with

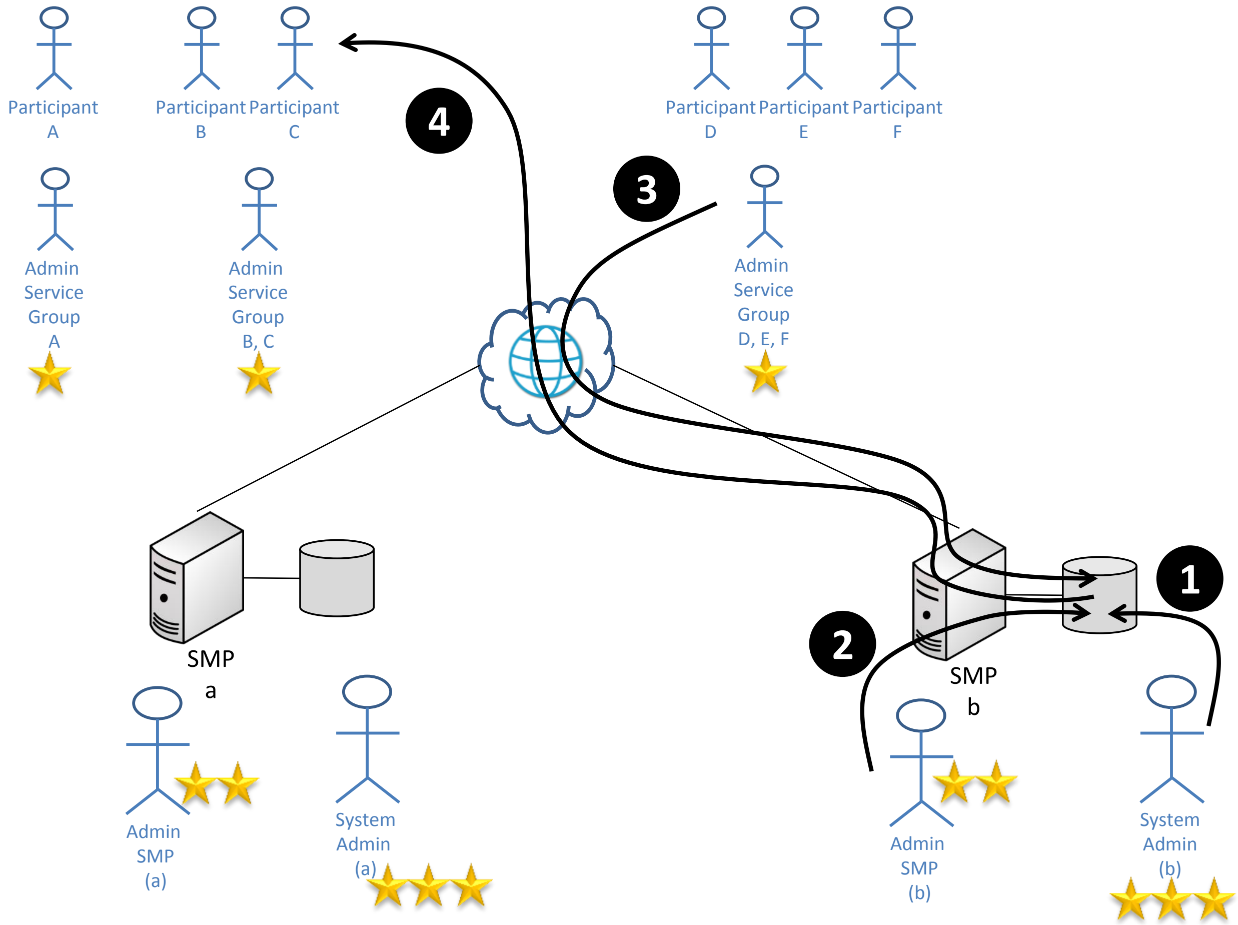
REST Service: DeleteServiceMetadata (cf. §2.4.5 - "UC05 - Erase Service Metadata").

In the picture below, "*Admin ServiceGroup D, E, F*" defines some of the services for one or several parties among D, E and F.

4. **Discovering a participant's services capabilities**

The Participant access the SMP with no authentication.
He uses "UC06 - Retrieve Service Group" (cf. §2.5.1) and "UC07 - Retrieve Service Metadata" (cf. §2.5.2) to collect eDelivery information on another participant he wants to exchange messages with.

In the picture below, "*Participant C* " collects metadata from one (and only one) participant among D, E and F.



2.6.1.2. Simple User

The regular users (Actor "User") are any user accessing the system's public services. As these users don't need to be authenticated, they don't have to be known in advance by the System and are therefore not preregistered in any way on the SMP.

2.6.1.3. System Admin

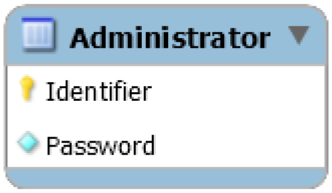
The "System Admin" actor is, as the name suggests, a system user having special accesses to the system. In the purpose of user administration for the SMP, this "system user" should be able to modify the content of the SMP configuration database, i.e. he must have full read/write data access on this configuration database, in particular table "User" described in §2.6.1.4 "Security tables".

He will be in charge of creating and maintaining the access rights for all "Admin SMP" users (as described by use case UC01).

2.6.1.4. Security tables

2.6.1.4.1. Administrator

This table identifies the **administrators** of the SMP; i.e. "*Admin SMP*" and "*Admin ServiceGroup*" actors introduced above.



There are two possible means to obtain access to the SMP non-public services:

- through **basic authentication**; i.e. with a simple **username/password** authentication method:
    - **Identifier** column contains then the username used to identify the user at logon
    - **Password** column contains then the hash of the password used to authenticate the user at logon

- thru **two-way SSL** using PKI infrastructure (i.e. X.509 certificates):
    - the **Identifier** column contains pieces of the client certificate that are forwarded by the reverse proxy in the http header to the server (cf. 2.6.3 – "HTTP Authentication")
    - **Password** column is unused for 2-way-ssl since the certificate is not validated by the application layer itself; the prerequisite being that the user's certificate is already present in the truststore of the reverse proxy server.

