DomiSMP Documentation

DomiSMP Documentation

Chapter 1. Architecture

1.1. Introduction

Service metadata publishing (SMP) was introduced to eDelivery network by PEPPOL project [REF7]. The purpose of the SMP is similar to an address book or business registry. eDelivery participants (message senders and receivers) use SMP to publish their transport/service capabilities and to discover partner's transport/service capabilities as: delivery addresses, supported business processes and document types, etc. The PEPPOL's SMP specification was submitted as input to the OASIS BDXR TC (Business Document Exchange Technical Committee) with the intent of defining a standardized and federated document transport infrastructure for business document exchange. It resulted into a new specification: OASIS Service Metadata Publishing Specification (OASIS SMP specification) [REF1].

The eDelivery Service Metadata Publisher Profile (eDelivery SMP profile) [REF2] provides a set of implementation guidelines for the OASIS SMP specification[REF1]. It is designed to be used in eDelivery with the dynamic receiver (and sender) discovery functionality.

The eDelivery Service Metadata Publisher application (DomiSMP) is the sample implementation of the eDelivery SMP profile (thus OASIS SMP spec as well).

▼ **Purpose**

This document is the Software Architecture Document of the DomiSMP application. It is intended to provide detailed information about the project:

- An overview of the solution
- A description of business and administration functions implemented in the DomiSMP
- A description of the application architecture and its modules
- An overview of code organization and code quality measurements
- An overview of technical requirements

▼ **References**

REF1

OASIS SMP Specification, Version 1.0. This specification defines documents and REST binding of SMP public interface.

REF2

eDelivery SMP profile. eDelivery profile of [REF1] specification.

REF3

eDelivery SMP Administration Guide (pdf). See Documentation section of SMP Software. SMP Administration Guide.

REF4

Interface Control Document (pdf).
See Documentation section of SMP Software.

Defines interface of eDelivery SMP – extends OASIS SMP specification

REF5

SML Administration Guide (pdf) See Documentation section of SML Software. Provides comprehensive details on eDelivery SML installation, configuration and maintenance.

eDelivery BDMSL (SML)

eDelivery BDMSL (SML). Application offered by eDelivery in SaaS model. Facilitates write access to the DNS zone needed for dynamic discovery of Participants. Exposes SOAP interface that is consumed by SMP in order to (un)register participant DNS entries.

PEPPOL

PEPPOL. The Pan-European Public Procurement On-Line (PEPPOL) project was a pilot project funded jointly by the European Commission and the PEPPOL Consortium members. After successful completion of the project new organization OpenPEPPOL Association was established. The organization is now responsible for the governance and maintenance of the PEPPOL specifications.

OASIS SMP 2.0

OASIS Service Metadata Publishing (SMP) specification, Version 2.0.
This document describes the version 2.0 of the Oasis SMP standard.

OASIS ebCore Party Id Type TS Version 1.0

OASIS ebCore Party Id Type Technical Specification Version 1.0. This document describes the OASIS ebCore Party Id Type.

▼ **Definitions**

Service Metadata Publisher (SMP)

REST service application providing set of CRUD operations for two web resources: ServiceGroup and ServiceMetadata. SMP is eDelivery implementation of [REF1] and [REF4].

Identifier

The identifier uniquely identifies DomiSMP entities such as resources and subresources. An identifier consists of a schema (namespace) and a value. An identifier has rules about how it is represented in the URL (concatenated format) and how it is written in the resource document. (See the chapter: 3.1 Identifiers)

ParticipantIdentifier

The ParticipantIdentifier is an entity that uniquely identifies receiver or sender (participants) in eDelivery process. Examples of identifiers are company registration and VAT numbers, DUNS numbers, GLN numbers, email addresses etc.

Resource

The DomiSMP URL resource is associated with a specific Participant Identifier. The resource can be Service Group (see below) or any other document type supported by the DomiSMP extensions.

ServiceGroup

The ServiceGroup contains list of services associated with a specific Participant Identifier that is handled by a Service Metadata Publisher. ServiceGroup XML representation is defined by XML Schema attached to [REF1].

Subresource

The DomiSMP URL (sub)resource is the sub-document of the resource. The resource can be ServiceMetadata (see below or any other document type supported by the DomiSMP extensions).

ServiceMetadata

TheServiceMetadatacontains all necessary metadata (endpoint URLs, certificate for encryption, document types, etc.) about a specific service that a participant (service requestor) needs to know in order to send a message to that service. ServiceMetadata XML representation is defined by an XML Schema included into [REF1].

SignedServiceMetadata

ServiceMetadata signed by Service Metadata Publisher (SMP).

DocumentIdentifier

represents document types in a service. It also contains scheme type which represents format of the identifier itself. XML representation is defined by an XML Schema included into [REF1] as part of ServiceMetadata.

BDMSL (SML)

Application offered by eDelivery in SaaS model. Facilitates write access to the DNS zone needed for dynamic discovery of Participants. Exposes a WSDL interface that is consumed by SMP in order to (un)register participants' DNS entries.

Domain

The Domain indicates the purpose of the exchange network, such as the E-Invoice exchange, eHealth record exchange, etc.
If the domain network uses the delegated dynamic discovery service, the domain has its own DNS zone handled by the BDSML application. For eDelivery SML the domains are:

- acc.edelivery.tech.ec.europa.eu: acceptance domain for testing SMP instances and subdomains.
- delivery.tech.ec.europa.eu: production domain.

Group

The domain participant group. The Domain can have one or more groups where the Group admin is responsible for the particular group of participants for creating and deleting the domain resources. For example, the domain groups allow the Domain’s resources (e.g., service groups) to be segmented into different countries, regions, etc. and managed by the responsible group admin

Subdomain

Subdomain defines business domains handled by BDSML application in particular DNS zone.

Examples of subdomain (business domain) are: peppol, ehealth, generalerds and they are all in part of domain (DNS zone) edelivery.tech.ec.europa.eu domain.

Dynamic Discovery

Dynamic Discovery is the process of discovering participants' service metadata.

1.2. Solution Overview

The eDelivery Service Metadata Publisher (DomiSMP) enables the participants of an eDelivery Messaging Infrastructure network to dynamically discover each other’s capabilities (Legal, Organisational, and Technical). For this to happen, each participant must publish into an SMP its capabilities and settings (including but not limited to):

- business processes that the participant supports
- the security setup (public key certificate)
- the transport protocol (AS2 or AS4)
- the location of the receiver’s access point

The SMP usually serves multiple participants to publish theirs exchange capabilities. But in eDelivery network/business domain can coexist in multiple SMPs. Because of this distributed architecture, each participant must have a unique ID in a particular subdomain. A central component, called Business Document Metadata Service Location (BDMSL) [REF6], uses these IDs to create URLs that, when resolved, direct the eDelivery Access Points towards the specific SMP of the participant.

The SMP software component described in this document implements the eDelivery SMP profile [REF2] based on the OASIS Service Metadata Publishing (BDX SMP) [REF1] specifications.

1.3. Functional View

This section describes interactions, data flows and dependencies between SMP and other integrated applications in dynamic discovery process. All use cases refer to the ICD document (cf. [REF4]), where they are presented with more interface-specific details.

The use cases

- UC06 – GET ServiceGroup
- UC07 – GET ServiceMetadata

are implementations of the service defined in OASIS SMP Specification [REF1]. All other use cases cover administration/maintenance services which are not part of the specifications.

The use cases cover RESTful CRUD operations for following SMP business objects:

ServiceGroup, under relative URL:

```
/\{ParticipantIdentifierScheme}::\{ParticipantIdentifierValue}
```

ServiceMetadata, under relative URL:

```
/\{ParticipantIdentifierScheme}::\{ParticipantIdentifierValue}/services/\{DocTypeIdent
ifierScheme}::\{DocTypeIdentifierValue}
```

1.3.1. Identifiers

The identifier uniquely identifies DomiSMP entities such as resources (e.g., ServiceGroups) and subresources (e.g., ServiceMetadata). The identifiers are being used in the URL requests as part of the URL request path segment, and also in the (sub)resource documents.

Example of the URL request (scheme: ‘oasis:names:tc:ebcore:partyid-type:iso6523:0088’, value: ‘4035811991021’) concatenated with single-colol ‘:’

```
http://my-app.example.eu/smp/urn:oasis:names:tc:ebcore:partyid-
type:iso6523:0088:4035811991021
```

Example of the document element with the participant identifier split to scheme attribute and element value:

```
<ParticipantIdentifier
scheme=▯urn:oasis:names:tc:ebcore:partyid-type:iso6523:0088▯>
4035811991021</ParticipantIdentifier>
```

▼ *Identifiers encoding*

According to OASIS SMP Specification [REF1] and [REF8] above, SMP deals with two types of identifiers: participant and document identifier. The specification [REF1] prescribes that both are built out of scheme and value, delimited by special character(s) such a double-colon separator "::" or single-colon separator ":" as defined in the OASIS ebCoreParty Id: [REF3] and [REF9].

ServiceGroup identifier, from business perspective known as Participant Identifier

```
ServiceGroup identifier :=
\{ParticipantIdentifierScheme}::\{ParticipantIdentifier}
```

ServiceMetadata Identifier, from business perspective known as Document Type Identifier

```
ServiceMetadata identifier :=
\{DocTypeIdentifierScheme}::\{DocTypeIdentifier}
```

All identifiers that are included in the URL of the REST request must be URL-encoded (note also the double-colon separator "::").

Example: the participant identifier (ServiceGroup identifier) built out of:

- ParticipantIdentifierScheme ="participant#domain#scheme"
- ParticipantIdentifier ="participant#id"

must be encoded in URL request to:

- participant%23domain%23scheme%3A%3Aparticipant%23id

Moreover, in some cases (all PUT requests), the identifiers are present in the URL and in the XML body of the request. In these cases, only identifiers in URL must be URL-encoded.

▼ *ebCore party identifier*

The eDelivery SMP has the feature to support handling participant identifiers as described in eDelivery SMP profile [REF3] in the chapter “Use with eDelivery ebCore Party Identifiers”. In this case, the participant starts with the: `urn:oasis:names:tc:ebcore:partyid-type:` following by the words: `unregistered` or `iso6523`

All ebCore party identifiers in the REST request must be URL-encoded using only one double-colon separator ":", as in below example:

- `urn:oasis:names:tc:ebcore:partyid-type:iso6523:0088:4035811991021`

URL-encoded example:

- `urn%23oasis%23names%23tc%23ebcore%23partyid-type%23iso6523%230088%234035811991021`

The eDelivery SMP has the option to serialize ebCore party Id to XML according to the OASIS SMP Specification [REF1] as separate values, as in below example:

```
<ParticipantIdentifier
    scheme=▯urn:oasis:names:tc:ebcore:partyid-type:iso6523:0088▯>
    4035811991021
</ParticipantIdentifier>
```

or according to the eDelivery SMP profile [REF2] as concatenated value:

```
<ParticipantIdentifier>
 urn:oasis:names:tc:ebcore:partyid-type:iso6523:0088:4035811991021
</ParticipantIdentifier>
```

The behaviour can be configured and is explained in more details in §5 – "Configuration".

▼ *Identifier’s case sensitivity*

SMP can handle identifiers (scheme and value) in case-sensitive or in a non-case-sensitive way. The behaviour can be configured: more details can be found in §5 – "Configuration".

When the SMP is configured as non-case-sensitive the SMP normalizes the identifiers extracted from the requests. Identifiers within incoming requests are considered as case-insensitive and converted to lowercase. Further processing like the storage and querying in the database is performed using lowercase letters only. If the case-sensitivity configuration is modified, the database records must be updated manually.

When the SMP is configured as case-sensitive, then Identifiers are not modified during the whole request processing.

1.3.2. BDMSL Integration

Creation or removal of ServiceGroup within SMP triggers a synchronous (un)registration of relevant record(s) in DNS. This process is required to allow Dynamic Discovery of SMPs to store Participant’s metadata.

Write access to DNS zone is facilitated by BDMSL (SML), a centralized application that exposes a SOAP interface for that purpose (cf. [REF6]). SMP is a consumer of the SML services. SML authorization of SMP is based on mutual HTTPS authentication. Therefore, SMP client TLS certificate with private key needs to be configured on SMP side.

If SMP serves data in only one domain, then a single certificate is needed. Otherwise, if the SMP is configured to work in multi-domain mode, the System Administrator will need to set up one certificate per subdomain. More details can be found in chapter §3.3 – "Domain Multitenancy" and §5 – "Configuration".

1.3.3. Domain Multitenancy

An SML subdomain can be considered as a set of an inter-network of eDelivery components: SML, SMPs and Access Points for a business domain. All these members communicate with each other within that subdomain and exchange messages according to the strict rules defined for that business domain. One network can be used to exchange invoices between participants, another one could exchange health information between hospitals and insurance companies, etc.

In most scenarios there will be multiple SMPs in a single business domain and each of them will handle ServiceMetadata sets of multiple participants from the same subdomain. The business domain authority can set its own SMP to administrate its participants and the SMP is used only in one domain. But an SMP could be used in more than one business domain at the same time. Because of SML restrictions such setup implies the following SMP functionality:

- The SMP must use a different SMP ID and a different certificate to authenticate for a particular SML subdomain.
- The SMP must be able to sign ServiceMetadata responses using a different certificate for each domain (one certificate per domain).

1.3.4. Roles

Roles are documented with more details in the ICD document (cf. [REF4]). The table below explains their meaning from a functional perspective:

Role Alias	Description
Anonymous	Any user that has not provided any authentication details. This user can query for public resources e.g.: `ServiceGroup` and sub resources, for example, `ServiceMetadata`.
User	User with the role can log in to the DomiSMP and has access and edit rights to resources according to memberships on resources, groups and domain. For example, user who is a member of the resource with Admin membership can perform administrative actions update service group extension data and add/update/delete service metadata for the service group. User who is member of the Group with Group Admin membership role is allowed to execute create and resource for the group. User who is member of the Domain with Domain Admin membership role, can create/delete groups for Domain and manage the domain memberships.
System Admin	System user who can administer domains, users, application properties, truststore and keystore on the DomiSMP.

1.3.5. Domain, Group and Resources

The DomiSMP supports 3-layer security realms.

- The most basic unit is the **Resource**. The Resource is identified by the unique ID, which is part of the URL of the resource as example:

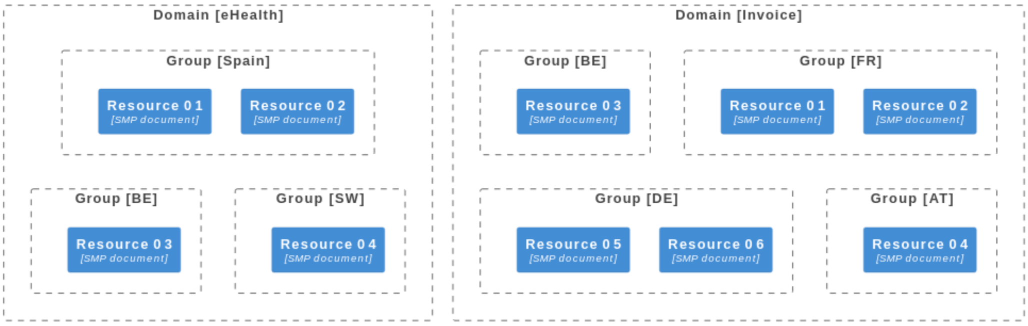
  http://localhost/smp/resource-identifier

An example of the Resource is the “Service Group” document from the Oasis SMP specification.

The user can be a Resource member with **Admin** or **Viewer** membership roles. If the user has an Admin membership role, it can modify resource document(s) and manage the resource memberships. If the user has role Viewer, it can view/read the Resource if the Resource has visibility set to: “Private”.

- The **Group** is a cluster of resources managed by the dedicated group administrators. The group admin(s) can create and delete the resource, but **only** the resource admins can modify data/documents for the resource. The user can be a Group member with **Admin** or **Viewer** membership roles. With Admin group membership, the user can create and delete group resources. If the user has group role Viewer, it can view/read the Resources if the Group has visibility set to: “Private”.
- The top layer is the **Domain**. It indicates the business purpose of the network of participants, such as invoice exchange, Health Records message exchanges, etc. The Domain usually has a domain owner who handles participant interoperability, defining message types, network authentication, and authorization methods such as Certificate PKI, Identity Service providers,

etc. In DomiSMP 5.0, the user with a Domain Admin role can create domain groups and assign users to them.



1.3.6. Extensions

One of the main DomiSMP sample implementation purpose is enabling the setup (and testing) of various network configurations. Designed with flexibility in mind, DomiSMP allows the implementation of custom logic of the document processing. To achieve this, developers can create custom extensions. These extensions are packaged as JAR files and extend one or more interface classes from the DomiSML module `eu.europa.ec.edelivery:smp-spi`.

Example of how to include an extension in your custom extension project:

```
<dependency>
 <groupId>eu.europa.ec.edelivery</groupId>
 <artifactId>smp-spi</artifactId>
 <version>$\{project.version}</version>
</dependency>
```

The extension JAR must be added to the DomiSMP extension library path before starting up the application. The path where the extensions must be deployed is defined with file property: `smp.libraries.folder`.

An example:

Extension folder
path where SMP extensions are located. The Folder is loaded by the SMP classloader at startup.

`smp.libraries.folder=/cef/test/smp/apache-tomcat-8.5.73/smp/ext-lib`

DomiSMP 5.0.x supports two types of extensions:

Resource Handling Extension

DomiSMP supports various document types via custom designed extension. This extension handles/processes the resource and sub resource documents.

Payload Validation Extension

with this extension, users can validate payloads/documents according to specific rules. Users can develop custom security scanning of all payload uploaded to the DomiSMP.

In the next section we describe both extensions in more detail.

When the DomiSMP library is loaded by the class loader, the extension registrar searches for Spring beans that implement the “ExtensionInfo” interface. These beans provide essential information about the extensions.

Key parameters

Identifier

The unique identifier represents the extension. If an extension is upgraded, its identifier must remain the same to ensure proper handling of existing extension data.
Example identifier: edelivery-oasis-smp-extension.

Name

The human-readable name of the extension. This name helps users understand the purpose and functionality of the extension.

- **Description**: A brief description of the extension, providing purpose and extension details.
- **Version**: The version number of the extension, indicating its release or revision.
- **ResourceTypes**: List of resource handling extensions.
- **PayloadValidators**: List of Payload validator extensions
  - import eu.europa.ec.smp.spi.resource.ResourceDefinitionSpi;
  - import java.util.List;

/ * DomiSMP extension information. When updating the extension it must have the same Name for DomiSMP to handle the upgrade correctly. /

```
public interface ExtensionInfo \{
String identifier();
String name();
String description();
String version();

List<ResourceDefinitionSpi> resourceTypes();
List<PayloadValidatorSpi> payloadValidators();
}
```

Resource Handling Extension

One of the most important purposes of DomiSMP is to allow users to publish various connectivity capability documents for serving or business message exchange. Examples of these documents include OASIS SMP 1.0 and OASIS SMP 2.0 documents, as well as ServiceGroup and ServiceMetadata documents, CPP from Oasis CPPA3, and any other text-based custom documents.

The resource extension enables the following tasks:

- Automatic Document Generation: the extension automatically generates sample documents for an extension. For example, Oasis SMP 1.0 extension can generate sample document for

ServiceGroup and ServiceMetadata which are used by the DomiSMP User interface when creating new resources.

- Document Validation During Registration: when a new document is registered, the extension validates its structure, metadata, and content. This ensures that only valid documents are accepted and published on the DomiSMP.
- Document Management: the extension can validate, modify, and update documents on read, store, and validate action.

When creating a resource handling extension, developers must implement the ResourceDefinitionSpi and optionally the SubresourceDefinitionSpi. These interfaces contain the resource definition such as identifier, version name, document mimetype, etc.

The ResourceDefinitionSpi and SubresourceDefinitionSpi contain the resource handling extension metadata for the Resources/Subresources. To process the document, the resource/subresource implementation must also contain the implementation of the ResourceHandlerSpi which handles the document processing such as: generation of empty document, validation of the document, and methods which are invoked while reading and storing the document. Below is the definition of the Resource handler interface.

```
package eu.europa.ec.smp.spi.resource;

import eu.europa.ec.smp.spi.api.model.RequestData;
import eu.europa.ec.smp.spi.api.model.ResponseData;
import eu.europa.ec.smp.spi.exceptions.ResourceException;

import java.util.List;


/
* The class implementing the ResourceHandlerSpi must support read transformation, store
transformation, and
* validation methods for the particular resource type, such as Oasis SMP 1.0 document, CPP
document, etc.
*
*
*/
public interface ResourceHandlerSpi \{

/
* Method get data from the resource in the input stream, and it writes transformation of the data as
they are returned to
*
* @param resourceData the resource data
* @param responseData the date object for setting the response
*/
void readResource(RequestData resourceData, ResponseData responseData) throws
ResourceException;
void storeResource(RequestData resourceData, ResponseData responseData) throws ResourceException;

/
* Validate resource schema and data. if resource is invalid the error is thrown
* @param resourceData the resource data
*/
void validateResource(RequestData resourceData) throws ResourceException;


/
* Validate resource schema and data. if resource is invalid the error is thrown
* @param resourceData the resource data
*/
void generateResource(RequestData resourceData, ResponseData responseData, List<String> fields) throws ResourceException;

}
```

For more detail, please see the DomiSMP code repositories with Maven module: smp-resource-extensions which contains example implementations of the Oasis SMP 1.0 and Oasis SMP 2.0 documents and basic Oasis CPPA3-CPP document.

More examples (JSON and property files) of DomiSMP Resource extensions can be found in Maven submodule: smp-examples/resource-spi-example. can be found in Maven submodule: smp-examples/resource-spi-example.

Payload Validation Extension

To increase security, the eDelivery SMP offers the possibility of registering custom extensions for security scanning/validations of all binary documents such as the certificates and the keystores. The certificates can be uploaded by the users when setting the user certificate for authentication. The keystores binaries can be uploaded by the System Administrators when managing the SMP keystore.

When the user loads one of the mentioned payloads, the eDelivery SMP validation framework is activated. At this point, the payload binary data is passed to all registered spring beans, which implement the `PayloadValidatorSpi` interface below:

```
package eu.europa.ec.smp.spi;

import eu.europa.ec.smp.spi.exceptions.PayloadValidatorSpiException;

import java.io.InputStream;

/
*
* SMP Service provider interface (SPI) for uploaded payload validation.
* This SPI interface is intended to allow antivirus validation using third-party antivirus
```

```
software.
*/
public interface PayloadValidatorSpi \{

/
* Validates the SMP payload. If the payload is invalid the method MUST
* throw PayloadValidatorSpiException
*
* @param payload The payload data to be validated
* @param mimeType The payload mime type
* @throws PayloadValidatorSpiException in case the validation does not pass
*/
void validatePayload(InputStream payload, String mimeType) throws
PayloadValidatorSpiException;
}
```

The implementers of the extension must implement the method `validatePayload` for payload validation. In the event of malware detection, the method MUST throw the `PayloadValidatorSpiException` to terminate the future payload handling by the eDelivery SMP.

A simple example of the `PayloadValidatorSpi` implementation can be found in the SMP project module `smp-examples/smp-spi-example/` (See chapter §4.1). /XREF

To register the extension in the eDelivery SMP, the interface implementation class must be

- located under the java package ***eu.europa.ec.smp.spi,***
- tagged with spring bean annotation ***@Component*** or ***@Service***,

as in below example:

```
package eu.europa.ec.smp.spi.example;

import eu.europa.ec.smp.spi.exceptions.PayloadValidatorSpiException;
import org.springframework.stereotype.Service;
import java.io.InputStream;

`@Service
public class ExamplePayloadValidatorSpiImpl implements PayloadValidatorSpi \{
public void validatePayload(InputStream payload, String mimeType) throws
PayloadValidatorSpiException \{

 1. . .
    }
    }
```

To prepare the extension for the deployment in the eDelivery SMP, the code must be compiled and stored in the java archive file format known as the JAR.

In the eDelivery SMP, the property**libraries.folder**must be configured in the* *smp.config.properties**to point to the folder where extension libraries are located. The SMP classloader loads the libraries in the folder at the startup of the SMP and registers the `PayloadValidatorSpi` beans.

1.4. Use Case Details

Here you can find a description of each Use Case.

1.4.1. UC01 Manage Administrators

▼ *UC01 Manage Administrators details*

Prerequisites

- User (system admin) has rights to modify content of SMP configuration tables.

Description

This use case does not involve SMP application, instead the user's management is implemented as a simple manual SQL queries. Users and its roles are not cached by the SMP, so they can be used immediately after the corresponding SQL transaction is committed. Sample SQLs inserting users authenticated by password or certificate are presented below. More details on users can be found in §4.3.4 – "Data layer" and §6 - "Security".

— user authenticated with password (oracle dialect)

insert into SMP_USER (ID, USERNAME, ACTIVE, APPLICATION_ROLE, EMAIL, CREATED_ON, LAST_UPDATED_ON) values

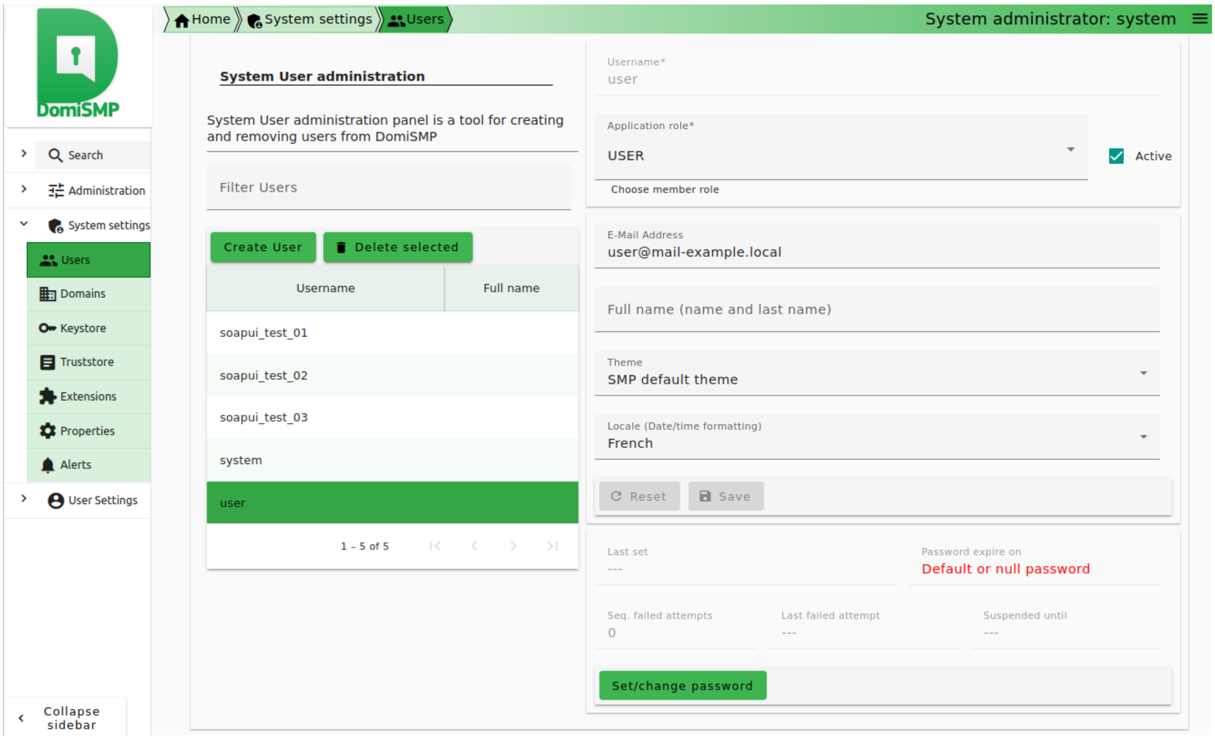
(SMP_USER_SEQ.NEXTVAL, 'smp_admin', 1, 'SYSTEM_ADMIN', 'system@mail-example.local', sysdate, sysdate);

insert into SMP_CREDENTIAL (FK_USER_ID, CREDENTIAL_ACTIVE, CREDENTIAL_NAME, CREDENTIAL_VALUE, CREDENTIAL_TYPE, CREDENTIAL_TARGET, CREATED_ON, LAST_UPDATED_ON) values

_((select id from SMP_USER where USERNAME='smp_admin'),1, 'smp_admin', '$2a$10$olcGeWKGEoRia2DPuFqRNeca0IEdRSmOrljLz57BAjf1jlC9SohrS', 'USERNAME_PASSWORD','UI', sysdate, sysdate);

_

If the system administrator user is already configured, the system administrator can use the eDelivery SMP UI tool to further manage users.



System User administration

System User administration panel is a tool for creating and removing users from DomiSMP

Username	Full name
soapui_test_01	
soapui_test_02	
soapui_test_03	
system	
user	

1 - 5 of 5

Username*: user

Application role*: USER

☑ Active

Choose member role

E-Mail Address: user@mail-example.local

Full name (name and last name)

Theme: SMP default theme

Locale (Date/time formatting): French

Last set: ---

Password expire on: Default or null password

Seq. failed attempts: 0

Last failed attempt: ---

Suspended until: ---

NOTE

For invoking the PUT or DELETE Use cases described in the sections below, credentials such as Access token or Client certificate must be used for the authentication.
See Security (DomiSMP Architecture).

1.4.2. UC02 PUT ServiceGroup

▼ *UC02 PUT ServiceGroup details*

Prerequisites

PUT ServiceGroup (Create or Update):

- The authenticated user has the role of "Admin SMP".
- If the ServiceGroup is managed remotely, the "Resource Admin" must have been created before in the "Administrator" table.
- If the SMP is serving multiple domains, the header field "Domain" must be populated and refer to one of the domains served by the SMP.

Description

`PUT ServiceGroup` is an idempotent create/update REST action.

+ NOTE: Idempotence is the property of certain operations in whereby they can be applied multiple times without changing the result beyond the initial application.

+ If the SMP is configured to be integrated with BDMSL, then additional synchronous request is performed to register the newly created Participant in the DNS. A sample request is presented below, with the following conventions:

Dark-grey HTTP headers are optional.

Identifiers present in the body of the request and in the URL marked in yellow must match.

Successful responses:

HTTP 200 (OK) – ServiceGroup was updated HTTP 201 (Created) – New ServiceGroup was created

PUT http://smp.eu/participant-domain-scheme#%3A%3Aparticipant-id

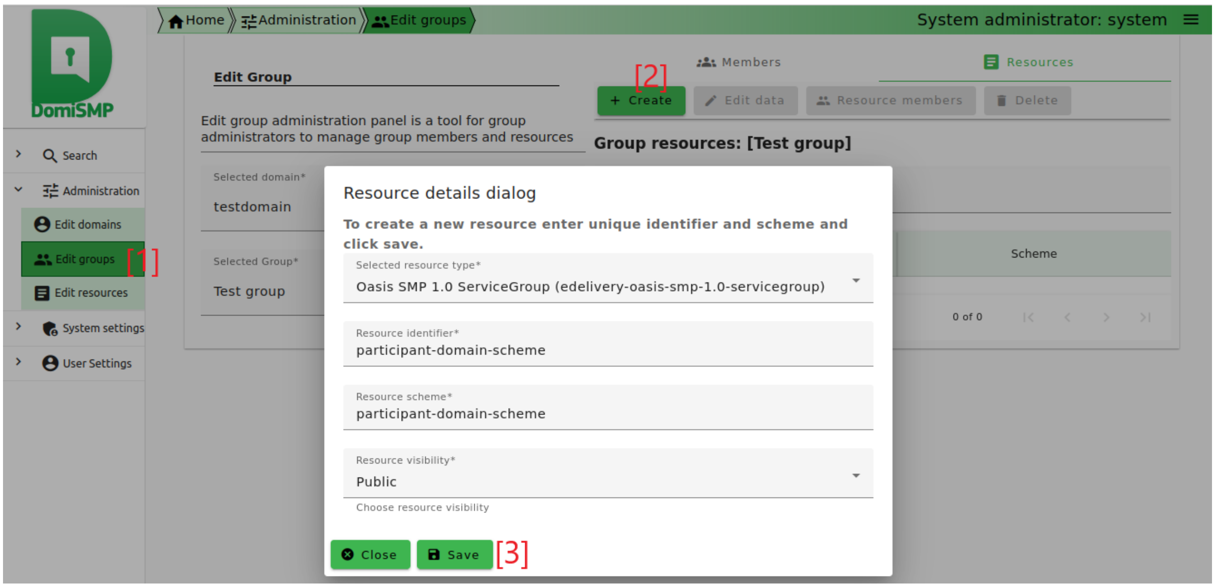
HTTP/1.1

Accept-Encoding: gzip,deflate Content-Type: text/xml;charset=UTF-8 Authorization: Basic c21wX2FkbWluOmNoYW5nZWl0