In all cases, it is the responsibility of the SMP to hash the password (and apply the same algorithm for authentication). The participant will send the password in 'clear' in the XSD content.

2.6.1.4.2. Ownership (of service group)

1-N relationship materialization between the service groups and the "Admin ServiceGroup" type of users of the SMP. More details are available under §2.6.1.6 – "Admin ServiceGroup".

This relationship allows the system to identify which 'user' (singular) is allowed to modify(/delete) all the information related to all the ServiceMetadata of one given '*ServiceGroup*'.

2.6.1.5. Admin SMP

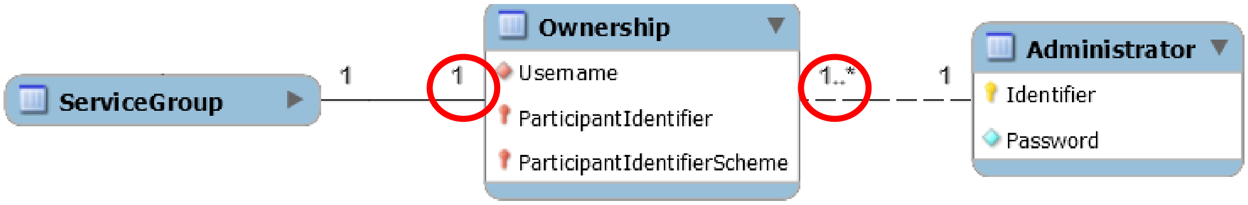
The "Admin SMP" user is created by the system administrator (cf. §2.3.1 – "Actors" and §2.4.1 – "UC01 - Manage Administrators").

Some information in the system (not detailed here) allows the system to identify this specificity of such users.

2.6.1.6. Admin ServiceGroup

The "Admin ServiceGroup" user of one specific participant will be allowed to use all the services that modify the definition of the ServiceGroups; i.e. to create, modify or delete ServiceGroup definitions.

To allow the access right verification, the configuration holds a link between the "Admin ServiceGroup" and the related ServiceGroup via an "ownership relationship" materialized as shown here in the configuration:



One and only one "Admin ServiceGroup" user will be allowed for managing each ServiceGroup and will be the ONLY user allowed to do so. Such administrator may though manage multiple ServiceGroups.

This link is established when the ServiceGroup is created (or updated).

2.6.2. Access rights

The following matrix clarifies the access rights of each actors to all use cases and the type of authentication method that are supported for each user role:

		System Admin	Admin SMP	Admin Service Group	User
UC01	**Manage Administrators**	X			
UC02	**Create or Update Service Group**		X		
UC03	**Erase Service Group**		X		
UC04	**Create or Update Service Metadata**		X	X	
UC05	**Erase Service Metadata**		X	X	
UC06	**Retrieve Service Group**	X	X	X	X
UC07	**Retrieve Service Metadata**	X	X	X	X

Authentication method (Acceptance and Production at EC)

System + database authentication	X			
HTTP Basic authentication		X	-	
HTTP 2-way-ssl			X	
None				X

Authentication method (Test at EC)

System + database authentication	X			
HTTP Basic authentication		X	X	
HTTP 2-way-ssl			-	
None				X

NB: beware: "Admin SMP" user may act on behalf of all the "Admin ServiceGroups" defined in the SMP.

2.6.3. HTTP Authentication

SSL will be used at all time (i.e. for any exchange of message between a SMP and any participant, acting as a sender or as a receiver.) to guarantee the validity of the information provided by the SMP to the sender and receiver.

Two authentication methods are supported and vary with services and/or user's roles:

1. Basic HTTP authentication (username/password) – for “Admin SMP” users and optionally for “Admin ServiceGroup” users ;
2. HTTP 2-way SSL for “Admin ServiceGroup” users (only) when and if this method is preferred for those to basic authentication (see "Authentication method" tables in §2.6.2 above).

If HTTP basic authentication is available for both types of users, 2-way SSL will also be usable for authenticating “Admin ServiceGroup” users. In order to achieve this, all the PUT and DELETE services on ServiceMetadata data type (cf. UC04 and UC05) will be able to use that type of authentication.

In order to provide this possibility, the certificates of the authorized administrators (“Admin ServiceGroup” users) will be deployed on the necessary SMPs on dedicated keystores. This will allow the transport layers to establish necessary trust without any addition to the existing message structure.

Also, the fields in *User* table will be used as follows differently in the different possible cases (by user roles and authentication methods):

User role:	**Admin SMP**	**Admin ServiceGroup**	
Authentication type:	**Basic Authentication**	**2 way-ssl**	**Basic Authentication**

Username:	Basic username	HTTP client cert	Basic username
Password:	password hash	n/a	password hash

NB. Only basic authentication is allowed for “Admin SMP” user since they are intended to be “intranet” users rather than “internet” ones.

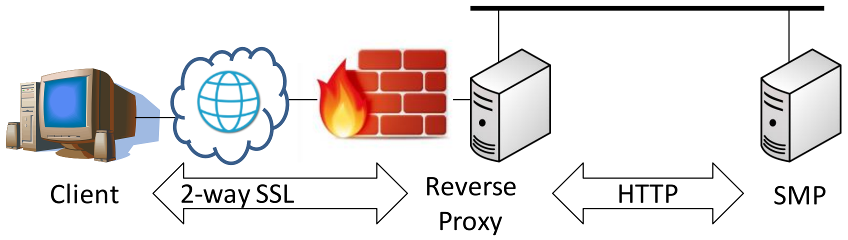
The password field, when applicable, will hold a hash value of the password.