ServiceGroup-Owner: anotherownerusername Domain: domain2 Content-Length: 284

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<ServiceGroup xmlns="http://docs.oasis-open.org/bdxr/ns/SMP/2016/05">
<ParticipantIdentifier scheme="participant-domain-scheme">
    participant-id
</ParticipantIdentifier>
<ServiceMetadataReferenceCollection/>
</ServiceGroup>
```

The DomiSMP group administrator can also register a ServiceGroup with the DomiSMP UI tool for Service group management (see [1] on the picture below). The ServiceGroup is registered by activating/clicking the save button (see [3] on the picture below) after all the necessary data are entered.



Edit Group

Edit group administration panel is a tool for group administrators to manage group members and resources

Selected domain*: testdomain

Selected Group*: Test group

Group resources: [Test group]

Resource details dialog

To create a new resource enter unique identifier and scheme and click save.

Selected resource type*: Oasis SMP 1.0 ServiceGroup (edelivery-oasis-smp-1.0-servicegroup)

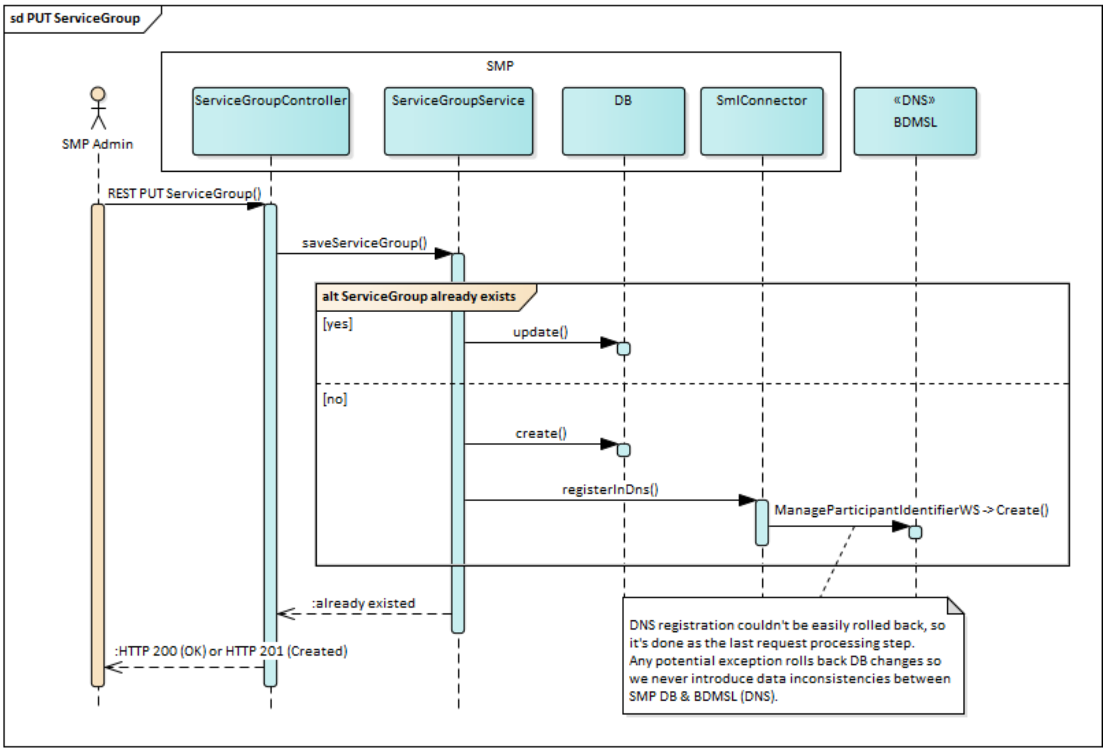
Resource identifier*: participant-domain-scheme

Resource scheme*: participant-domain-scheme

Resource visibility*: Public

Choose resource visibility

If BDMSL integration is enabled and configured for the selected domain, the SML request is submitted when the ServiceGroup is created.



DNS registration couldn't be easily rolled back, so it's done as the last request processing step. Any potential exception rolls back DB changes so we never introduce data inconsistencies between SMP DB & BDMSL (DNS).

ServiceGroup-Owner HTTP header

Specifying Owner User
Only the DomiSMP Group administrator has permission to register (or delete) the ServiceGroup. The Group administrator usually creates a ServiceGroup for the end-user with the "Resource Admin" role, which has only the permission to update the ServiceGroup service metadata.

By default, the Admin of the ServiceGroup is the user who created the ServiceGroup. But this can be changed at creation time by setting the*ServiceGroup-Owner*HTTP header with a different owner’s identifier. The identifier of the service owner can be the username, the users access token identifier, or the certificate identifier.

Below are examples of HTTP header **ServiceGroup-Owner:**

ServiceGroup-Owner: anotherownerusername

Non-ASCII characters must be URL-encoded, i.e. user **Żółty Jérôme** should be encoded in this way:

ServiceGroup-Owner: %C5%BB%C3%B3%C5%82ty%20J%C3%A9r%C3%B4me

Users authenticated by certificate can become owners as well, i.e. user **CN=new owner,O=EC,C=BE:000000000000100f** should be encoded:

ServiceGroup-Owner: CN%3Dnew%20owner,O%3DEC,C%3DBE%3A000000000000100f

Domain HTTP header

Specifying Domain
This feature is used only when the SMP is set up in multi-domain mode. When creating new ServiceGroup the Domain HTTP header must be specified in the PUT ServiceGroup request

Domain: domain2

More details on Multitenancy can be found in §3.3 – "Domain Multitenancy".

1.4.3. UC03 DELETE ServiceGroup

▼ *UC03 DELETE ServiceGroup details*

Prerequisites

- The authenticated user has the role of "Admin SMP".
- If the ServiceGroup is managed remotely, the "Resource Admin" must have been created before in the "Administrator" table.
- If the SMP is serving multiple domains, the header field "Domain" must be populated and refer to one of the domains served by the SMP.

Description

This action removes the specified ServiceGroup from SMP's database **including all related ServiceMetadata**.

If the SMP is configured to integrate the BDMSL, then an additional synchronous request is issued in order to unregister the Participant from the DNS.

Successful responses:

HTTP 200 (OK) – ServiceGroup was removed

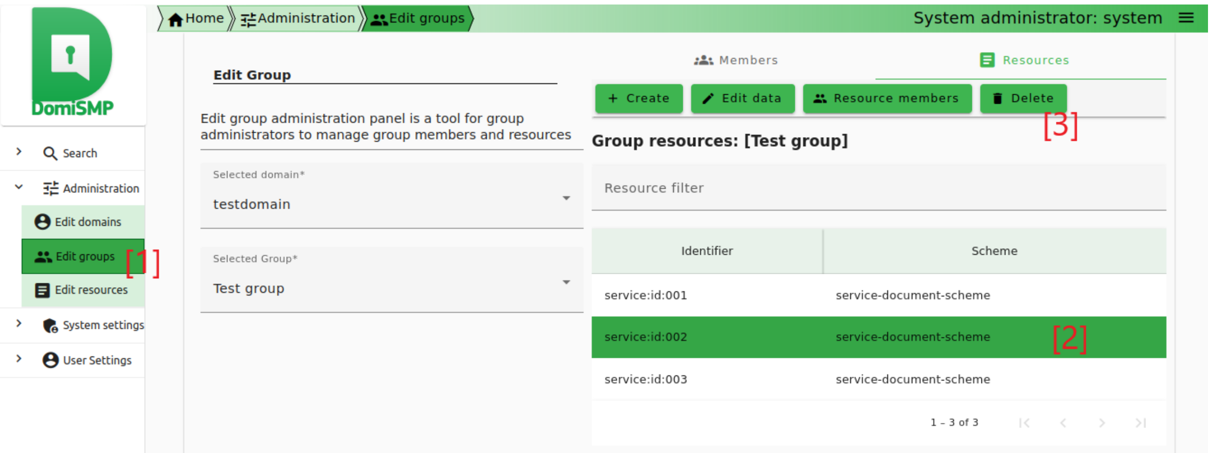
DELETE http://smp.eu/participant-domain-scheme%3A%3Aparticipant-id HTTP/1.1

Accept-Encoding: gzip,deflate

Authorization: Basic c21wX2FkbWluOmNoYW5nZWl0

Content-Length: 0

The Group Admin can delete a ServiceGroup with the DomiSMP UI tool for group administration [1] management. The ServiceGroup can be deleted by selecting the ServiceGroup row [2], clicking the Delete button (see [3] in the figure below).



Edit Group

Edit group administration panel is a tool for group administrators to manage group members and resources

Selected domain*: testdomain

Selected Group*: Test group

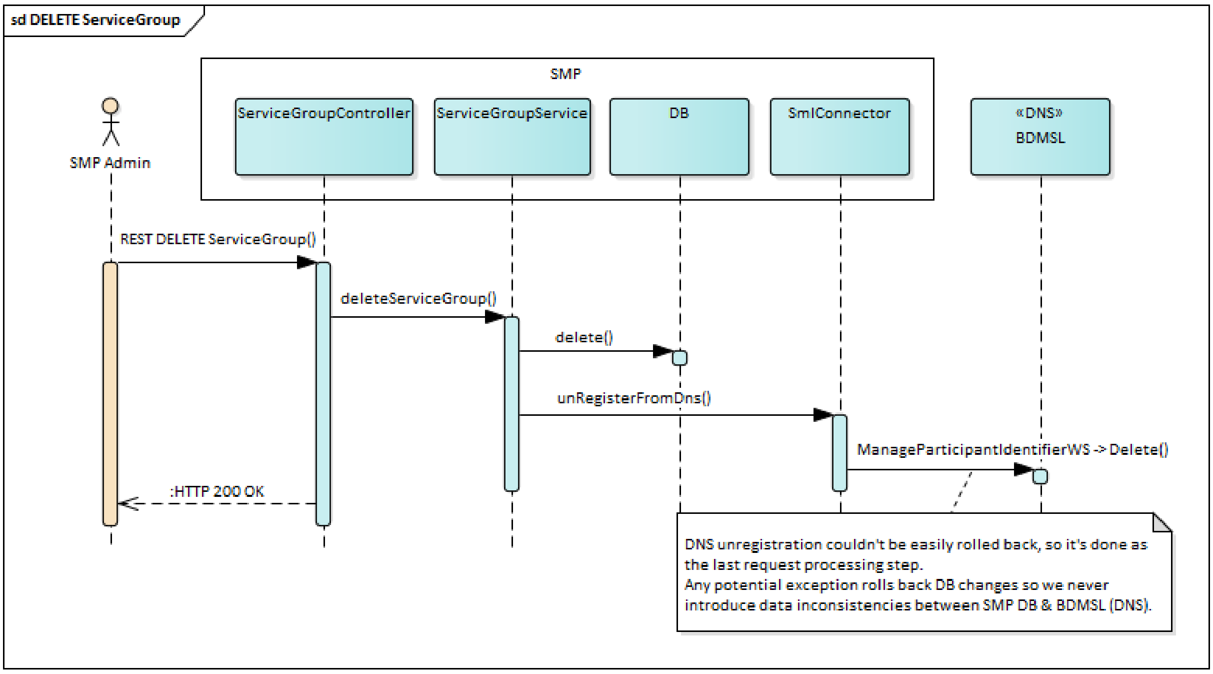
Group resources: [Test group]

Resource filter

Identifier	Scheme
service:id:001	service-document-scheme
service:id:002	service-document-scheme
service:id:003	service-document-scheme

1 – 3 of 3

If BDMSL integration is enabled and configured for the selected domain, the SML delete request is submitted when the ServiceGroup is deleted.


sd DELETE ServiceGroup
SMP
SMP Admin
ServiceGroupController
ServiceGroupService
DB
SmlConnector
«DNS»
BDMSL
REST DELETE ServiceGroup()
deleteServiceGroup()
delete()
unRegisterFromDns()
ManageParticipantIdentifierWS -> Delete()
:HTTP 200 OK
DNS unregistration couldn't be easily rolled back, so it's done as the last request processing step.
Any potential exception rolls back DB changes so we never introduce data inconsistencies between SMP DB & BDMSL (DNS).

1.4.4. UC04 PUT ServiceMetadata

▼ *UC04 PUT ServiceMetadata details*

Create or Update ServiceMetadata

Prerequisites

- The authenticated user has the role of "Resource Admin" (or "Admin SMP").
- Resource Admin user initiating the request is linked to the specified ServiceGroup
- The certificate of the "Resource Admin" is valid.
- The certificate information of the "Resource Admin" was previously stored in the configuration.

Description

`PUT ServiceMetadata` is an idempotent create/update REST action. A sample request is presented below.

NOTE Identifiers present in the body of the request and in the URL marked in yellow must match.

ServiceMetadata is processed and stored as the whole unaltered XML document represented as string (including original whitespaces and comments between nodes). ServiceMetadata can be signed by ServiceGroup owner and e-signature can be placed in <Extension> node. To preserve integrity of signed metadata, SMP does not perform any transformation, canonicalization, or decomposing XML document into separate database records. While querying for the metadata (UC07 – GET ServiceMetadata) original XML document is returned.

Successful responses:

HTTP 200 (OK) – ServiceMetadata was updated HTTP 201 (Created) – New ServiceMetadata was created

PUT http://smp.eu/participant-domain-scheme#%3A%3Aparticipant-id/services/doc-type-scheme#%3A%3Adoc-type-id

HTTP/1.1

Accept-Encoding: gzip,deflate Content-Type: text/xml;charset=UTF-8 Authorization: Basic c21wX2FkbWluOmNoYW5nZWl0 Content-Length: 2152

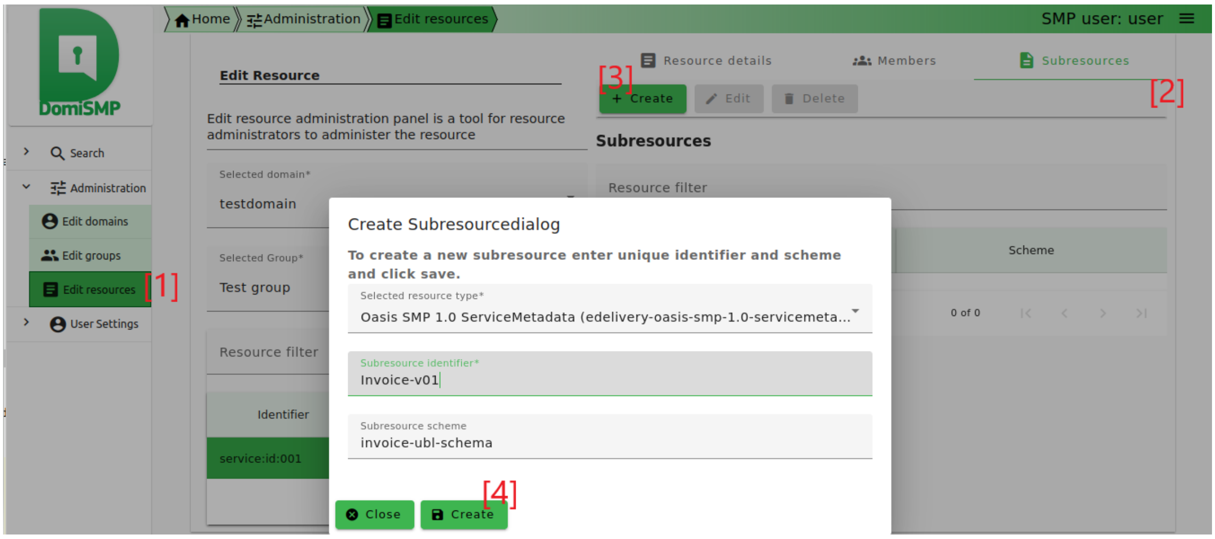
```
<ServiceMetadata xmlns="http://docs.oasis-open.org/bdxr/ns/SMP/2016/05">
<ServiceInformation>
<ParticipantIdentifier
scheme="participant-domain-scheme">
    participant-id
</ParticipantIdentifier>
<DocumentIdentifier scheme="doc-type-scheme">
    doc-type-id
</DocumentIdentifier>
<ProcessList>
<Process>
<ProcessIdentifier scheme=▯process-scheme">"process-id</ProcessIdentifier>
<ServiceEndpointList>

<Endpoint transportProfile="busdox-transport-start">
<EndpointURI>https://poland.pl/theService</EndpointURI>
<RequireBusinessLevelSignature>true </RequireBusinessLevelSignature>
<ServiceActivationDate>2003-01-01T00:00:00</ServiceActivationDate>

<ServiceExpirationDate>2020-05-01T00:00:00</ServiceExpirationDate>
<Certificate>SAMPLEBASE64ENCODEDCERT</Certificate>
<ServiceDescription>Sample description of invoicing service</ServiceDescription>
<TechnicalContactUrl>https://example.com </TechnicalContactUrl>
</Endpoint>

</ServiceEndpointList>
</Process>
</ProcessList>
</ServiceInformation>
</ServiceMetadata>
```

The Resource Admin, can register a ServiceMetadata with the DomiSMP UI tool for Service group management (see [1] in picture below). To add ServiceMetadata, click first on tool Edit Resources (see [1] in picture below), choose the resource and select tab Subresources [2]. Click Create [3] and enter the ServiceMetadata identifiers in the dialog and click Create button [4].



Edit Resource

Edit resource administration panel is a tool for resource administrators to administer the resource

Selected domain*
testdomain

Selected Group*
Test group

Create Subresourcedialog

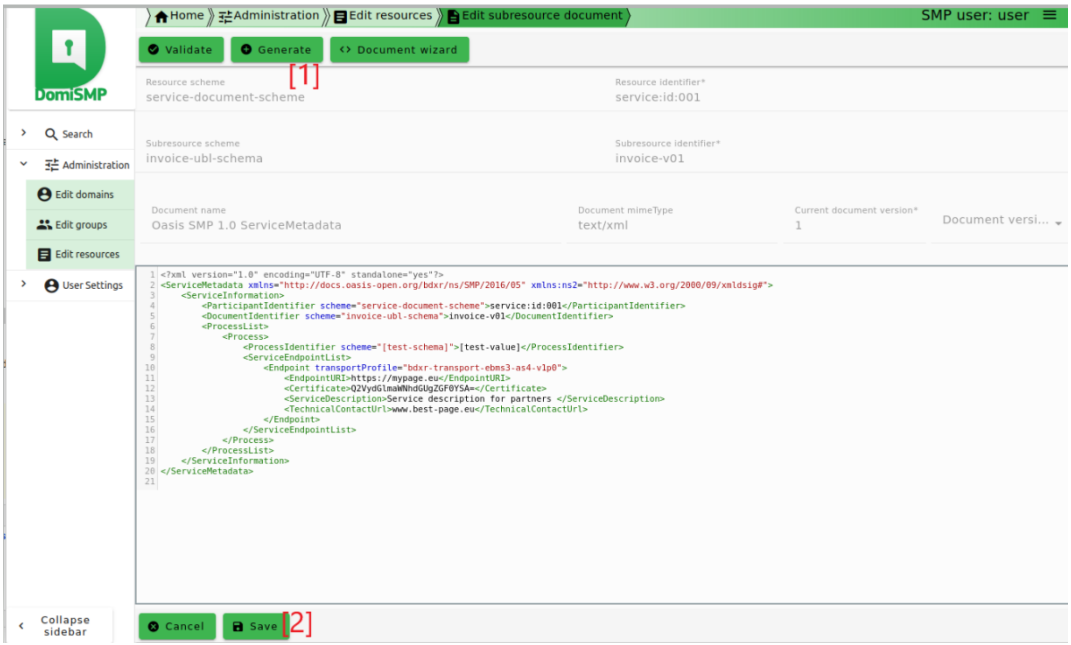
To create a new subresource enter unique identifier and scheme and click save.

Selected resource type*
Oasis SMP 1.0 ServiceMetadata (edelivery-oasis-smp-1.0-servicemeta...

Subresource identifier*
Invoice-v01

Subresource scheme
invoice-ubl-schema

Once the record is created, click the edit button to enter the document editor for adding the ServiceMetadata XML (see the image below). To generate the ServiceMetadata click the button Generate [1] and then save button [3] below.



Resource scheme
service-document-scheme

Resource identifier*
service:id:001

Subresource scheme
invoice-ubl-schema

Subresource identifier*
invoice-v01

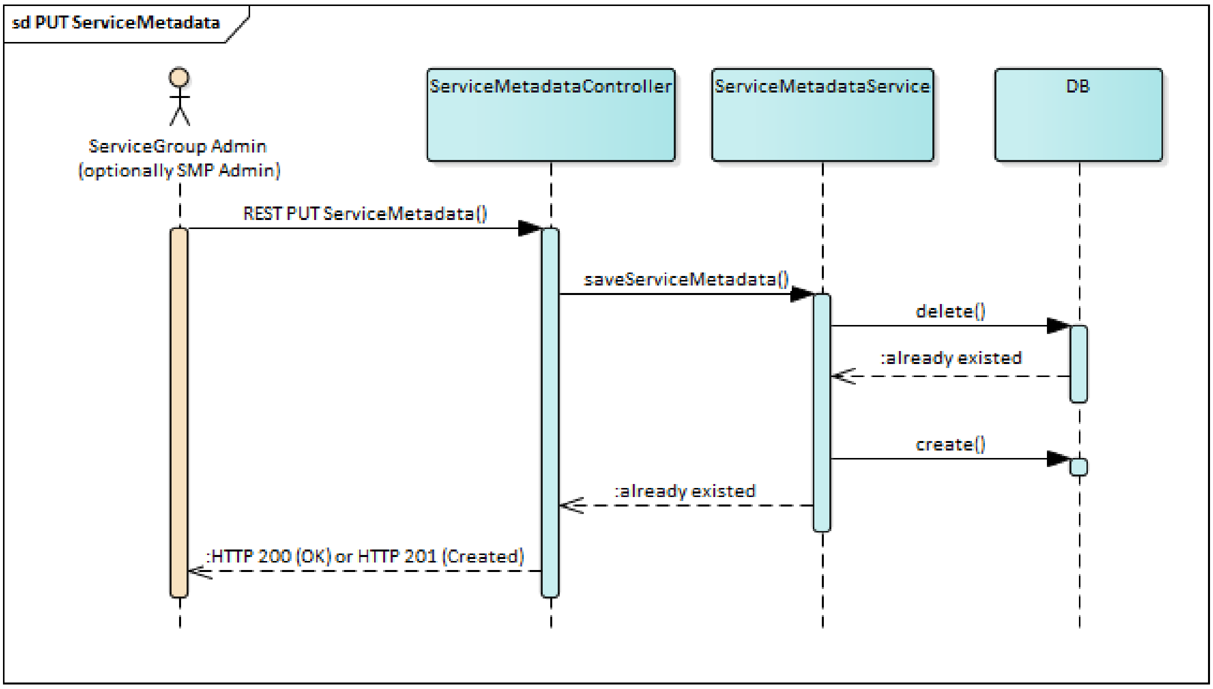
Document name
Oasis SMP 1.0 ServiceMetadata

Document mimeType
text/xml

Current document version*
1

Document versi...

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ServiceMetadata xmlns="http://docs.oasis-open.org/bdxr/ns/SMP/2016/05" xmlns:ns2="http://www.w3.org/2000/09/xmldsig#">
    <ServiceInformation>
        <ParticipantIdentifier scheme="service-document-scheme">service:id:001</ParticipantIdentifier>
        <DocumentIdentifier scheme="invoice-ubl-schema">invoice-v01</DocumentIdentifier>
        <ProcessList>
            <Process>
                <ProcessIdentifier scheme="[test-schema]">[test-value]</ProcessIdentifier>
                <ServiceEndpointList>
                    <Endpoint transportProfile="bdxr-transport-ebms3-as4-v1p0">
                        <EndpointURI>https://mypage.eu</EndpointURI>
                        <Certificate>Q2VydGlmaWNhdGUgZGF0YSA=</Certificate>
                        <ServiceDescription>Service description for partners </ServiceDescription>
                        <TechnicalContactUrl>www.best-page.eu</TechnicalContactUrl>
                    </Endpoint>
                </ServiceEndpointList>
            </Process>
        </ProcessList>
    </ServiceInformation>
</ServiceMetadata>
```


sd PUT ServiceMetadata
ServiceGroup Admin (optionally SMP Admin)
ServiceMetadataController
ServiceMetadataService
DB
REST PUT ServiceMetadata()
saveServiceMetadata()
delete()
:already existed
create()
:already existed
:HTTP 200 (OK) or HTTP 201 (Created)

1.4.5. UC05 DELETE ServiceMetadata

▼ *UC05 DELETE ServiceMetadata details*

Prerequisites

- Resource Admin initiating the request is linked to the specified ServiceGroup (or is "Admin SMP").
- The authenticated user has the role of "Resource Admin".
- The referenced ServiceMetadata exists.

Description

This action removes the specified ServiceMetadata from the SMP’s database. The SMP validates the request and deletes corresponding records.

Successful responses:

HTTP 200 (OK) – ServiceGroup was removed

DELETE http://smp.eu/participant-domain-scheme%3A%3Aparticipant-id/services/doc-type-scheme%3A%3Adoc-type-id HTTP/1.1

Accept-Encoding: gzip,deflate

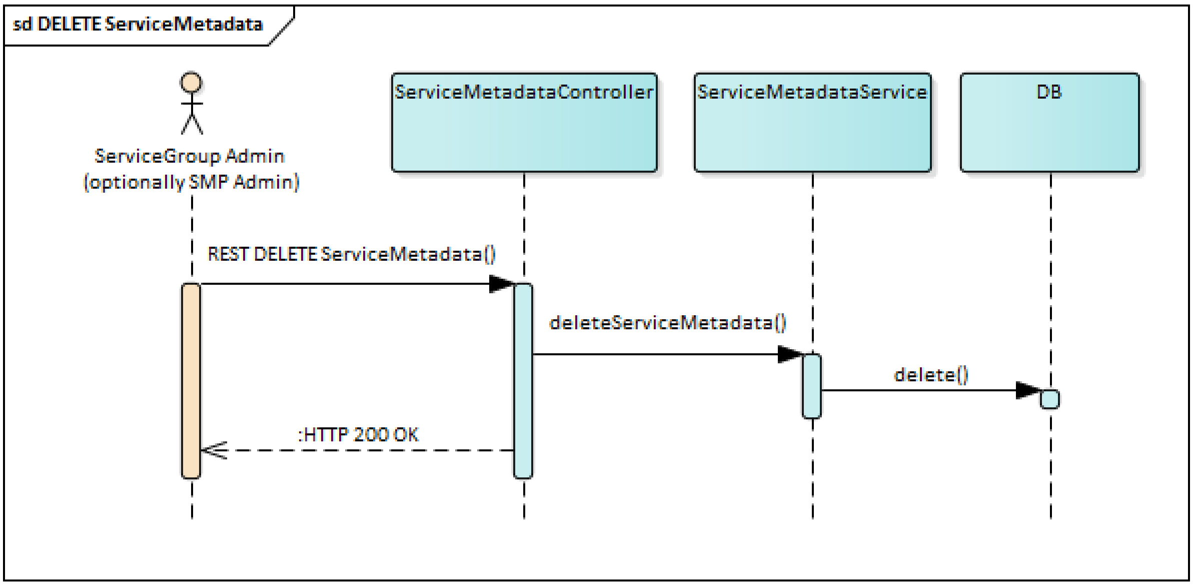
Authorization: Basic c21wX2FkbWluOmNoYW5nZWl0

Content-Length: 0

The Resource Admin, can delete a ServiceMetadata with the DomiSMP UI tool for Editing the Resource s (see [1] in picture below). To delete ServiceMetadata, select the resource which contains the ServiceMetadata [2]. Then in subresources table select the service metadata for the deletion [3]. Finally, click the delete button [4].

image::image12.png[Graphical user interface, application Description automatically generated,width=604,height=282]

Below is the ServiceMedatada delete flow:


sd DELETE ServiceMetadata
ServiceGroup Admin (optionally SMP Admin)
ServiceMetadataController
ServiceMetadataService
DB
REST DELETE ServiceMetadata()
deleteServiceMetadata()
delete()
:HTTP 200 OK

1.4.6. UC06 GET ServiceGroup

▼ *UC06 GET ServiceGroup details*

Prerequisites

- ServiceGroup exists.

Description

The SMP retrieves the details of the specified ServiceGroup from its database including references to all associated ServiceMetadata and returns them in XML format.

GET http://smp.eu/participant-domain-scheme%3A%3Aparticipant-id

HTTP/1.1

Accept-Encoding: gzip,deflate

Successful response:

HTTP/1.1 200

Content-Type: text/xml;charset=UTF-8

Content-Length: 496

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>

<ServiceGroup xmlns="http://docs.oasis-open.org/bdxr/ns/SMP/2016/05">

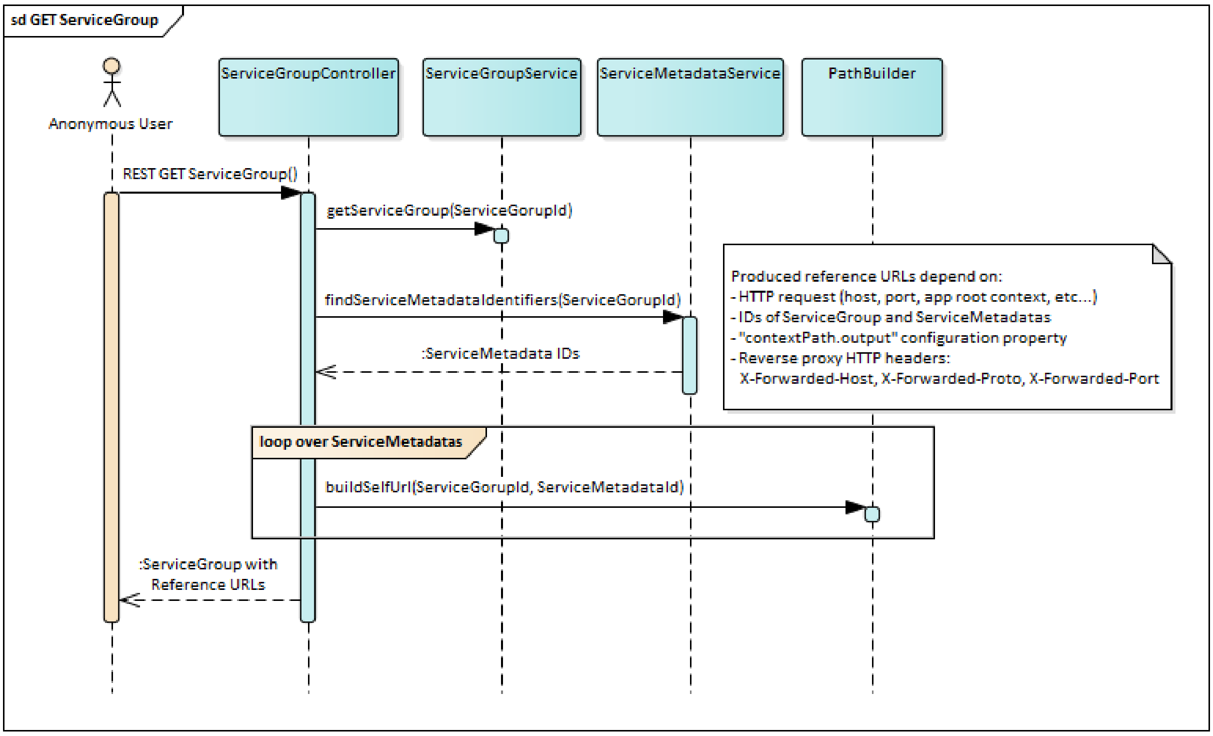
<ParticipantIdentifier scheme="participant-domain-scheme">participant-id</ParticipantIdentifier>

<ServiceMetadataReferenceCollection>

<ServiceMetadataReference href="http://smp.eu/participant-domain-scheme%3A%3Aparticipant-id/services/doc-type-scheme%3A%3Adoc-type-id"/>

</ServiceMetadataReferenceCollection>

</ServiceGroup>



Produced reference URLs depend on:
- HTTP request (host, port, app root context, etc...)
- IDs of ServiceGroup and ServiceMetadatas
- "contextPath.output" configuration property
- Reverse proxy HTTP headers:
X-Forwarded-Host, X-Forwarded-Proto, X-Forwarded-Port

Reference URLs

The URL references inside the <ServiceMetadataReferenceCollection> node refers to the same SMP and can be immediately used by the client to retrieve ServiceMetadata details. Because the SMP is usually deployed behind a ReverseProxy, when the load balancer or the router redirects the request to the backend system, it adds below listed X-Forwarded-* parameters when constructing the URLs:

- X-Forwarded-Host: identifying the original host requested by the client in theHostHTTP request header, since the host name and/or port of the reverse proxy (load balancer) may differ from the origin server handling the request.
- X-Forwarded-Proto: identifying the originating protocol of an HTTP request, since a reverse proxy (or a load balancer) may communicate with a web server using HTTP even if the request to the reverse proxy is HTTPS.

The ReverseProxy can also hide application root context, for instance, if the application is deployed on the server: http://localhost/smp/. Depending on the ReverseProxy configuration, the application can be accessed from internet without root context: http://smp.eu/ or with root context: http://smp.eu/smp/. To properly build the URL, the parameter `contextPath.output` must be set accordingly
(see chapter §5 –"Configuration").

1.4.7. UC07 GET ServiceMetadata

▼ *UC07 GET ServiceMetadata details*

Prerequisites

ServiceMetadata exists in the database.

Description

Service returns details of specified ServiceMetadata from the database. ServiceMetadata is signed and wrapped into the SignedServiceMetadata node.

GET http://smp.eu/participant-domain-scheme%3A%3Aparticipant-id/services/doc-type-scheme%3A%3Adoc-type-id HTTP/1.1

Accept-Encoding: gzip,deflate

Successful sample response with SMP’s XMLDSIG signature marked in dark-grey:

HTTP/1.1 200

Content-Type: text/xml;charset=UTF-8

Content-Length: 4939

<?xml version="1.0" encoding="UTF-8" standalone="no"?>

<SignedServiceMetadata xmlns="http://docs.oasis-open.org/bdxr/ns/SMP/2016/05">

<ServiceMetadata>

<ServiceInformation>

<ParticipantIdentifier scheme="participant-domain-scheme">participant-id</ParticipantIdentifier>

<DocumentIdentifier scheme="doc-type-scheme">doc-type-id</DocumentIdentifier>

<ProcessList>

<Process>

<ProcessIdentifier scheme="cenbii-procid-ubl">urn:www.cenbii.eu:profile:bii04:ver1.0</ProcessIdentifier>

<ServiceEndpointList>

<Endpoint transportProfile="busdox-transport-start">

&lt;EndpointURI&gt;<a href="https://poland.pl/theService&lt;/EndpointURI&gt" class="bare">https://poland.pl/theService&lt;/EndpointURI&gt</a>;

<RequireBusinessLevelSignature>true </RequireBusinessLevelSignature>

```
<ServiceActivationDate>2003-01-01T00:00:00</ServiceActivationDate>

<ServiceExpirationDate>2020-05-01T00:00:00</ServiceExpirationDate>

<Certificate>BASE64ENCODEDSAMPLECERT</Certificate>

<ServiceDescription>Sample description of invoicing service</ServiceDescription>

<TechnicalContactUrl>https://example.com </TechnicalContactUrl>

</Endpoint>

</ServiceEndpointList>

</Process>

</ProcessList>

</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/2001/04/xmldsig-more#rsa-sha256"/>

<Reference URI="">

<Transforms>

<Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"/>

</Transforms>

<DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/>

<DigestValue>BASE64SAMPLEDIGEST</DigestValue>

</Reference>

</SignedInfo>

<SignatureValue>BASE64SAMPLESIGNATUREVALUE</SignatureValue>

<KeyInfo>

<X509Data>

<X509SubjectName>Certificate subject name</X509SubjectName>
```

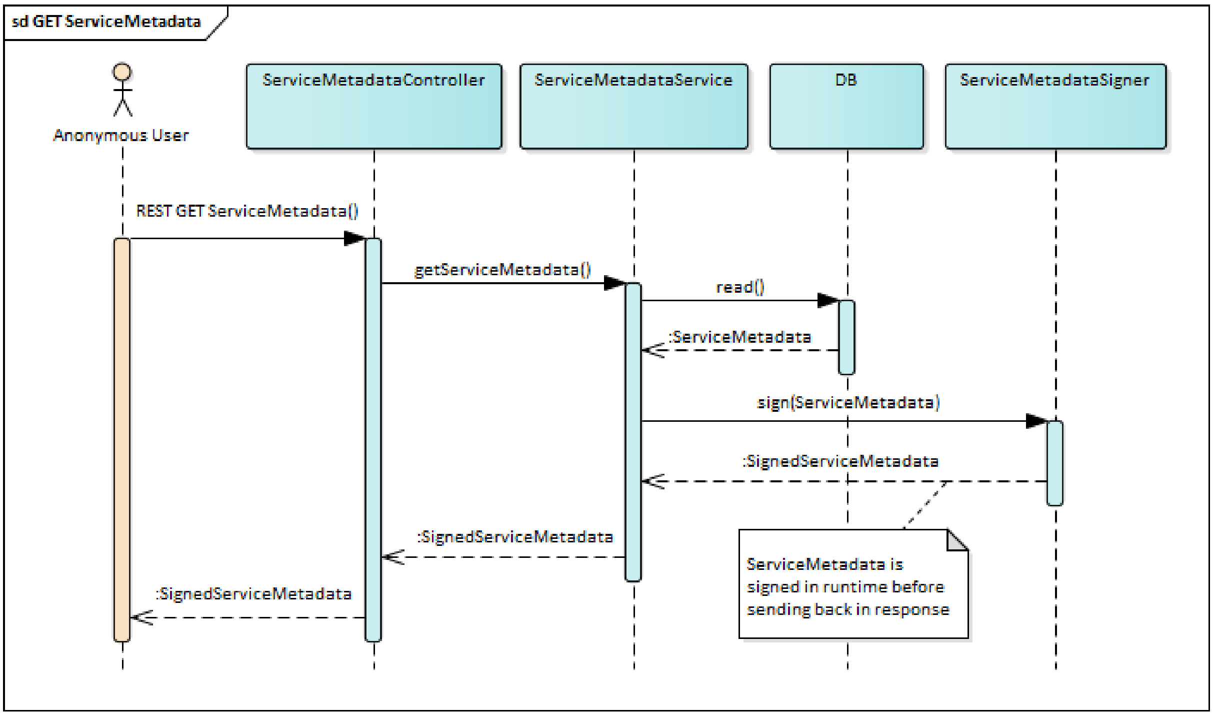
<X509Certificate>BASE64CERTUSEDFORSIGNING</X509Certificate>

</X509Data>

</KeyInfo>

</Signature>

</SignedServiceMetadata>



1.5. Implementation View

1.5.1. Source Code Overview

SMP is a Java REST application shipped and packaged as a `.war` file.

The SMP project uses Maven 3 for its build process and dependency management. Below is a description of SMP’s Maven project structure.

▼ *SMP Project Modules*

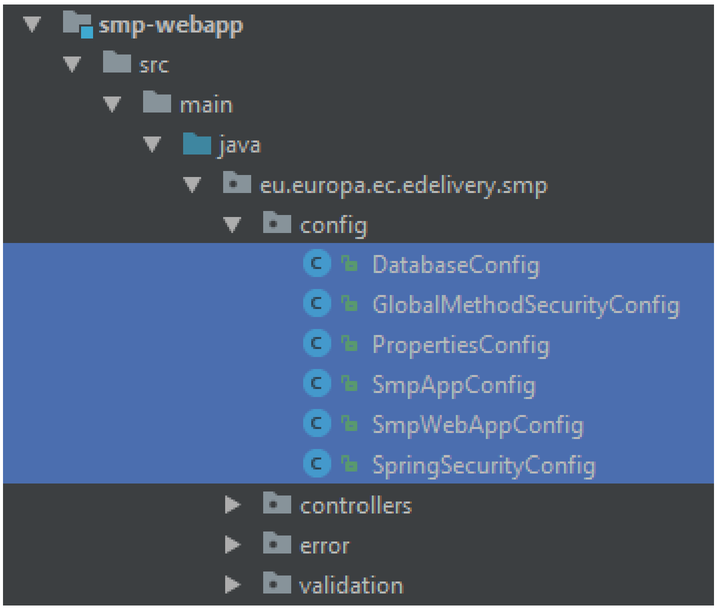
Module	Description
`smp-api`	Module contains OASIS SMP response schemas and administration API schemas. Module purpose is to generate java API classes from predefined XML schemas. Module also contains utility classes used for conversion and validation. This module is used by the SMP REST service implementation and can also be used for building SMP client.
`smp-parent-pom`	Parent POM contains dependency and plugin management used in sub-modules.
`smp-angular`	Angular web fragment for UI.

Module	Description
`smp-server-library`	SMP core library. Covers database access and business logic. This module does not have any HTTP/REST dependencies.
`smp-resource-extension`	The module contains the default resource extensions for Oasis SMP 1.0 and Oasis SMP 2.0 standard.
`smp-soapui-tests`	Module contains Soap UI tests for regression testing in the CI server.
`smp-ui-tests`	Module contains UI regression tests.
`smp-webapp`	REST interface over the core library. Defines REST binding, adds web-specific validations and security. Module also build SMP artefact for deploying to application server and package SMP setting examples, its output is WAR application and ZIP file smp_setup.zip with configuration files and Soap UI test project.
`smp-docker`	Module contains files for building docker images for weblogic/oracle and mysql/tomcat setup. Project also contains compose files to start the setups. The main purpose of the module is to prepare the environment for API and UI integrational testing.
`smp-examples`	The module contains SMP examples of API and SPI implementations. Currently, SPI payload validation example.