2.6.4. Reverse proxy

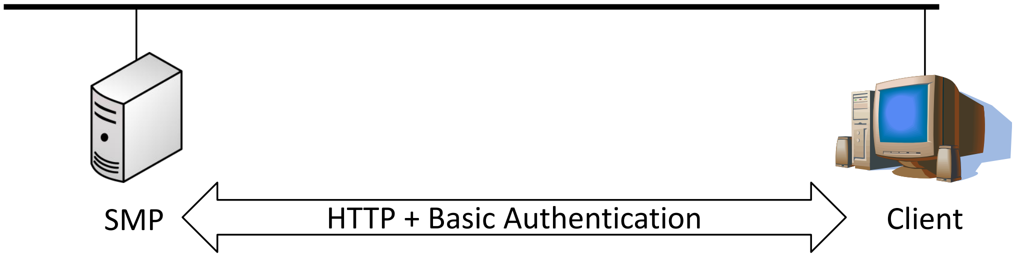
This paragraph discusses the specific deployment at the European Commission for information only.

An existing BDMSL server that is already hosted at the European Commission behind a “Reverse Proxy” as explained in [REF12] §11.2.2 “*Reverse proxy with SSL*”. In this case, 2-way SSL is set up on the reverse proxy and the application server hosting the application can use the HTTP protocol.

A similar configuration could be used at the European Commission for SMP’s where 2-way SSL must be used.



As stated above, this type of access will be provided for “Admin ServiceGroup” type of users only, and is optional; basic authentication also allowed in such a case:



As a consequence, the authentication mechanism for services modifying Service Metadata will behave as follow:

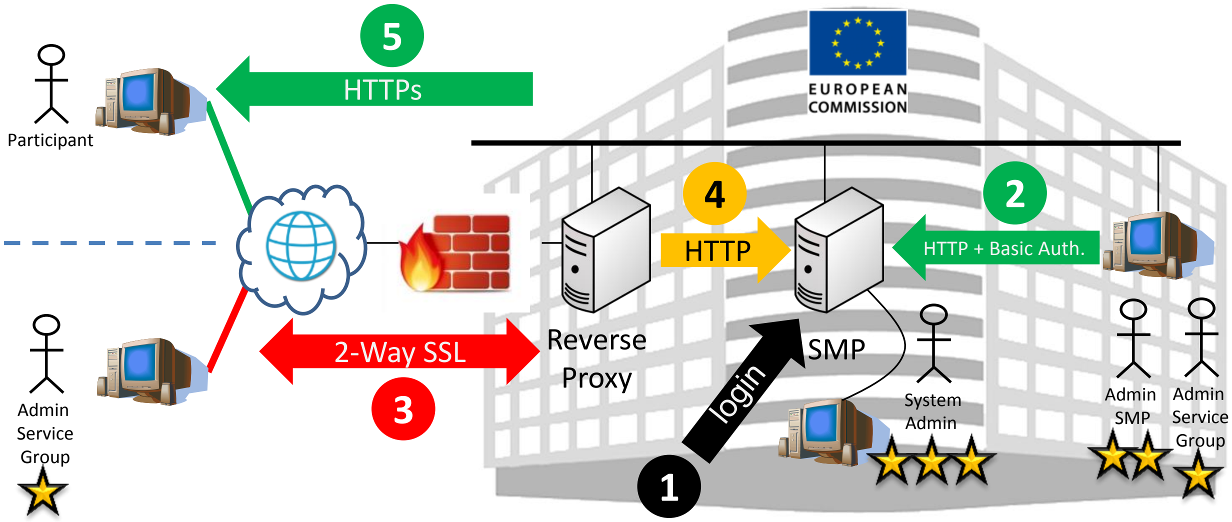
- Search HTTP header for “Client Certificate” data (conversion performed by the reverse proxy). If present, use these to authenticate user against the “username” present in table “User”.
  The “Client Certificate” values will be inserted in the HTTP header to the SMP by the Reverse Proxy out of the X.509 Certificate.

  The X.509 attributes to be used will be defined in the detailed design.

  The value stored in the “User” table column “username” should contain necessary information to validate that the provided values match.
- If no “Client certificate” information is available (meaning there is no reverse proxy between the client and the SMP), use Basic HTTP authentication: check provided username and password (clear value) to identify and authenticate the requesting user and authorize access.

In summary, with such configuration, accesses will be the following for SMP deployed at the European Commission:

1. Direct System & database logins are used by the System Admin.
2. Basic authentication over HTTP is used for the Admin SMP and Admin Service Group's that are on the same local network than the SMP itself.
   SMP authenticates local Admin SMP's based on the hash of the password was stored by the System admin.
3. Certificates of remote "*Admin Service Group*'s" are authenticated by the Reverse Proxy.
4. Information of the client's certificate is provided to the SMP for authorization (*Client-Cert* attribute) – password is blank
5. Parties don't have to authenticate themselves, but may use the SMP's certificate to authenticate it.


5
HTTPs
Participant
EUROPEAN COMMISSION
4
HTTP
2
HTTP + Basic Auth.
2-Way SSL
3
Reverse Proxy
SMP
login
1
System Admin
Admin SMP
Admin Service Group
Admin Service Group

2.6.5. Auditing

All SMP services will log relevant information regarding the access as specified in the table below:

		Manage Administrators	Create or Update Service Group	Erase Service Group	Create or Update Service Metadata	Erase Service Metadata	Retrieve Service Group	Retrieve Service Metadata
Column	**Description**	**UC01**	**UC02**	**UC03**	**UC04**	**UC05**	**UC06**	**UC07**
AdministratorIdentifier	Whom the request was initiated from	n/a	X	X	X	X	-	-
AccessTime	When the access was made	n/a	X	X	X	X	X	X
Operation	What was performed (servicename)	n/a	X	X	X	X	X	X
ParticipantIdentifier	The identifier of the participant	n/a	X	X	X	X	X	X
ParticipantIdentifierScheme	The scheme of the identifier of the participant	n/a	X	X	X	X	X	X
DocumentIdentifier	The identifier of the document	n/a	-	-	X	X	-	X
DocumentIdentifierScheme	The scheme of the identifier of the document	n/a	-	-	X	X	-	X
IpAddress	The source IP address from which the request was initiated	n/a	X	X	X	X	X	X
RequestHeader	The HTTP Header of the request	n/a	X	X	X	X	X	X
RequestText	The text of the request (XML)	n/a	X	-	X	-	-	-
ResponseHeader	The HTTP Header of the response	n/a	X	X	X	X	X	X
ResponseText	The text of the response (XML)	n/a	-	-	-	-	X	X
HTTP code	The HTTP response code	n/a	X	X	X	X	X	X
Business code	The application level error code for HTTP error 500	n/a	X	X	X	X	-	-
ErrorDescription	The description of the error (free text)	n/a	X	X	X	X	-	-

It will be a design decision to save this auditing information either in a database table, log files or any type of persistence solution provided that the information is saved and is searchable.

Audited information must be kept accessible (online or offline) during at least 3 months.

No hard link (with foreign keys) will be established between this table and the User or the participant identifier one to allow:

- Keeping the logs relating to one user or one participant that is later removed from the database (if ever applicable);
- Keeping track of unauthorized calls for unidentified users or erroneous participant identifications.

2.7. Special requirements

- The SMP should be available 99%.
- Response time should be less than 5s for the GET services for 90% of the requests
- Response time should be less than 10s for the PUT/DELETED services for 90% of the requests

3. CONTACT INFORMATION

CEF Support Team

By email: CEF-EDELIVERY-SUPPORT@ec.europa.eu

By phone: +32 2 299 09 09

- Standard Service: 8am to 6pm (Normal EC working Days)
- Standby Service*: 6pm to 8am (Commission and Public Holidays, Weekends)

* *Only for critical and urgent incidents and only by phone*

4. Annex

4.1. XSD files

4.1.1. *Original official OASIS SMP XSD*

Reference: https://docs.oasis-open.org/bdxr/bdx-smp/v1.0/cs01/schemas/bdx-smp-201407.xsd

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
<!--

   Service Metadata Publishing (SMP) Version 1.0
   Committee Specification 01
   18 December 2014
   Copyright (c) OASIS Open 2014. All Rights Reserved.
   Source: http://docs.oasis-open.org/bdxr/bdx-smp/v1.0/cs01/schemas/
-->
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns="http://docs.oasis-
open.org/bdxr/ns/SMP/2014/07" xmlns:ds="http://www.w3.org/2000/09/xmldsig#"xmlns:wsa="http://www.w3.org/2005/08/
addressing" elementFormDefault="qualified" targetNamespace="http://docs.oasis-
open.org/bdxr/ns/SMP/2014/07" id="ServiceMetadataPublishing">
<xs:import namespace="http://www.w3.org/2000/09/xmldsig#" schemaLocation="http://www.w3.org/TR/xmldsig-
core/xmldsig-core-schema.xsd"/>
<xs:import namespace="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd"/>
<xs:element name="ServiceGroup" type="ServiceGroupType"/>
<xs:element name="ServiceMetadata" type="ServiceMetadataType"/>
<xs:element name="SignedServiceMetadata" type="SignedServiceMetadataType"/>
<xs:complexType name="SignedServiceMetadataType">
<xs:sequence>
<xs:element ref="ServiceMetadata"/>
<xs:element ref="ds:Signature"/>
</xs:sequence>
</xs:complexType>
<xs:complexType name="ServiceMetadataType">
<xs:choice>
<xs:element name="ServiceInformation" type="ServiceInformationType"/>
<xs:element name="Redirect" type="RedirectType"/>
</xs:choice>
</xs:complexType>
<xs:complexType name="ServiceInformationType">
<xs:sequence>
<xs:element ref="ParticipantIdentifier"/>
<xs:element ref="DocumentIdentifier"/>
<xs:element name="ProcessList" type="ProcessListType"/>
<xs:element name="Extension" type="ExtensionType" minOccurs="0"/>
</xs:sequence>
</xs:complexType>
<xs:complexType name="ProcessListType">
<xs:sequence>
<xs:element name="Process" type="ProcessType" maxOccurs="unbounded"/>
</xs:sequence>
</xs:complexType>
<xs:complexType name="ProcessType">
<xs:sequence>
<xs:element ref="ProcessIdentifier"/>
<xs:element name="ServiceEndpointList" type="ServiceEndpointList"/>
<xs:element name="Extension" type="ExtensionType" minOccurs="0"/>
</xs:sequence>
</xs:complexType>
<xs:complexType name="ServiceEndpointList">
```