1.5.2. Application Skeleton

Spring annotations context setup

The SMP application is built with SpringFramework, the context is set up by classes with @Configuration annotations which are organized hierarchically. List of configuration classes, sample classes defining dependencies, scanning rules in packages and importing another context configuration are presented below.



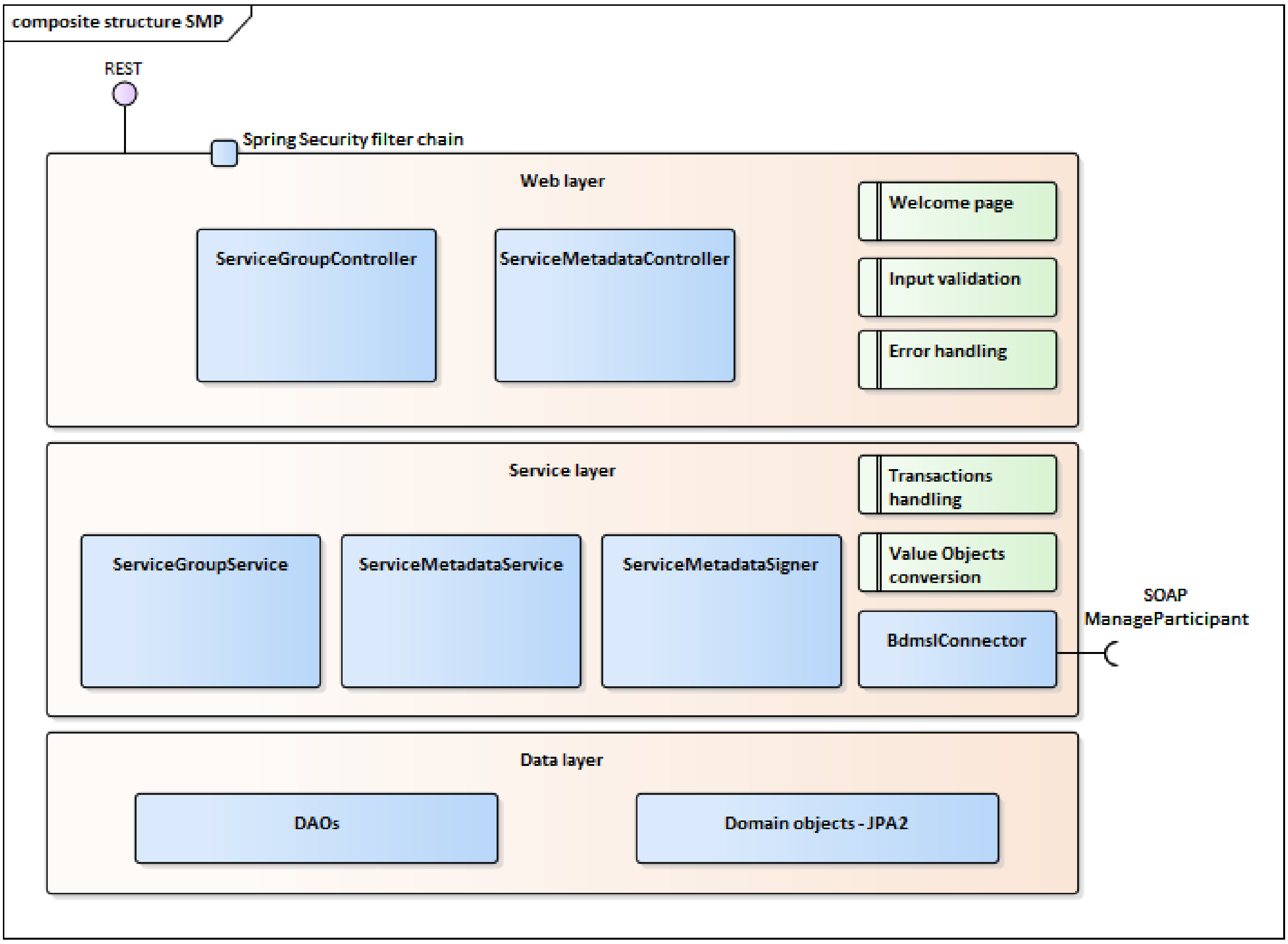
```
▼ smp-webapp
  ▼ src
    ▼ main
      ▼ java
        ▼ eu.europa.ec.edelivery.smp
          ▼ config
              DatabaseConfig
              GlobalMethodSecurityConfig
              PropertiesConfig
              SmpAppConfig
              SmpWebAppConfig
              SpringSecurityConfig
          ▶ controllers
          ▶ error
          ▶ validation
```

@Configuration @ComponentScan(basePackages = \{ "eu.europa.ec.edelivery.smp.validation", "eu.europa.ec.edelivery.smp.services", "eu.europa.ec.edelivery.smp.sml" "eu.europa.ec.edelivery.smp.conversion"})

@Import(DatabaseConfig.class) public class SmpAppConfig \{}

1.5.3. Layers Overview

- Spring MVC
- Business Services Layer
- Data Layer



Spring MVC

REST interface layer

The top layer, implemented within the smp-webapp module, uses Spring MVC's framework. Both resources (ServiceGroup, ServiceMetadata) have a dedicated Controller implementation. Each controller has 3 public methods (GET, PUT, and DELETE) which share the same URL defined by *@RequestMapping* annotation at the Controller class level.

A sample method definition, utilizing also metadata transferred in the request headers is presented below.

This layer is responsible for: REST binding, security validation (more details in §6 – "Security"), request data validation, forwarding request to services layer and forwarding response back to the caller and for error handling.

@RestController @RequestMapping("/{serviceGroupId}") public class ServiceGroupController \{

```
@PutMapping @Secured("ROLE_SMP_ADMIN") public ResponseEntity saveServiceGroup(

@PathVariable String serviceGroupId,

@RequestHeader(name = "ServiceGroup-Owner", required = false) String serviceGroupOwner,

@RequestHeader(name = "Domain", required = false) String domain,

@RequestBody String body) throws XmlInvalidAgainstSchemaException,
UnsupportedEncodingException \{ /* . . . */ }
```

Business Services Layer

The business logic is implemented within the *smp-server-library module*. Business logic is implemented as ServiceGroup and ServiceMetadata Services. Module contains additional classes for Integration with BDMSL, signing messages and transaction handling with use of Spring *@Transactional* annotation and TransactionManager.

Because the SMP is a small application without need of polymorphism, the implementation does not use interface patterns for its services.

Sample Service method definition is presented below:

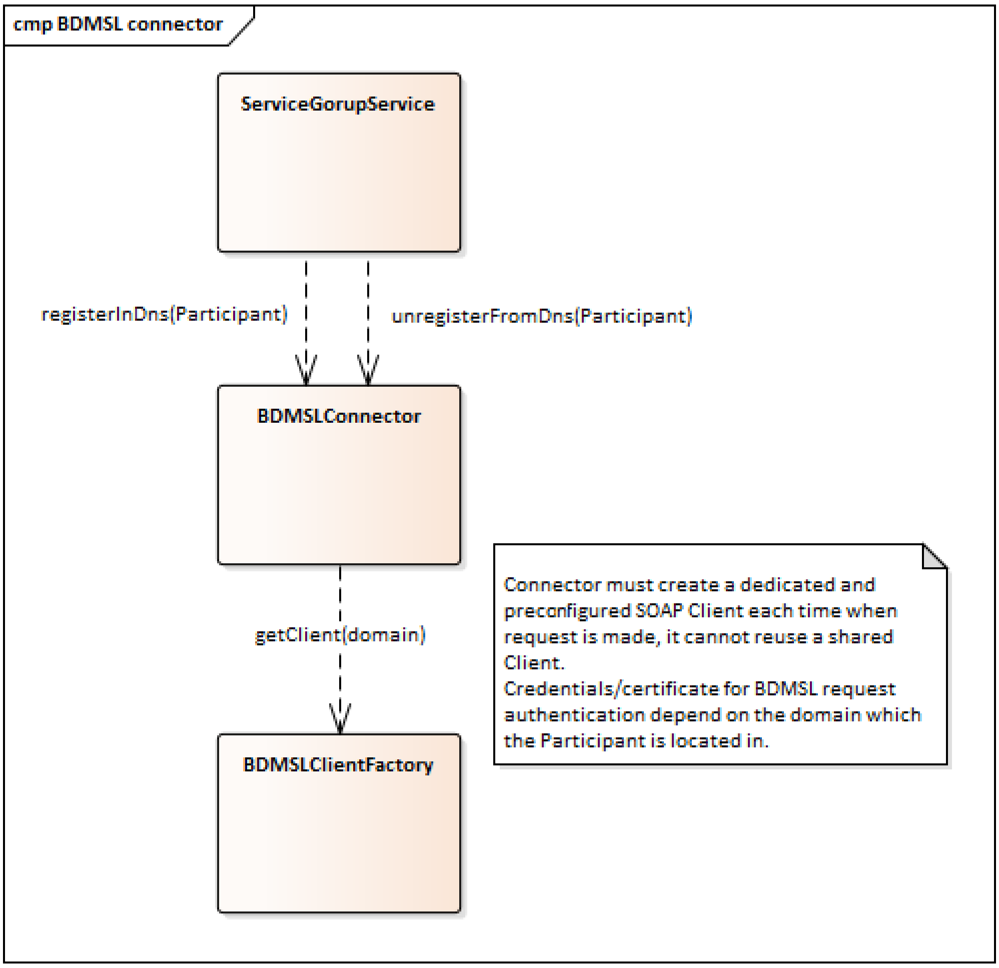
```
@Service public class ServiceMetadataService \{

@Transactional public boolean saveServiceMetadata(ParticipantIdentifierType serviceGroupId,
DocumentIdentifier documentId, String xmlContent) \{ /* . . . */ }
```

BDMSL Integration

The BDMSL integration used by *ServiceGroupService* is implemented by *BDMSLConnector*. Participant’s (un)registration is called synchronously as the last action Service’s method to make sure that any potential *RuntimeException* causes rollback of the whole transaction, including database changes.

To support multiple domains functionality (See chapter §3.3 – "Domain Multitenancy") *BDMSLClientFactory* was introduced. Its responsibility is to create and preconfigure client (*BDMSLConnector*) to set up needed HTTP headers, configure proxy, manage client X509 Certificate, for each domain.


cmp BDMSL connector
ServiceGorupService
registerInDns(Participant)
unregisterFromDns(Participant)
BDMSLConnector
getClient(domain)
BDMSLClientFactory
Connector must create a dedicated and preconfigured SOAP Client each time when request is made, it cannot reuse a shared Client.
Credentials/certificate for BDMSL request authentication depend on the domain which the Participant is located in.

Case Sensitivity Normalisation

Case Sensitivity Support

As functionally described in §4.3.2.2 –"ebCore party identifier":

The eDelivery SMP has the feature to support handling participant identifiers as described in eDelivery SMP profile [REF3] in the chapter “Use with eDelivery ebCore Party Identifiers”. In this case, the participant starts with the: ***urn:oasis:names:tc:ebcore:partyid-type:*** following by the words: **unregistered** or **iso6523.**

All ebCore party identifiers in the REST request must be URL-encoded using only one double-colon separator ":", as in below example:

-

urn:oasis:names:tc:ebcore:partyid-type:iso6523:0088:4035811991021URL-encoded example:

-

urn%23oasis%23names%23tc%23ebcore%23partyid-type%23iso6523%230088%234035811991021 The eDelivery SMP has the option to serialize ebCore party Id to XML according to the OASIS SMP Specification [REF1] as separate values, as in below example:

<ParticipantIdentifier scheme=”urn:oasis:names:tc:ebcore:partyid-type:iso6523:0088” >4035811991021</ParticipantIdentifier>

or according to the eDelivery SMP profile [REF2] as concatenated value:

<ParticipantIdentifier>urn:oasis:names:tc:ebcore:partyid-type:iso6523:0088:4035811991021</ParticipantIdentifier>

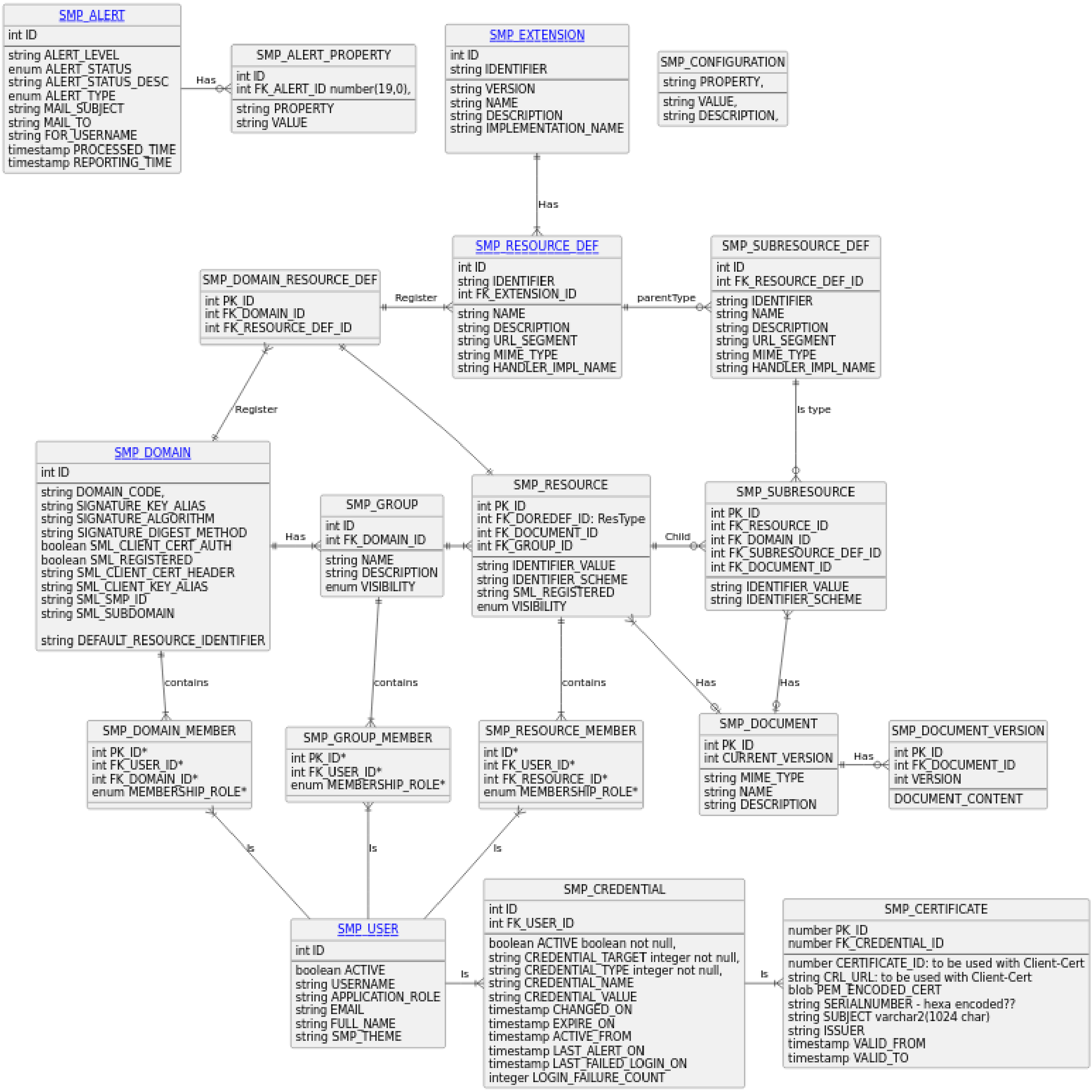
The behaviour can be configured and is explained in more details in §5 – "Configuration".

Identifier's case sensitivity" and §5 – "Configuration" is implemented by the *CaseSensitivityNormalizer* bean. Normalization is performed at the very beginning of each service method processing. Moreover, by separating this to a dedicated bean, normalization can be used as well for permissions verification in connection with Spring Security's *@PreAuthorize anotation*:

@PreAuthorize("hasAnyAuthority('ROLE_SMP_ADMIN', @caseSensitivityNormalizer.normalizeParticipantId(#serviceGroupId))")

Data Layer

The SMP stores data in a relational database. MySQL and Oracle DDL scripts are released with the application in *smp-setup.zip* file. The database object relations are presented in the following figure:



Besides all the necessary metadata used by the DomiSMP business logic, the database is also used to store XML documents in table (oracle: blob, mysql: TEXT type). The Resources and Subresources store versions of the document into the table SMP_DOCUMENT_VERSION. The documents are stored as a binary data because it could be electronically signed by Resource owner. Decomposing and composing XML could compromise the xml signature. When a user is querying for the resource/subresource, the original xml is returned with a valid xml signature.

The Java data access layer is implemented within the *smp-server-library* module. *DataSource, EntityManager* and *TransactionManager* are configured and registered into Spring context in the *DatabaseConfig* class.

Java classes located in *eu.europa.ec.edelivery.smp.data.model* package define the Model with the use of JPA2 annotations. All model classes implement the *BaseEntity* interface. Separate @Embeddable classes are defined for all composite primary keys:

```
@Entity

@Table(name = "smp_service_group")

public class DBServiceGroup implements BaseEntity \{

@EmbeddedId

@Override

public DBServiceGroupId getId() \{ return serviceGroupId; }

/* . . . */

}

@Embeddable

public class DBServiceGroupId implements Serializable \{

@Column(name = "businessIdentifierScheme", nullable = false, length = MAX_IDENTIFIER_SCHEME_LENGTH)

public String getBusinessIdentifierScheme() \{ return participantIdScheme; }

@Column(name = "businessIdentifier", nullable = false, length = MAX_IDENTIFIER_VALUE_LENGTH)

public String getBusinessIdentifier() \{ return participantIdValue; }

/* . . . */

}
```

All DAO classes located in the *eu.europa.ec.edelivery.smp.data.dao* package extend the *BaseDao* generic abstract class that already provides most common DAO operations (find, remove, etc.).

```
@Repository

public class ServiceGroupDao extends BaseDao<DBServiceGroup> \{}

public abstract class BaseDao<E extends BaseEntity> \{

@PersistenceContext

protected EntityManager em;

private final Class<E> entityClass;

public BaseDao() \{

entityClass = (Class<E>) GenericTypeResolver.resolveTypeArgument(getClass(), BaseDao.class);

}

public E find(Object primaryKey) \{

return em.find(entityClass, primaryKey);

}

/* . . . */

}
```

Exception Handling

Detailed functional description of all errors that might occur is presented in the Interface Control Document (cf. [REF4]). This section presents a generalized view on error groups and focuses on implementation perspective.

eDelivery SMP utilizes HTTP error codes according to the best RESTful recommendations, i.e., given codes are always returned for:

- **200 (OK)** or **201 (Created)** – Successful responses (Resource was created/updated/retrieved/deleted).
- **4xx (Bad request)** – Invalid or unauthenticated request.
- **5xx (Server Error)** – SMP technical issue, could be related to configuration, internal networking, integration with BDMSL or DB, etc.