```
<xs:sequence>
<xs:element name="Endpoint" type="EndpointType" maxOccurs="unbounded"/>
</xs:sequence>
</xs:complexType>
<xs:complexType name="EndpointType">
<xs:sequence>
<xs:element name="EndpointURI" type="xs:anyURI"/>
<xs:element name="RequireBusinessLevelSignature" type="xs:boolean"/>
<xs:element name="MinimumAuthenticationLevel" type="xs:string" minOccurs="0"/>
<xs:element name="ServiceActivationDate" type="xs:dateTime" minOccurs="0"/>
<xs:element name="ServiceExpirationDate" type="xs:dateTime" minOccurs="0"/>
<xs:element name="Certificate" type="xs:base64Binary"/>
<xs:element name="ServiceDescription" type="xs:string"/>
<xs:element name="TechnicalContactUrl" type="xs:anyURI"/>
<xs:element name="TechnicalInformationUrl" type="xs:anyURI" minOccurs="0"/>
<xs:element name="Extension" type="ExtensionType" minOccurs="0"/>
</xs:sequence>
<xs:attribute name="transportProfile" type="xs:string" use="required"/>
</xs:complexType>
<xs:complexType name="ServiceGroupType">
<xs:sequence>
<xs:element ref="ParticipantIdentifier"/>
<xs:element name="ServiceMetadataReferenceCollection" type="ServiceMetadataReferenceCollectionType"/>
<xs:element name="Extension" type="ExtensionType" minOccurs="0"/>
</xs:sequence>
</xs:complexType>
<xs:complexType name="ServiceMetadataReferenceCollectionType">
<xs:sequence>
<xs:element name="ServiceMetadataReference" type="ServiceMetadataReferenceType" minOccurs="0" maxOccurs="unbound
ed"/>
</xs:sequence>
</xs:complexType>
<xs:complexType name="ServiceMetadataReferenceType">
<xs:attribute name="href" type="xs:anyURI"/>
</xs:complexType>
<xs:complexType name="RedirectType">
<xs:sequence>
<xs:element name="CertificateUID" type="xs:string"/>
<xs:element name="Extension" type="ExtensionType" minOccurs="0"/>
</xs:sequence>
<xs:attribute name="href" type="xs:anyURI" use="required"/>
</xs:complexType>
<xs:complexType name="ExtensionType">
<xs:sequence>
<xs:any/>
</xs:sequence>
</xs:complexType>
<xs:element name="ParticipantIdentifier" type="ParticipantIdentifierType"/>
<xs:element name="DocumentIdentifier" type="DocumentIdentifierType"/>
<xs:element name="ProcessIdentifier" type="ProcessIdentifierType"/>
<xs:element name="RecipientIdentifier" type="ParticipantIdentifierType"/>
<xs:element name="SenderIdentifier" type="ParticipantIdentifierType"/>
<xs:complexType name="ParticipantIdentifierType">
<xs:simpleContent>
<xs:extension base="xs:string">
<xs:attribute name="scheme" type="xs:string"/>
</xs:extension>
</xs:simpleContent>
</xs:complexType>
<xs:complexType name="DocumentIdentifierType">
<xs:simpleContent>
<xs:extension base="xs:string">
<xs:attribute name="scheme" type="xs:string"/>
</xs:extension>
</xs:simpleContent>
```

```
</xs:complexType>
<xs:complexType name="ProcessIdentifierType">
<xs:simpleContent>
<xs:extension base="xs:string">
<xs:attribute name="scheme" type="xs:string"/>
</xs:extension>
</xs:simpleContent>
</xs:complexType>
</xs:schema>
```

4.1.2. *Extended SMP XSD for standard services (GET)*

These paragraph shows the above XSD with extend fields definitions to support new requirements:

- In order for the SMP response to be valid (the Extension element is causing issues), the attribute "processContents" must be added to the ExtensionType with the value "lax" (or "skip").

  Proposed change (bolded text):

```
<xs:complexType name="ExtensionType">
   <xs:sequence>

      <xs:any processContents="lax"/>

   </xs:sequence>
</xs:complexType>
```

NB: cf. [REF11] / CR003.

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
<!--

   Service Metadata Publishing (SMP) Version 1.0
   Committee Specification 01
   18 December 2014
   Copyright (c) OASIS Open 2014. All Rights Reserved.
   Source: http://docs.oasis-open.org/bdxr/bdx-smp/v1.0/cs01/schemas/
-->
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns="http://docs.oasis-
open.org/bdxr/ns/SMP/2014/07" xmlns:ds="http://www.w3.org/2000/09/xmldsig#"xmlns:wsa="http://www.w3.org/2005/08/
addressing" elementFormDefault="qualified" targetNamespace="http://docs.oasis-
open.org/bdxr/ns/SMP/2014/07" id="ServiceMetadataPublishing">
<xs:import namespace="http://www.w3.org/2000/09/xmldsig#" schemaLocation="http://www.w3.org/TR/xmldsig-
core/xmldsig-core-schema.xsd"/>
<xs:import namespace="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd"/>
<xs:element name="ServiceGroup" type="ServiceGroupType"/>
<xs:element name="ServiceMetadata" type="ServiceMetadataType"/>
<xs:element name="SignedServiceMetadata" type="SignedServiceMetadataType"/>
<xs:complexType name="SignedServiceMetadataType">
<xs:sequence>
<xs:element ref="ServiceMetadata"/>
<xs:element ref="ds:Signature"/>
</xs:sequence>
</xs:complexType>
<xs:complexType name="ServiceMetadataType">
<xs:choice>
<xs:element name="ServiceInformation" type="ServiceInformationType"/>
<xs:element name="Redirect" type="RedirectType"/>
</xs:choice>
</xs:complexType>
<xs:complexType name="ServiceInformationType">
<xs:sequence>
<xs:element ref="ParticipantIdentifier"/>
<xs:element ref="DocumentIdentifier"/>
<xs:element name="ProcessList" type="ProcessListType"/>
<xs:element name="Extension" type="ExtensionType" minOccurs="0"/>
</xs:sequence>
</xs:complexType>
<xs:complexType name="ProcessListType">
<xs:sequence>
<xs:element name="Process" type="ProcessType" maxOccurs="unbounded"/>
</xs:sequence>
</xs:complexType>
<xs:complexType name="ProcessType">
<xs:sequence>
<xs:element ref="ProcessIdentifier"/>
<xs:element name="ServiceEndpointList" type="ServiceEndpointList"/>
<xs:element name="Extension" type="ExtensionType" minOccurs="0"/>
</xs:sequence>
</xs:complexType>
<xs:complexType name="ServiceEndpointList">
<xs:sequence>
<xs:element name="Endpoint" type="EndpointType" maxOccurs="unbounded"/>
</xs:sequence>
</xs:complexType>
<xs:complexType name="EndpointType">
<xs:sequence>
<xs:element name="EndpointURI" type="xs:anyURI"/>
<xs:element name="RequireBusinessLevelSignature" type="xs:boolean"/>
<xs:element name="MinimumAuthenticationLevel" type="xs:string" minOccurs="0"/>
<xs:element name="ServiceActivationDate" type="xs:dateTime" minOccurs="0"/>
```

```
<xs:element name="ServiceExpirationDate" type="xs:dateTime" minOccurs="0"/>
<xs:element name="Certificate" type="xs:base64Binary"/>
<xs:element name="ServiceDescription" type="xs:string"/>
<xs:element name="TechnicalContactUrl" type="xs:anyURI"/>
<xs:element name="TechnicalInformationUrl" type="xs:anyURI" minOccurs="0"/>
<xs:element name="Extension" type="ExtensionType" minOccurs="0"/>
</xs:sequence>
<xs:attribute name="transportProfile" type="xs:string" use="required"/>
</xs:complexType>
<xs:complexType name="ServiceGroupType">
<xs:sequence>
<xs:element ref="ParticipantIdentifier"/>
<xs:element name="ServiceMetadataReferenceCollection" type="ServiceMetadataReferenceCollectionType"/>
		<xs:element name="Extension" type="ExtensionType" minOccurs="0"/>
</xs:sequence>
</xs:complexType>
<xs:complexType name="ServiceMetadataReferenceCollectionType">
<xs:sequence>
<xs:element name="ServiceMetadataReference" type="ServiceMetadataReferenceType" minOccurs="0" maxOccurs="unbounded"/>
</xs:sequence>
</xs:complexType>
<xs:complexType name="ServiceMetadataReferenceType">
<xs:attribute name="href" type="xs:anyURI"/>
</xs:complexType>
<xs:complexType name="RedirectType">
<xs:sequence>
<xs:element name="CertificateUID" type="xs:string"/>
<xs:element name="Extension" type="ExtensionType" minOccurs="0"/>
</xs:sequence>
<xs:attribute name="href" type="xs:anyURI" use="required"/>
</xs:complexType>
<xs:complexType name="ExtensionType">
<xs:sequence>
		<xs:any processContents="lax"/>
</xs:sequence>
</xs:complexType>
<xs:element name="ParticipantIdentifier" type="ParticipantIdentifierType"/>
<xs:element name="DocumentIdentifier" type="DocumentIdentifierType"/>
<xs:element name="ProcessIdentifier" type="ProcessIdentifierType"/>
<xs:element name="RecipientIdentifier" type="ParticipantIdentifierType"/>
<xs:element name="SenderIdentifier" type="ParticipantIdentifierType"/>
<xs:complexType name="ParticipantIdentifierType">
<xs:simpleContent>
<xs:extension base="xs:string">
<xs:attribute name="scheme" type="xs:string"/>
</xs:extension>
</xs:simpleContent>
</xs:complexType>
<xs:complexType name="DocumentIdentifierType">
<xs:simpleContent>
<xs:extension base="xs:string">
<xs:attribute name="scheme" type="xs:string"/>
</xs:extension>
</xs:simpleContent>
</xs:complexType>
<xs:complexType name="ProcessIdentifierType">
<xs:simpleContent>
<xs:extension base="xs:string">
<xs:attribute name="scheme" type="xs:string"/>
</xs:extension>
</xs:simpleContent>
</xs:complexType>
</xs:schema>
```

4.1.3. Extended SMP XSD for ADMIN services (PUT)

This paragraph shows the above XSD that defines the structures needed for the administration of the SMP specifying the input for text PUT REST services (i.e. for use cases "UC02 - Create or Update Service Group" and "UC04 - Create or Update Service Metadata")

This one is based on the standard one, with some updates on the structure related to the specific nature of the administration functionalities.

In summary:

- ServiceGroup type, used as input by UC02 has been modified as follows:
    - Does not contain the identifier of the Participant (since it is in the URL)
    - Contains additional credentials information to allow authorization of the "Admin ServiceGroup" user
    - Does not contain any reference data since these are returned in the GET operation, but are to be populated through the PUT ServiceMedata service not the PUT ServiceGroup one.
- ErrorResponse used as response has been added in order to allow returning some detailed information on the error that as occurred (extend 500 HTML error code)
- ServiceMetadata, used as input by UC04 :
    - Does not contain the identifiers of the Participant and of the Document (since they are in the URL):

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="http://docs.oasis-open.org/bdxr/ns/SMP/2014/07"
  xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
  xmlns:wsa="http://www.w3.org/2005/08/addressing"
  targetNamespace="http://docs.oasis-open.org/bdxr/ns/SMP/2014/07"
  elementFormDefault="qualified"
  id="ServiceMetadataPublishing">

    <xs:import namespace="http://www.w3.org/2000/09/xmldsig#" schemaLocation="http://www.w3.org/TR/xmldsig-
core/xmldsig-core-schema.xsd"/>
    <xs:import namespace="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd"
schemaLocation="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd"/>

    <xs:element name="ServiceGroup" type="ServiceGroupType"/>
    <xs:element name="ServiceMetadata" type="ServiceMetadataType"/>
    <xs:element name="SignedServiceMetadata" type="SignedServiceMetadataType"/>
    <xs:element name="ErrorResponse" type="ErrorResponseType"/>

     <xs:complexType name="ErrorResponseType">
     <xs:sequence>
       <xs:element name="BusinessCode" type="xs:string" minOccurs="1" maxOccurs="1"/>
       <xs:element name="ErrorDescription" type="xs:string" minOccurs="0" maxOccurs="1"/>
     </xs:sequence>
     </xs:complexType>