OASIS SMP specification (cf. [REF1]) does not specify error messages, thus eDelivery SMP introduces its own simple XSD with XML namespace: `ec:services:SMP:1.0`. This one describes the structure of error response messages as the sample below:

```
<ErrorResponse xmlns="ec : services:SMP:1.0">

<BusinessCode>NOT_FOUND</BusinessCode>

<ErrorDescription>ServiceMetadata not found, ServiceGroupID: 'x ::y', DocumentID:
```

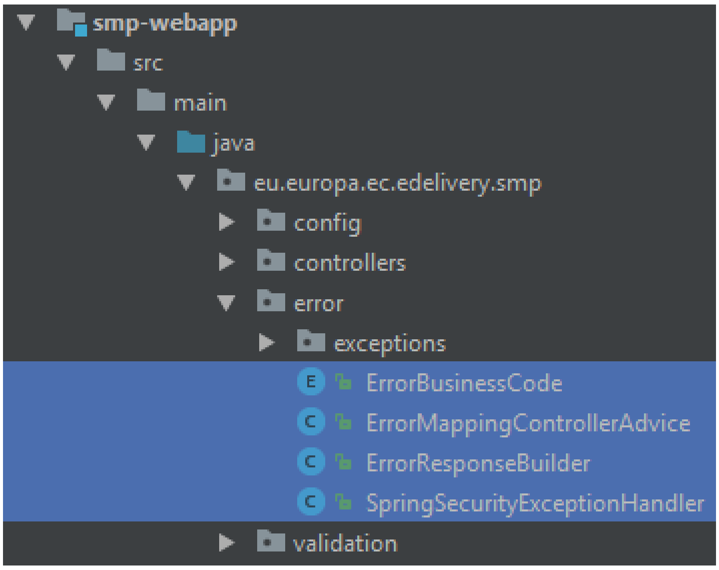
'a::b'</ErrorDescription>

<ErrorUniqueId>2018-03-27T15:07:35.470CEST:d3ba543a-7233-4e69-9f34-655e3998cb3c</ErrorUniqueId>

</ErrorResponse>

▼ *Error handling mechanism implementation*

All classes for processing errors are located in package `eu.europa.ec.edelivery.smp.error`:



```
▼ smp-webapp
    ▼ src
        ▼ main
            ▼ java
                ▼ eu.europa.ec.edelivery.smp
                    ▶ config
                    ▶ controllers
                    ▼ error
                        ▶ exceptions
                          E ErrorBusinessCode
                          C ErrorMappingControllerAdvice
                          C ErrorResponseBuilder
                          C SpringSecurityExceptionHandler
                    ▶ validation
```

▼ *ErrorMappingControllerAdvice*

All backend exceptions are mapped to REST responses within one single class registered in Spring context with *@RestControllerAdvice* and by its many handler-methods annotated with *@ExceptionHandler*. The class uses *ErrorResponseBuilder* and is responsible for:

- mapping exceptions to HTTP response codes and *ErrorBusinessCodes*
- logging user errors as WARN level and technical errors as ERROR level including *uniqueErrorId* for easier maintenance and debugging

Class declaration, sample handler-method (one of many) and internal re-used buildAndWarn method:

```
@RestControllerAdvice

public class ErrorMappingControllerAdvice \{

@ExceptionHandler(NotFoundException.class)

public ResponseEntity handleNotFoundException(NotFoundException ex) \{

return buildAndWarn(NOT_FOUND, ErrorBusinessCode.NOT_FOUND, ex.getMessage(), ex);

}

/* . . . /
private ResponseEntity buildAndWarn(HttpStatus status, ErrorBusinessCode
```

businessCode, String msg, Exception exception) \{ / . . . */ }

}

▼ *ErrorResponseBuilder*

`ErrorResponseBuilder` implementing builder pattern is responsible for building Spring's `ResponseEntity`, based on provided HTTP status code, `ErrorBusinessCode` and text message. Produced response not only is compliant with introduced dedicated XSD, but contains a `uniqueErrorId` that in future problem investigation can be easily found out in log files once user provides error message details.

Every `uniqueErrorId` is built out of:

- **Timestamp** – this information facilitates support and development by specifying when the error occurred and in which rolled log file more details can be found.
- **UUID** – helps in uniquely locating the error stack trace.

2018-03-27T15:07:35.470CEST:d3ba543a-7233-4e69-9f34-655e3998cb3c

▼ *ErrorBusinessCode*

`ErrorBusinessCode` is a simple `Enum` with given values, used by other error-handling classes:

Business error code	Description
XSD_INVALID	Bad request, XML document provided by the user does not pass schema validation
WRONG_FIELD	Bad request, one of the request fields is wrong, e.g., specified Domain does not exist.
OUT_OF_RANGE	Bad request, e.g., specified dates from-to are overlapped.
FORMAT_ERROR	Bad request, e.g., provided identifier format does not comply to OASIS SMP specifications (cf. [REF1])
UNAUTHORIZED	Unauthorized (HTTP 401), the user has no permission to access requested resource.
NOT_FOUND	Bad request, the requested resource does not exist (GET or DELETE).
USER_NOT_FOUND	Bad request, e.g., the newly created ServiceGroup cannot be owned by a user that does not exist.
TECHNICAL	Technical problem on SMP or infrastructure side (BDMSL integration, database etc). This error is always returned with HTTP 500 "Internal Server Error" code. The specific cause of this error is not communicated in the response since Exceptions' messages might eventually reveal sensitive information.

SpringSecurityExceptionHandler

SpringSecurityExceptionHandler is a glue code that allows exceptions thrown by SpringSecurity to be processed by a common exception-handling mechanism. As a result, all security error responses follow the same pattern as other error responses.

SpringSecurity is implemented as a filter chain at the very beginning of the processing of HTTP requests.

1.6. Configuration

SMP configuration (database, keystore, authentication type …) is placed in the property file `smp.config.properties`. File with default values is already included in deployment war package. To override custom values the copy of `smp.config.properties` with updated values must be placed in the application server classpath. More details on configuring classpath can be found in the Administration Guide (cf. [REF3]) and in the §5.1 – "Environment specific configuration".

When the SMP is used in multi-tenancy as described in chapter §3.3 – "Domain Multitenancy", the configuration properties for domain (SMP ID, BDMSL authentication data) are located in database table: SMP_DOMAIN. One record represents one domain, columns represent configuration parameters which are applied for that specific domain. More details on domain configuring can be found in the Administration Guide (cf. [REF3])

Environment specific configuration

Detailed configuration steps for Windows and UNIX systems are covered in the SMP Administration Guide [REF3]. This section is focused explaining the motivation behind particular configuration rather than configuration steps themselves.

1.6.1. WebLogic

Classpath:

The SMP requires configuration file: `smp.config.properties` to be placed in the classpath. On weblogic server custom classpath folder (for example, `/conf_dir_path`) can be set by modifying CLASSPATH variable in scripts `setDomainEnv.sh`:

```
EXPORT CLASSPATH="$CLASSPATH$\{CLASSPATHSEP}/conf_dir_path"
```

Authentication:

WebLogic by default validates username/password (*BasicAuth*) credentials if such are present in any incoming request. Because SMP handles *BasicAuth* with SpringSecurity this feature must be turned off. This is achieved by changing **enforce-valid-basic-auth-credentials** property in **config.xml** file to **false**.

1.6.2. Tomcat

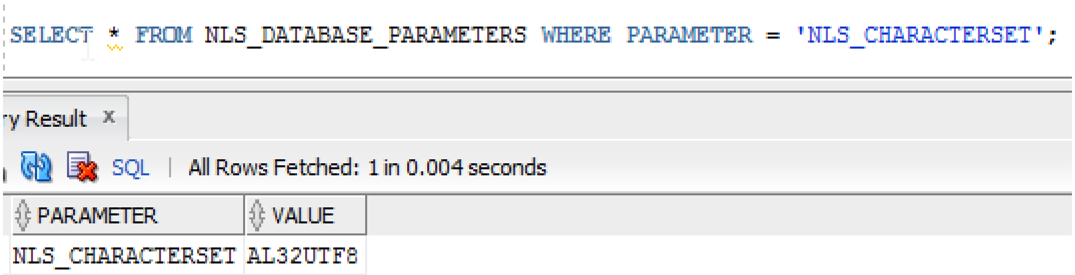
Classpath:

The SMP requires configuration file: **smp.config.properties** to be placed in the classpath. On tomcat server custom classpath folder (e.g. /conf_dir_path) can be set by modifying the starting scripts in the same way as for WebLogic, or by adding this entry in `context.xml` file:

```
<Resources
 className="org.apache.catalina.webresources.StandardRoot"
 cachingAllowed="true" cacheMaxSize="100000">
 <PreResources
   className="org.apache.catalina.webresources.DirResourceSet"
   base="/conf_dir_path"
   internalPath="/"
   webAppMount="/WEB-INF/classes" />
</Resources>
```

1.6.3. Oracle

`NLS_CHARACTERSET` must be set to `AL32UTF8`, otherwise SMP will face issues with non-ASCII characters.

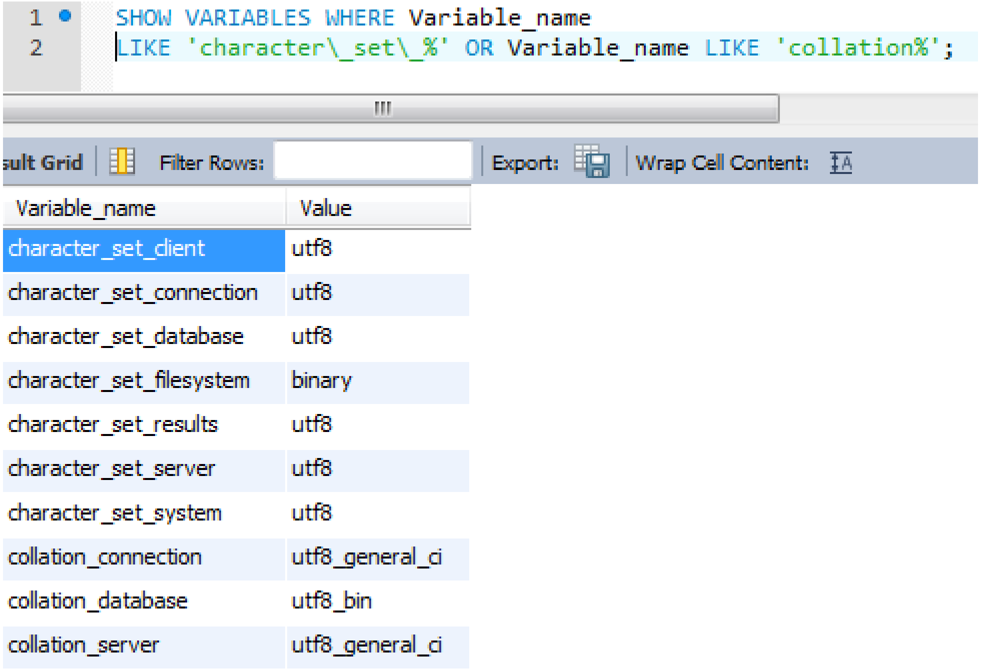


```
SELECT * FROM NLS_DATABASE_PARAMETERS WHERE PARAMETER = 'NLS_CHARACTERSET';
```

PARAMETER	VALUE
NLS_CHARACTERSET	AL32UTF8

1.6.4. MySql

Character set, collation and especially JDBC connection protocol encoding – all must be set to UTF-8, otherwise SMP will face issues with non-ASCII characters.



```
SHOW VARIABLES WHERE Variable_name
LIKE 'character\_set\_%' OR Variable_name LIKE 'collation%';
```

Variable_name	Value
character_set_client	utf8
character_set_connection	utf8
character_set_database	utf8
character_set_filesystem	binary
character_set_results	utf8
character_set_server	utf8
character_set_system	utf8
collation_connection	utf8_general_ci
collation_database	utf8_bin
collation_server	utf8_general_ci

1.7. Security

The SMP is secured with the SpringSecurity. The spring security configuration is executed at the eDelivery startup in the following classes:

- `WSSecurityConfigurerAdapter.java`: class that handles the webservice endpoint security configuration;
- `UISecurityConfigurerAdapter.java`: class that handles the UI endpoint security configuration;
- `SMPCasConfigurer.java`: class that handles the UI Cas configuration.

1.7.1. Authentication

The Authentication Manager (id = `smpAuthenticationManager`) utilizes two Authentication One handles basic username/ password authentication and the second is SpringSecurity implementation `PreAuthenticatedAuthenticationProvider` class configured to handle `X509Certificate` and `BlueCoat` authentication. The pre-authenticated scenarios take precedence over basic authentication. That means if a client provided a valid certificate and also valid username and password, then he is logged in using his certificate and username/password is ignored.

Username and password authentication (Basic Authentication forUI)

Standard SpringSecurity mechanism is used to verify username and BCrypt hashed passwords using the SMPAuthenticationProvider. Username/Password authentication can be used for the UI authentication.

Access token authentication (Basic Authentication for web-services)

eDelivery SMP uses different credentials for UI and for WebService authentication.
The access token is randomly generated access token id and access token value. Together they are used as HTTP basic authentication when invoking the web-services.

Client certificate authentication

Client Certificate authentication can be used only for authentication when invoking the REST API services. The purpose of the certificate authentication is to support mutual 2-way TLS authentication for machine-to-machine integration.

SMP supports two types of Client Certificate authentications: X509 certificate authentication and Authentication behind Reverse Proxy. Both scenarios are performed in 2 steps:

1. Certificate details are extracted to the eDelivery-specific text format. This step is handled by two custom filters: `x509AuthFilter` and `blueCoatReverseProxyAuthFilter`, separately for both scenarios.
2. `PreauthAuthProvider` verifies that if certificate-defined user exists in the database.

`X509Certificate` and Certificates HTTP Client-Cert header are validated with the following attributes:

- Valid from: if “current date” is smaller than “valid from” date, then authentication is rejected
- Valid to: if “current date” is greater than “certificates valid to” date, then authentication is rejected
- Revocation List: certificates are validated by CRL which is downloaded and cached till the CRL “valid to” date. CRL URL endpoint is defined in `SMP_CERTIFICATE.CRL_URL` column and is used for HTTP Client-Cert authentication and for X509Certificate authentication. If the CRL is not reachable, SMP silently ignores the CRL verification, if the configuration attribute

"smp.certificate.crl.force" is set to false. If the attribute is set to true, then Client is not authenticated due to technical issues.

- Truststore: If the SMP truststore is not empty, then formatted issuer or subject is verified if it exists in the truststore. If none of the values exists in the truststore, then certificate authentication is rejected.

Users that are authenticated by certificate are stored in the SMP_USER table, together with users authenticated by password. The USERNAME value of certificate authenticated users is a string value created from parts of certificate distinguish name (DN) and serial number by the following pattern (eDelivery format):

```
CN={common name},O={organisation},C={country}:{16-digit-zero-padded-hex-serial}
```

Example:

```
CN=CEF eDelivery,O=European Commission,C=BE:000000000000c41f
```

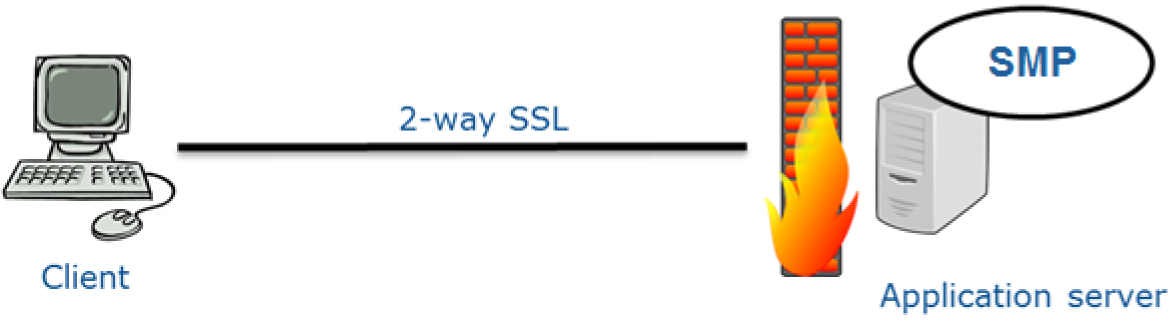
Application distinguished certificate authenticated users from password-authenticated user by an empty PASSWORD column.

Most eDelivery projects supporting client certificate authentication, utilize the same client certificate text representation and BlueCoat Client-Cert HTTP header patterns. For this reason, custom Java code responsible for client certificate authentication has been extracted and released within a separate JAR library; maven dependency gropuId/artifactId:
eu.europa.ec.edelivery/edelivery-springsecurity-2-way-ssl-auth.

X509 certificate authentication

The client X509 certificate authentication uses server's (Tomcat or WebLogic) certificate authentication settings. After the request passes the server validation successfully, *x509AuthFilter* extract certificate details and then authentication proceeds in the way as described above.

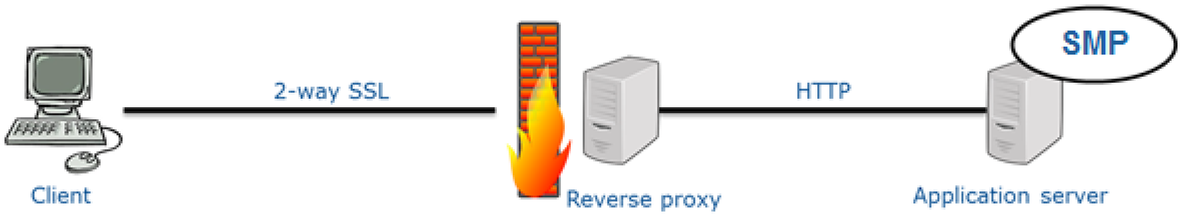
The filter itself (class *EDeliveryX509AuthenticationFilter*) is a simple extension of SpringSecurity's *X509AuthenticationFilter* class, which is a ready-to-use implementation handling *java.security.cert.X509Certificate*.