    <xs:complexType name="SignedServiceMetadataType">
        <xs:sequence>
            <xs:element ref="ServiceMetadata"/>
            <xs:element ref="ds:Signature"/>
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="ServiceMetadataType">
        <xs:choice>
            <xs:element name="ServiceInformation" type="ServiceInformationType"/>
            <xs:element name="Redirect" type="RedirectType"/>
        </xs:choice>
    </xs:complexType>
    <xs:complexType name="ServiceInformationType">
        <xs:sequence>
            <xs:element name="ProcessList" type="ProcessListType"/>
            <xs:element name="Extension" type="ExtensionType" minOccurs="0"/>
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="ProcessListType">
        <xs:sequence>
            <xs:element name="Process" type="ProcessType" maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="ProcessType">
        <xs:sequence>
            <xs:element ref="ProcessIdentifier"/>
            <xs:element name="ServiceEndpointList" type="ServiceEndpointList"/>
            <xs:element name="Extension" type="ExtensionType" minOccurs="0"/>
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="ServiceEndpointList">
        <xs:sequence>
            <xs:element name="Endpoint" type="EndpointType" maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="EndpointType">
        <xs:sequence>
            <xs:element name="EndpointURI" type="xs:anyURI"/>
            <xs:element name="RequireBusinessLevelSignature" type="xs:boolean"/>
            <xs:element name="MinimumAuthenticationLevel" type="xs:string" minOccurs="0"/>
            <xs:element name="ServiceActivationDate" type="xs:dateTime" minOccurs="0"/>
            <xs:element name="ServiceExpirationDate" type="xs:dateTime" minOccurs="0"/>
            <xs:element name="Certificate" type="xs:base64Binary"/>
            <xs:element name="ServiceDescription" type="xs:string"/>
            <xs:element name="TechnicalContactUrl" type="xs:anyURI"/>
            <xs:element name="TechnicalInformationUrl" type="xs:anyURI" minOccurs="0"/>
            <xs:element name="Extension" type="ExtensionType" minOccurs="0"/>
        </xs:sequence>
        <xs:attribute name="transportProfile" type="xs:string" use="required"/>
    </xs:complexType>
    <xs:complexType name="ServiceGroupType">
        <xs:sequence>
            <xs:element name="CertificateAuthentication" type="CertificateAuthenticationType" minOccurs="0"
maxOccurs="1"/>
            <xs:element name="Extension" type="ExtensionType" minOccurs="0" maxOccurs="1"/>
        </xs:sequence>

     </xs:complexType>

     <xs:complexType name="CertificateAuthenticationType">
           <xs:sequence>
              <xs:element name="CertificateIdentifier" type="xs:string"/>
           </xs:sequence>
     </xs:complexType>

    <xs:complexType name="ServiceMetadataReferenceCollectionType">
        <xs:sequence>
            <xs:element name="ServiceMetadataReference" type="ServiceMetadataReferenceType" minOccurs="0"
maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="ServiceMetadataReferenceType">
        <xs:attribute name="href" type="xs:anyURI"/>
    </xs:complexType>
    <xs:complexType name="RedirectType">
        <xs:sequence>
            <xs:element name="CertificateUID" type="xs:string"/>
            <xs:element name="Extension" type="ExtensionType" minOccurs="0"/>
        </xs:sequence>
        <xs:attribute name="href" type="xs:anyURI" use="required"/>
    </xs:complexType>
    <xs:complexType name="ExtensionType" >
        <xs:sequence>
            <xs:any processContents="skip" minOccurs="0"/>
        </xs:sequence>
    </xs:complexType>
    <xs:element name="ParticipantIdentifier" type="ParticipantIdentifierType"/>
    <xs:element name="DocumentIdentifier" type="DocumentIdentifierType"/>
    <xs:element name="ProcessIdentifier" type="ProcessIdentifierType"/>
    <xs:element name="RecipientIdentifier" type="ParticipantIdentifierType"/>
    <xs:element name="SenderIdentifier" type="ParticipantIdentifierType"/>
    <xs:complexType name="ParticipantIdentifierType">
        <xs:simpleContent>
            <xs:extension base="xs:string">
                    <xs:attribute name="scheme" type="xs:string"/>
            </xs:extension>
        </xs:simpleContent>
    </xs:complexType>
    <xs:complexType name="DocumentIdentifierType">
        <xs:simpleContent>
            <xs:extension base="xs:string">
                    <xs:attribute name="scheme" type="xs:string"/>
            </xs:extension>
        </xs:simpleContent>
    </xs:complexType>
    <xs:complexType name="ProcessIdentifierType">
        <xs:simpleContent>
            <xs:extension base="xs:string">
                    <xs:attribute name="scheme" type="xs:string"/>
            </xs:extension>
        </xs:simpleContent>
    </xs:complexType>
</xs:schema>
```

4.2. Errors codes table

The following table summarizes all possible errors returned by the SMP services:

HTTP code	HTTP Message	Business code	Meaning	Applicable UC						
				UC01	UC02	UC03	UC04	UC05	UC06	UC07
				n/a	PUT	DEL	PUT	DEL	GET	GET
200	OK	n/a	The request was completed successfully	-	X	X	X	X	X	X
201	Created	n/a	The PUT operation completed successfully	-	X		X			
400	Bad Request	n/a	There is a format error in the request	-	X	X	X	X		
401	Unauthorized	n/a	The user is not granted the right to issue this request	-	X	X	X	X		
404	Resource not found	n/a	The requested information was not found	-		X		X	X	X
500	Server Internal Error	TECHNICAL	Some unexpected technical error occurred.(detailed information available in the response)	-	X	X	X	X	X	X
500	Server Internal Error	XSD_INVALID	The XML included in the request is not validate against the XSD defining the input structure	-	X		X		▨	▨
500	Server Internal Error	MISSING_FIELD	Some field that is optional in the XSD but mandatory for this invocation is missing (missing field's name in description)	-	X		X		▨	▨
500	Server Internal Error	WRONG_FIELD	Some field is valid against XSD definition, but the more specific content is invalid (erroneous field's name in description)	-	X		X		▨	▨
500	Server Internal Error	OUT_OF_RANGE	Some numeric (or date field) is out of the valid range (erroneous field's name in description)	-	X		X		▨	▨

				Applicable UC						
				UC01	UC02	UC03	UC04	UC05	UC06	UC07
HTTP code	HTTP Message	Business code	Meaning	n/a	PUT	DEL	PUT	DEL	GET	GET
500	Server Internal Error	UNAUTHOR_FIELD	Some field that is optional in the XSD but forbidden for this invocation is present (unauthorized field's name in description)	-	X		X		▨	▨
500	Server Internal Error	FORMAT_ERROR	Some field is expected to have a specific format is not valid (erroneous field's name in description)	-	X		X		▨	▨

4.3. Detailed Errors’ structure

In case of ‘Server Internal Error’ (HTML ERROR 500), a response text will be provided, in an “ErrorResponse” type of element (cf. definition in 4.1.3 – “Extended SMP XSD for ADMIN services (PUT)”)

Example:

```
<?xml version="1.0" encoding="UTF-8"?>
<ErrorResponse xmlns="http://docs.oasis-open.org/bdxr/ns/SMP/2014/07">
    <BusinessCode>TECHNICAL</BusinessCode>
    <ErrorDescription>Some unexpected technical error occurred.(detailed information available here)</ErrorDescription>
</ErrorResponse>
```

4.4. eHealth specificities

This paragraph discuss major specificities related to eHealth project specificities and is therefore not to be considered as reference for other implementations.

4.4.1. OASIS/EHealth equivalence

The following table introduces major SMP terms and link them to eHealth equivalents.

SMP	Definition and references to eHealth
Access Point (AP)	NCP
ServiceMetadata	Data structure registered in a SMP holding Network addresses, web service endpoints and certificates of a country's epSOS service providers and consumers
ServiceMetadataCollection	Data structure are registered in a SMP holding all the services exposed by a NCP
ServiceMetadataReference	URL of the ServiceMetadata file associated for the service. The scheme of the URL is /{identifier scheme}::{id}/services/{docType} which expands to /{identifier scheme}::{id}/services/{document identifier scheme}::{document identifier} Example: http://smp.location.de/ehealth-participantid-qns%3A%3Aurn%3Aehealth%3Ade%3Ancpb-idp/services/ehealth-resid-qns%3A%3Aurn%3A%3Aepsos%3Aservices%23%23epsos-11
ParticipantIdentifier	unique uri of the NCP (e.g., urn:germany:ncpeh)
DocumentIdentifier/@Scheme	"ehealth-resid-qns"
ProcessIdentifier	URI of the specific service
ProcessIdentifier/@Scheme	"ehealth-procid-qns"
ServiceEndpointList	Service information for each service endpoint served by this service. E.g., if service is Patient Service, the operation is List().
Endpoint/@transportProfile	Profile for this service. E.g.: • urn:ihe:iti:2013:xcpd • urn:ihe:iti:2013:xds • urn:ihe:iti:2013:xca • urn:ihe:iti:2013:xcf
EndpointURI	WSE of a specific service
RequireBusinessLevelSignature	not used by eHealth

SMP	Definition and references to eHealth
TechnicalContactURL	Information related to the technical contact
TechnicalInformationURL	URL pointer to the remote service technical description

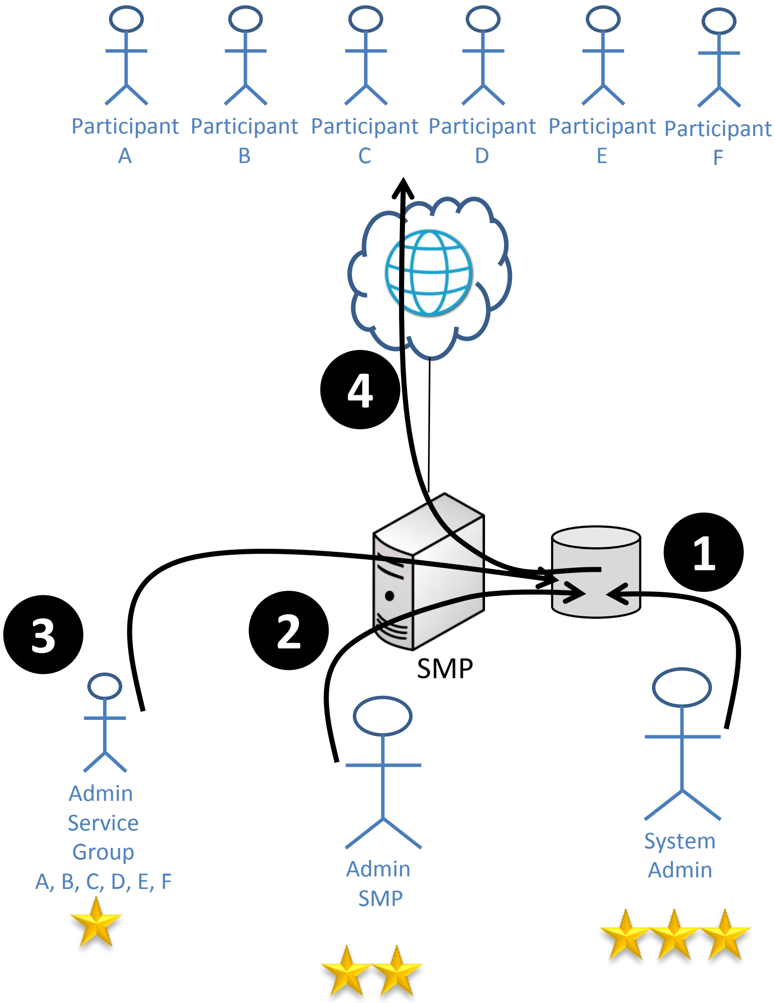
4.4.2. *eHealth SMP administration process with a single centralized SMP*

In the SMP administration model required by eHealth, only one SMP manages all ServiceGroups. That single SMP instance could be either hosted by DIGIT or the responsible authority of the business. eHealth will host its SMP instance.

The central authority owns the two users “System Admin” and “Admin SMP”.

This process is a simplified version of the distributed one. In that case:

- there may be only one "Admin ServiceGroup" (still multiple may also be defined as above).
- Since "Admin ServiceGroup" users are in a secure environment, basic authentication could be used instead of a certificate with 2-way-ssl.



In step ❸ of the picture besides, "*Admin ServiceGroup A, B, C, D, E, F*" defines some of the services for one or several parties among all.

4.5. Document source files

The attached files contain table and drawings included upper in this document and are included to facilitate future updates of this document.

Use cases.xlsm

SMP ICD
drawings.ppt

TODO (for each release): replace updated Excel and Powerpoint files.