Authentication behind Reverse Proxy

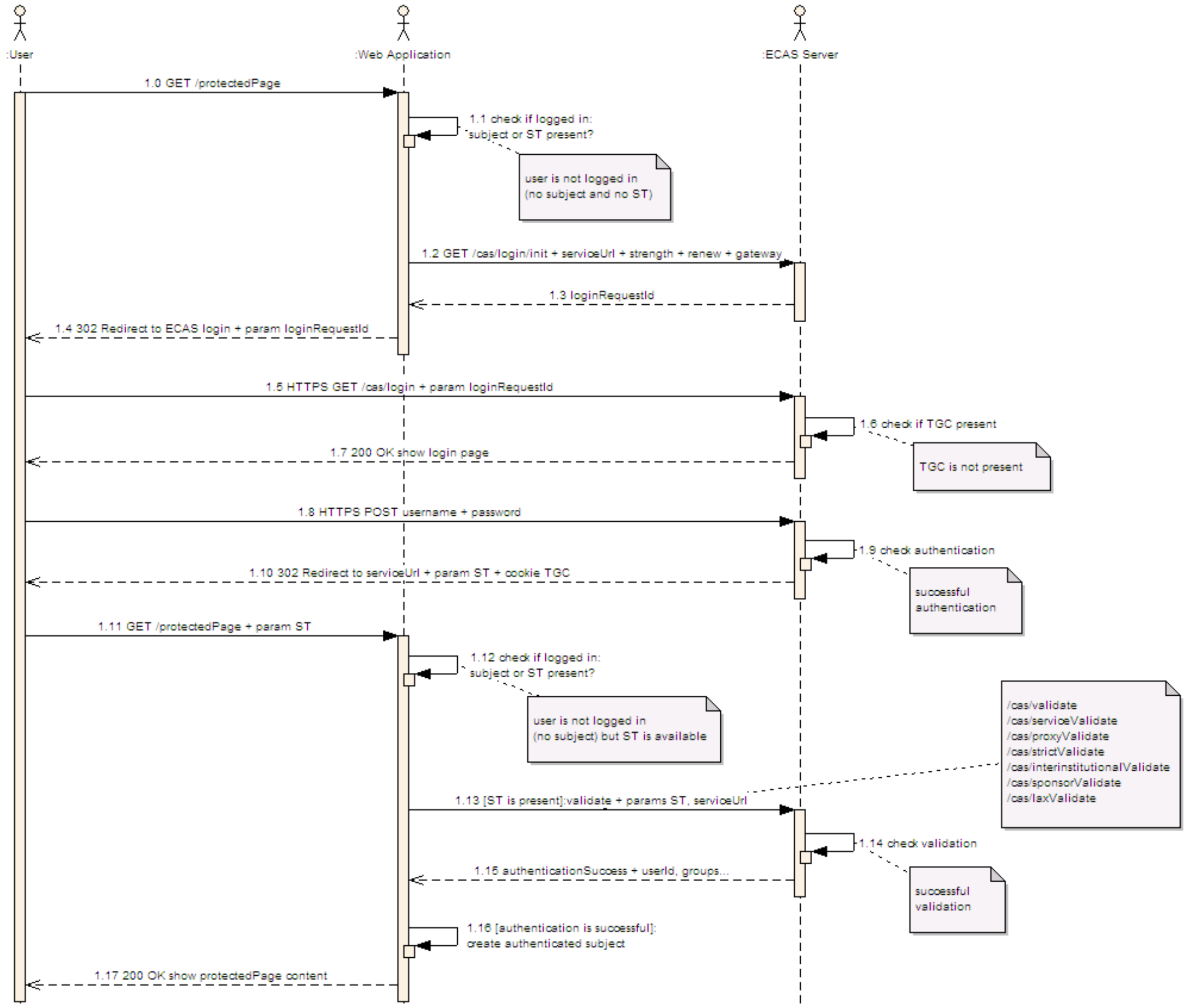
In this setup the basic certificate validation is configured in the BlueCoat reverse proxy. After certificate validation passed successfully, the BlueCoat reverse proxy adds a "Client-Cert" HTTP header and forwards the request to the SMP over HTTP(S). The spring filter *blueCoatReverseProxyAuthFilter* extracts the header, converts it from Bluecoat's to the eDelivery format specified above and then authentication proceeds in the way as described above.

The filter itself (class *BlueCoatAuthenticationFilter*) is based on the SpringSecurity's *RequestHeaderAuthenticationFilter*, dedicated for similar scenarios.



SSO Central Authentication service with EU-LOGIN

CAS authentication can be used only for the UI authentication, and it was made with intention to integrate with ECAS also called EU-Login. ECAS is based on the Central Authentication Service (CAS) version 2 developed at Yale University1. It is an authentication service to protect Web-based applications. SMP was tested only with ECAS, but it should also work with any CAS 2.0 implementation,



:User
:Web Application
:ECAS Server
1.0 GET /protectedPage
1.1 check if logged in: subject or ST present?
user is not logged in (no subject and no ST)
1.2 GET /cas/login/init + serviceUrl + strength + renew + gateway
1.3 loginRequestId
1.4 302 Redirect to ECAS login + param loginRequestId
1.5 HTTPS GET /cas/login + param loginRequestId
1.6 check if TGC present
TGC is not present
1.7 200 OK show login page
1.8 HTTPS POST username + password
1.9 check authentication
successful authentication
1.10 302 Redirect to serviceUrl + param ST + cookie TGC
1.11 GET /protectedPage + param ST
1.12 check if logged in: subject or ST present?
user is not logged in (no subject) but ST is available
/cas/validate
/cas/serviceValidate
/cas/proxyValidate
/cas/strictValidate
/cas/interinstitutionalValidate
/cas/sponsorValidate
/cas/laxValidate
1.13 [ST is present]:validate + params ST, serviceUrl
1.14 check validation
successful validation
1.15 authenticationSuccess + userId, groups...
1.16 [authentication is successful]: create authenticated subject
1.17 200 OK show protectedPage content

When the SMP does not find a service ticket granting access it redirects to EUs login page for user authentication. After user authenticates via the EU login, the response redirects the page back to the SMP UI page with granting ticket.

SMP validates ticket with ECAS. If validation is successful, the SMP authorize access to the user according to user authorization defined on SMP user configuration.

1.7.2. Authorization

Authorities and Roles

Authorities

Authorities in SMP are organized into a two-dimensional space, with Roles as first dimension and **Error! Reference source not found.** as the second one.

Roles

Roles are documented with more details in ICD (cf. [REF4]). The table below explains their meaning from the implementation perspective:

Role alias	Description
ROLE_ANONYMOUS	Any user that has not provided any authentication details.
ROLE_USER	Any authenticated user existing in the database and who doesn't have system admin permissions. Such user is supposed to be a member of the Domain, Group or Resource.
ROLE_SYSTEM_ADMIN	Role for UI enables administration of domains and users.

ICD mentions "System Admin" role, but it's rather a sysadmin, not the business role to be considered in SMP source code.

Authorities execution

Authorities' verification is very flexible thanks to loading all granted authorities to the security context.

HTTP methods: GET/PUT/DELETE

The first level of verification is made on HTTP method level. GET is allowed to everybody, while all modifying actions are allowed only to authenticated users, which is configured in `spring-security.xml` file:

```
<intercept-url method="PUT" access=" ! isAnonymous()" pattern="/*"/>
<intercept-url method="DELETE" access=" ! isAnonymous()" pattern="/*"/>
```

Business object and action level

Once all granted authorities are present in the security context, they are validated at the business methods level with SpringSecurity's annotations and Spring Expression Language (SpEL):

```
@Secured("ROLE_SMP_ADMIN")
```

- action allowed only for Group Admin, or:

```
@PreAuthorize("hasAnyAuthority('ROLE_SMP_ADMIN',
@caseSensitivityNormalizer.normalizeParticipantId(#serviceGroupId))")
```

- action allowed either for Group Admin or Resource admin owing the "serviceGroupId" provided as methods' parameter.

1.8. Quality

SMP quality is supervised by Code Reviews and Continuous Integration processes, which are out of the scope of this document. The quality measurement details presented below focus on technical and source-code point of view.

Unit tests

All utility classes that do not interact with many other classes, which are mostly responsible for conversions, mappings, etc., are unit tested with using Junit and Mockito libraries. Test class name pattern in this case is: `{testedClassName}Test.java`. Tests are run at application build time.

Integration tests

Service classes that combine multiple application modules and in most of the cases require database access are tested in classes with name pattern: `{testedClassName}IntegrationTest.java`.

Tests are executed with JUnit library and configured Spring test context. Also, database instance must be created and defined in maven project files with the following properties:

Property	Description
jdbc.driver	Database Configuration: Driver • MySQL: ◦ com.mysql.jdbc.Driver • Oracle Database: ◦ oracle.jdbc.OracleDriver

Property	Description
jdbc.url	Database Configuration: url - MySQL: jdbc:mysql://dbhost:dbport/smp_database - Oracle Database: jdbc:oracle:thin:@dbhost:dbport:smp_database or jdbc:oracle:thin:@dbhost:dbport/smp_service
jdbc.password	Database User/Password Configuration: User
jdbc.password	Database User/Password Configuration: Password
target-database	Target Database Backend type/Brand: For MySQL, use: MySQL For Oracle Database, use: Oracle
jdbc.read-connections.max	Database Configuration: Max Read Connection

Example:

```
<properties>
 <jdbc.driver>com.mysql.jdbc.Driver</jdbc.driver>
 <jdbc.url>jdbc:mysql://localhost/smp</jdbc.url>
 <jdbc.user>smp</jdbc.user>
 <jdbc.password>smp</jdbc.password>
 <target-database>MySQL</target-database>
 <jdbc.read-connections.max>10</jdbc.read-connections.max>
</properties>
```

SoapUI integration tests

All functionalities are covered with SoapUI integration tests that run REST requests against the SMP and in some cases access the database directly with SQL statements. The SoapUI project can be found in submodule `smp-soapui-tests\soapui\SMP4.0-Generic-soapui-project.xml` file. These tests are bound to maven build and can be activated at build time with maven profile `-Prun-soapui` switch.

Sonar source code statistics

Maven build is configured to collect standard Sonar code statistics (code test coverage, static code analysis, etc). Apart from that, code test coverage is gathered also when running SoapUI tests. This requires a manual installation of Jacoco Agent in JRE with J2EE container where the SMP is deployed and pointing to this agent when running a build by adding these attributes to maven run:

`-DjacocoRemotePort=65000 □DjacocoRemoteAddress`

Once build with SoapUI tests is done, statistics from all the sources are gathered by sonar plugin by running `mvn sonar:sonar` goal.

1.9. Technical requirements

This section describes the minimum and recommended system requirements to operate the SMP component.

Hardware

Type	Minimum	Recommended
Processor	1 CPU core	4 CPU core
Memory (RAM)	2GB	8GB or more
Disk space	5GB	Depends on usage

Recommended stack

- Ubuntu 22.04 LTS 64 bits
- Oracle Java EE 8
- MySQL 8

Operating Systems and Software

OS

Any operating system that is compliant with the supported JVM.

Java Virtual Machines

- Oracle Java JRE 8/11

Java Application Servers

- Apache Tomcat 9.x
- Oracle WebLogic Server 12.2c or 14.1C

Databases

- MySQL 8
- Oracle Database 19c

Web Browsers

n/a

Chapter 2. Administration Guide

Contents

This guide provides information on how to:

- **Deploy and configure SMP on supported application servers and databases.**
  See Prerequisites and Relevant Resources.
- Perform relevant security configurations (certificates).
- Consume the Soap UI to create, update and delete SMP Service Groups and Metadata and an alternative method to perform creation, update and deletions operations using Swagger UI.

Target Audience

This guide is intended for Administrators who are in charge of installing, managing and troubleshooting an eDelivery SMP.

2.1. Prerequisites and Relevant Resources

Software Requirements

SMP requires:

- one supported Java Runtime Environment (JRE)
- one supported Webserver
- one support Database Management Systems (DBMS)

Supported versions of required software for SMP

Java Runtime Environment	
	JRE 8 or 11: Download it here.
Webservers	
	Weblogic • Weblogic 12.2.1.4 (tested with JDK 8) or higher • Weblogic 14.1c (tested with JDK 11) or higher *Tomcat* • Tomcat 9.x (tested with JDK 8) or higher
Databases	
	• MySQL 8.0.x * tested version, future versions might also work • Oracle 11g XE and Oracle 19c * tested version, future versions might also work

NOTE For more information and installation details for third-party software, refer to their specific documentation.

Binaries Repository

The DomiSMP artefacts can be downloaded from the eDelivery Digital Portal.

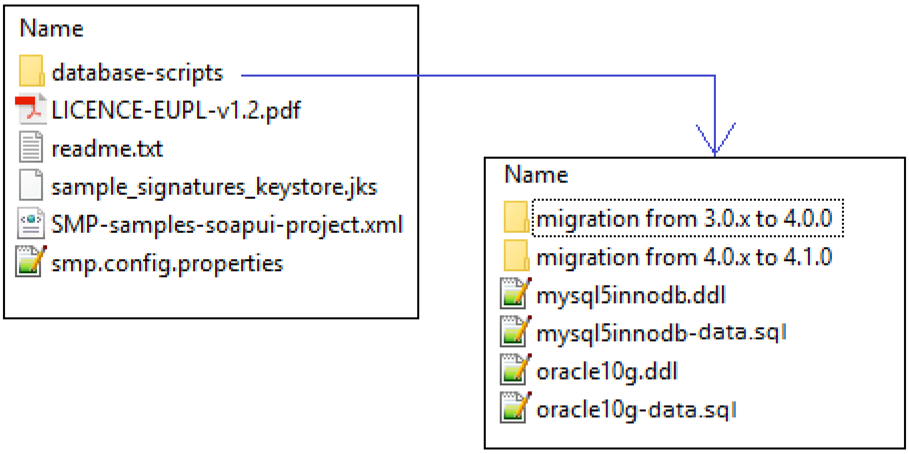
Source Code Repository

The source code of eDelivery DomiSMP is available in the **GIT** repository at the following location: https://ec.europa.eu/digital-building-blocks/code/projects/EDELIVERY/repos/smp/browse

+ image::image3.png[width=627,height=453]

Database Scripts

The scripts to create (or migrate) the Oracle or MySQL databases are included in the following downloadable zip file from the Digital site: `smp-x-setup.zip`. See Downloading the Source Code.



Name
- database-scripts
- LICENCE-EUPL-v1.2.pdf
- readme.txt
- sample_signatures_keystore.jks
- SMP-samples-soapui-project.xml
- smp.config.properties

Name
- migration from 3.0.x to 4.0.0
- migration from 4.0.x to 4.1.0
- mysql5innodb.ddl
- mysql5innodb-data.sql
- oracle10g.ddl
- oracle10g-data.sql

2.2. Deployment Overview

▼ *SMP Deployment Steps Overview*

As mentioned in the prerequisites, the deployment of the SMP is only supported on Tomcat or WebLogic application servers.

The deployment of the SMP on both platforms is almost identical but minor platform specific changes will be documented in a dedicated section of this manual.

The deployment of the SMP is summarized in the following mandatory steps:

- STEP1 Database Configuration
- STEP2 Application Server Preparation (Weblogic and Tomcat) for SMP
- STEP3 SMP Initial Configuration
- STEP4 SMP .war file Deployment

NOTE The environment variable, `AS_HOME`, refers to the application server home folder where the SMP package is installed.

- For Tomcat, it refers to `CATALINA_HOME`.

- For Oracle WebLogic, it refers to DOMAIN_HOME.

NOTE

The environment variable, SETUP_PATH, refers to the folder where the deployment SMP package smp-4.x-setup.zip is extracted.

▼ *Folder structure*

The following subdirectories must be created in the AS_HOME directory. The document describes the default folder settings and can be named or created in a location other than the AS_HOME directory.

AS_HOME/smp

the folder contains the basic SMP settings, and the folder must be configured as a classpath; see sections:

- Configuring Extra Class Path in WebLogic
- Configuring Extra Class Path in Tomcat

AS_HOME/logs

the purpose of the folder is to contain SMP logs.

AS_HOME/security

the previous versions of the SMP have security artifacts (truststore, keystore, etc.) under the /smp folder. We recommend creating a separate folder for a more transparent handling of the security artifacts. In case of setting SMP in an application server cluster, this folder must be shared among the cluster nodes.

The location of the folder must be set in the SMP application property: smp.security.folder.

NOTE

before DomiSMP 5.0 version, the application property's name was configuration.dir.

2.3. STEP1 Creating the Database

This section describes the steps necessary to create the database, tables and the SMP database user (dbuser used for database connection purpose).

It also includes the creation of an initial SMP user account that will be used by REST clients to connect to the SMP.

The SMP uses a direct connection to the database, which removes the need to configure a data source within WebLogic.

For this step you need to use the script included in the zip file downloaded in section §3.3 Database Scripts. (check here)

2.3.1. MySQL

1. Open a command prompt and navigate to the SETUP_PATH/sql-scripts folder

2. Execute the following MySQL commands:

```
mysql -h localhost -u root_user □-password=root_password -e \
"DROP SCHEMA IF EXISTS <smp_schema>;
CREATE SCHEMA <smp_schema>;
ALTER DATABASE <smp_schema> charset=utf8;
CREATE USER smp_dbuser@localhost IDENTIFIED BY 'smp_password';
GRANT ALL ON <smp_schema>.* TO smp_dbuser@localhost;"
```

   This creates a <smp_schema> and an <smp_dbuser> with all privileges to the <smp_schema>.

3. Execute the following command to create the required objects (tables, etc.) in the database:

```
mysql -h localhost -u <root_user> --password=<root_password> <smp_schema> <
mysql5innodb.ddl
```

   Script download:

   - mysql5innodb.ddl

4. Execute the following command to fill initial test data:

```
mysql -h localhost -u root_user --password=root_password smp_schema < mysql5innodb-
data.sql
```

2.3.2. Oracle Database

1. Navigate to `SETUP_PATH/sql-scripts` directory

2. Execute the following commands :

```
sqlplus sys as sysdba ①
```

   Where:

   ① Password is the one defined during the Oracle installation.

3. Once logged in Oracle:

```
CREATE USER <smp_dbuser> IDENTIFIED BY <smp_dbpassword>;
GRANT ALL privileges TO <smp_dbuser>; connect <smp_dbuser>
show user; ①
@oracle10g.ddl ②
@oracle10g-data.ddl ③
exit
```

   ① Expected return: `<smp_dbuser>`.

② Run the scripts with the `@` sign from the location of the scripts.

③ Fill initial test data.

Scripts Download:

- oracle10g.ddl
- oracle10g-data.ddl.

2.4. STEP2 Configuring the Server

- Weblogic/Oracle
- Tomcat

2.4.1. Configuring WebLogic/Oracle

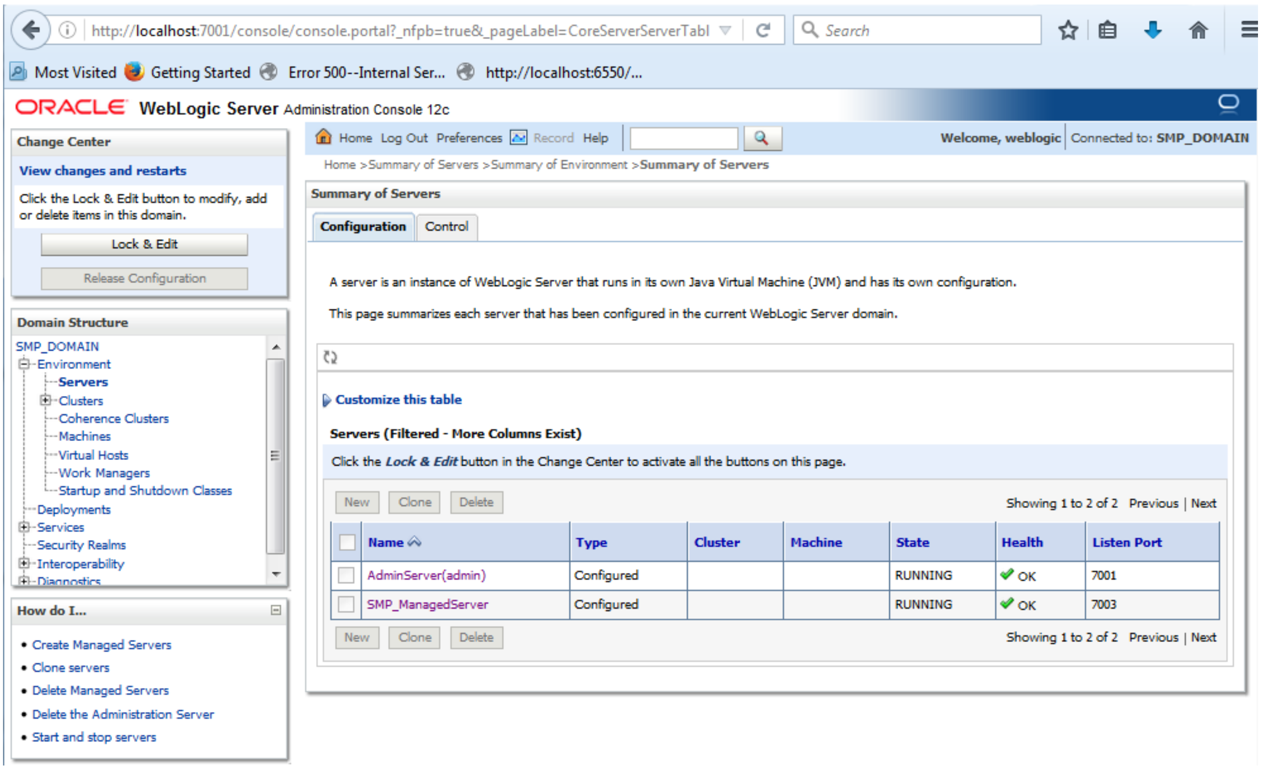
This section does not include the installation of a WebLogic application server. It is assumed that the WebLogic Server is installed, and a WebLogic domain is created with an administration server and a managed server on which the SMP will be deployed.

Hereafter the domain location will be referred as `DOMAIN_HOME` (user-defined name).

In the examples below, we will use the following Domain and Server names:

- Domain Name : SMPDOMAIN
- Administration Server : AdminServer
- SMP Managed Server : SMP_ManagedServer

As shown below:



Click the Lock & Edit button to modify, add or delete items in this domain.

A server is an instance of WebLogic Server that runs in its own Java Virtual Machine (JVM) and has its own configuration.

This page summarizes each server that has been configured in the current WebLogic Server domain.

Servers (Filtered - More Columns Exist)

Click the *Lock & Edit* button in the Change Center to activate all the buttons on this page.

□	Name	Type	Cluster	Machine	State	Health	Listen Port
□	AdminServer(admin)	Configured			RUNNING	✓ OK	7001
□	SMP_ManagedServer	Configured			RUNNING	✓ OK	7003

To deploy the SMP on the WebLogic Application Server platform, two preliminary steps need to be completed:

- Disabling the application basic Authentication on the Weblogic Server,
- Configuring the Extra CLASSPATH for WebLogic,
- Setup sun HTTP Handler.

  This is described in the following two sections.

Disabling Authentication

By default, WebLogic performs its own basic authentication checks requests before passing the request to deployed application (e.g. eDelivery SMP). The eDelivery SMP has its own authentication mechanism that makes the WebLogic authentication redundant, and it is therefore important to disable the WebLogic Authentication to stop it from interfering with the SMP authentication.

To do so, edit the `config.xml` file in `SMPDOMAIN/config` by adding the `</security-configuration>` closing tag:

```
<enforce-valid-basic-auth-credentials>false</enforce-valid-basic-auth-credentials>
```

Here is an example:

```
<!-- ... -->
    <enforce-valid-basic-auth-credentials>false</enforce-valid-basic-auth-credentials>
</security-configuration>
<!-- ... -->
```

Configuring Extra CLASSPATH

in WebLogic

Here we describe how to set up the smp folder as a classpath on the WebLogic server. See also Deployment Overview).

Edit the WebLogic `DOMAIN_HOME/bin/setDomainEnv.sh`.

Linux:

Add the `EXPORT CLASSPATH=${CLASSPATH}:${DOMAIN_HOME}/smp` statement at the end of the CLASSPATH definition as shown below:

```
../

if [ "${PRE_CLASSPATH}" != "" ] ; then
fi
/..
```

Windows:

```
../
If NOT "%PRE_CLASSPATH%"=="" (
set CLASSPATH=%PRE_CLASSPATH%;%CLASSPATH% )
set CLASSPATH=%CLASSPATH%;%DOMAIN_HOME%\smp /..
```

Configuring Sun HTTP handler

Edit the WebLogic `DOMAIN_HOME/bin/setDomainEnv.sh` and add the following system parameter.

```
../
JAVA_OPTIONS=-DUseSunHttpHandler=true export JAVA_OPTIONS
/..
```

2.4.2. Configuring Tomcat

To deploy the SMP on Tomcat, the steps below need to be completed.

Configuring Extra CLASSPATH

im Tomcat

The purpose of the section is to describe how to set up folder smp as a classpath on the Tomcat server.
See also Deployment Overview.

Linux:

Edit the `CATALINA_HOME/bin/setenv.sh` file.

```
#!/bin/sh
# Set CLASSPATH to include $CATALINA_HOME/smp
# where the smp “smp.config.properties” is located export CLASSPATH=$CATALINA_HOME/smp
```

For Windows:

Edit the `%CATALINA_HOME%/bin/setenv.bat` file.

```
REM Set CLASSPATH to include $CATALINA_HOME/smp
REM where the “smp.config.properties” is located
set classpath=%classpath%;%catalina_home%\smp
```

JDBC Driver

The JDBC driver needs to be downloaded from the manufacturer website:

- For Oracle Database : https://www.oracle.com/database/technologies/appdev/jdbc-downloads.html

- For Mysql : https://www.mysql.com/products/connector/

The JDBC driver (`.jar` file) must be copied to `AS_HOME/lib`.

2.5. STEP3 Configuring SMP

2.5.1. SMP Configuration Resources

The SMP configuration is performed in two different locations, the:

- `smp.config.properties` file
- `smp_configuration` table

▼ *Properties Overview*

The DomiSMP 5.0.x configuration has two types of properties:

- The system configuration properties: the properties are located in `smp.config.properties` file and define environment settings such as JDBC connection, logging configuration, SMP extension library folder, etc. Before the first eDelivery SMP startup, the mandatory database connection properties must be set. The complete system property list is described in §15.2.1 .
- The SMP application properties: the property list with default values is stored at initial startup in the database table **SMP_CONFIGURATION**. System administrators can change most properties during the runtime without application restart. The complete application property list is described in section §15.2.2. In case we want to set different init value for particular property at first SMP startup, the property can be set in the `smp.config.properties`.

For this step, use the `smp.config.properties` example delivered within the zip file downloaded in section §3.2.
The `smp.config.properties` file must be copied to the CLASSPATH folder configured in $7.1 for Tomcat and §6.2 for WebLogic).

▼ *Multitenancy and Multidomain Support*

The SMP is able to support multiple certificates in the same SMP. This is very useful in the Acceptance environment where multiple domains like ISA ITB, eHealth and others are hosted.

The SMP has the capability of keeping a relationship between a particular **Service Group** and its related **domain**.

As a result of this feature, the SMP Administration has the option, if need be, to define extra domains for newly created **Service Groups** meaning that the SMP can handle multiple domains environments.

NOTE	In normal circumstances, when any one SMP is used for only one domain, the domain used is then considered as the "domain by default" (or "default domain") for configuration purposes. The domain, in this case, does not need to be specified in the **Service Group** definitions or other configurations of the SMP as in previous versions of SMP.

The SMP configuration is performed in two different locations: in the smp.config.properties file as well as in the smp_configuration table. The following section describes the details of the parameters that are included in the configuration.

2.5.2. Properties Configuration File

The eDelivery SMP configuration is performed via the smp.config.properties file.

The initial eDelivery SMP configuration is performed via the smp.config.properties file. The file contains basic configuration for defining the database connection, logging file configuration and smp folder for deploying the extensions.

This file is delivered by default embedded within the SMP .war file.

▼ *Sample of* smp.config.properties *file*

```
#
# Copyright 2018 European Commission | CEF eDelivery
#

# Licensed under the EUPL, Version 1.2 or - as soon they will be approved by
# the European Commission - subsequent versions of the EUPL (the "Licence");
# You may not use this work except in compliance with the Licence.
#
# You may obtain a copy of the Licence attached in file: LICENCE-EUPL-v1.2.pdf
#
# Unless required by applicable law or agreed to in writing, software
distributed under the Licence is distributed on an "AS IS" basis,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
# See the Licence for the specific language governing permissions and
limitations under the Licence.
#
#
**************************************************************************

# Database connection can be achieved using custom datasource configuration
# or reusing application server datasource.
#
**************************************************************************

## set database hibernate dialect
# smp.database.hibernate.dialect=org.hibernate.dialect.Oracle10gDialect
smp.database.hibernate.dialect=org.hibernate.dialect.MySQL5InnoDBDialect

# *********************************
# Custom defined datasource
# *********************************

# mysql database example
smp.jdbc.driver = com.mysql.jdbc.Driver
smp.jdbc.url = jdbc:mysql://localhost:3306/smp
smp.jdbc.user = smp
smp.jdbc.password=secret123

# Oracle database example
# smp.jdbc.driver = oracle.jdbc.driver.OracleDriver
# smp.jdbc.url=jdbc:oracle:thin:@localhost:1521/xe
# smp.jdbc.user=smp
# smp.jdbc.password=secret123

# *********************************
# Datasource JNDI configuration alternative
# *********************************

# weblogic datasource JNDI example
# smp.datasource.jndi=jdbc/eDeliverySmpDs
# tomcat datasource JNDI example
# smp.datasource.jndi=java:comp/env/jdbc/eDeliverySmpDs

# *********************************
# Logging properties
# *********************************
# smp log folder
smp.log.folder=../logs/
# custom log4j configuration file
# smp.log.configuration.file=smp-logback.xml

# *********************************
# Extension folder
# *********************************

# path where SMP extensions are located. The Folder is loaded by the SMP classloader
at startup.
smp.libraries.folder=/cef/test/smp/apache-tomcat-8.5.73/smp/ext-lib
```

SMP Configuration Properties

The `WEB-INF/classes/smp.config.properties` file is used to configure the initial SMP properties needed for the SMP startup.

▼ **SMP Configuration Properties**

Configuration Property	Description and Usage	Default
`smp.configuration.file`	*Configuration property file path.*	`smp.config.properties`
`smp.init.configuration.file`	*Init configuration property file path.*	`smp.init.properties`
`smp.security.folder`	*Security folder for storing the keystore and the truststore.*	`smp`

Configuration Property	Description and Usage	Default
`smp.jdbc.driver`	*Database Configuration - Driver* • MySQL: `com.mysql.jdbc.Driver` • Oracle: `oracle.jdbc.OracleDriver`	`com.mysql.jdbc.Driver`
`smp.jdbc.url`	*Database Configuration - URL* • MySQL: `jdbc:mysql://dbhost:dbport/smp_database` • Oracle: ◦ `jdbc:oracle:thin:@dbhost:dbport:smp_database` ◦ `jdbc:oracle:thin:@dbhost:dbport/smp_service`	`jdbc:mysql://localhost:3306/smp`
`smp.jdbc.user`	*Database User/Password Configuration - User*	`smp`
`smp.jdbc.password`	*Database User/password Configuration - Password*	`The_password`
`smp.datasource.jndi`	_If the data source is configured on the application server (*recommended), the property defines the JNDI name of the database connection._	`jdbc/eDeliverySmpDs`
`smp.database.show-sql`	*Print generated sql queries to logs. The property is effective only when* `smp.mode.development=true`.	`false`
`smp.database.create-ddl`	*Auto create/update database objects. The property is effective only when* `smp.mode.development=true`.	`false`

Configuration Property	Description and Usage	Default
`smp.log.folder`	**IMPORTANT** Do NOT this feature in production, it is only intended for tests, demonstrations and development purposes. *The provided logback.xml configuration defines logging file as* `<file>${log.folder:-logs}/edelivery-smp.log</file>` *With the property we can define the folder for the logging files.*	`/var/logs/smp`
`smp.log.configuration.file`	*Custom logback configuration file (filepath can be absolute or relative to smp configuration.dir).*	`/opt/logging/smp-logback.xml`
`smp.libraries.folder`	*Path where SMP extensions are located. The folder is loaded by the SMP classloader at startup.*	`/opt/smp/extension-libs`
`smp.smp.mode.development`	*The development mode uses semi-random generators for password and key generation. Setting the property value to ‘true‘ makes the first startup and access token generation faster. To ensure high security, this option MUST NOT be enabled in production.*	`false`

SMP Application Configuration

eDelivery SMP application configuration values are stored in the database table `SMP_CONFIGURATION`. If the table is empty (usually at first SMP startup), edelivery SMP populates the table at startup with all properties and default values.

When updating properties via the user interface, the property values are taken into account immediately if the server starts in non-cluster mode (property: `smp.cluster.enabled = false`).

Otherwise, each node refreshes the properties on all cluster nodes at the same time in accordance with the property refreshes defined in the CRON expression via the `smp.property.refresh.cronJobExpression` property.

▼ **SMP Application Configuration Properties**

Configuration Property	Description and Usage	Default
`smp.instance.name`	*The name of the DomiSMP instance is used in email notifications and alerts to specify which SMP instance generated the notifications.* **Usage**: Requires restart: No Value Type: STRING	`Test DomiSMP Instance`
`contextPath.output`	*This property controls pattern of URLs produced by SMP in GET ServiceGroup responses.* **Usage**: Requires restart: Yes Value Type: BOOLEAN	`true`
`encodedSlashesAllowedInUrl`	*Allow encoded slashes in context path. Set to true if slashes are part of identifiers.* **Usage**: Requires restart: Yes Value Type: BOOLEAN	`true`
`smp.http.forwarded.headers.enabled`	*Use (value true) or remove (value false) forwarded headers. There are security considerations for forwarded headers since an application cannot know if the headers were added by a proxy, as intended, or by a malicious client.* **Usage**: Requires restart: No Value Type: BOOLEAN	`false`
`smp.http.httpStrictTransportSecurity.maxAge`	*How long (in seconds) should HSTS last in the browser cache (default one year).* **Usage**: Requires restart: Yes Value Type: INTEGER	`31536000`

Configuration Property	Description and Usage	Default

`smp.http.header.security.policy`	*Content Security Policy (CSP)*
	`default-src 'self';`
	`script-src 'self';`
	`connect-src 'self';`
	`img-src 'self';`
	`style-src 'self'`
	`'unsafe-inline';`
	`frame-ancestors 'self';`
	`form-action 'self';`
	Usage:
	Requires restart: Yes
	Value Type: STRING
		-

Configuration Property	Description and Usage	Default
`smp.proxy.host`	*The http proxy host.* **Usage**: Requires restart: No Value Type: STRING	-
`smp.noproxy.hosts`	*List of nor proxy hosts.* **Usage**: Requires restart: No Value Type: STRING **Default**: `localhost|127.0.0.1`	See Description
`smp.proxy.password`	*Base64 encrypted password for Proxy.* **Usage**: Requires restart: No Value Type: STRING	-
`smp.proxy.port`	*The http proxy port.* **Usage**: Requires restart: No Value Type: INTEGER	80
`smp.proxy.user`	*The proxy user.* **Usage**: Requires restart: No Value Type: STRING	-

Configuration Property	Description and Usage	Default

<table>
<tr><th>Configuration Property</th><th>Description and Usage</th><th>Default</th></tr>
<tr><td>identifiersBehaviour.ParticipantIdentifierScheme.validationRegex</td><td><em>Participant Identifier Schema of each PUT ServiceGroup request is validated against this schema.</em>
<strong>Usage</strong>:
Requires restart: No
Value Type: REGEXP
<strong>Default</strong>: $|(?!^.{26})(-[a-z0-9]-)$|^urn:oasis:names:tc:ebcore:partyid-type:(iso6523|unregistered)(:.)?$</td><td>See Description</td></tr>
<tr><td>identifiersBehaviour.ParticipantIdentifierScheme.validationRegexMessage</td><td><em>Error message for UI.</em>
<strong>Usage</strong>:
Requires restart: No
Value Type: STRING
<strong>Value Format</strong>:
Participant scheme must start with urn:oasis:names:tc:ebcore:partyid-type:(iso6523:|unregistered:)
<em>OR</em>
• must be up to 25 characters long with form [domain]-[identifierArea]-[identifierType]
• and may only contain the following characters: [a-z0-9].
<strong>Example</strong>: busdox-actorid-upis</td><td>-</td></tr>
<tr><td>identifiersBehaviour.scheme.mandatory</td><td><em>Scheme for participant identifier is mandatory.</em>
<strong>Usage</strong>:
Requires restart: No
Value Type: BOOLEAN</td><td>true</td></tr>
</table>

<table>
<tr><th>Configuration Property</th><th>Description and Usage</th><th>Default</th></tr>
<tr><td>smp.ui.session.idle_timeout.admin</td><td><em>Specifies the time, in seconds, between client requests before the SMP will invalidate session for ADMIN users (System).</em>
<strong>Usage</strong>:
Requires restart: No
Value Type: INTEGER</td><td>300</td></tr>
</table>

Configuration Property	Description and Usage	Default
`smp.ui.session.idle_timeout.user`	*Specifies the time, in seconds, between client requests before the SMP will invalidate session for users (Service group, SMP Admin).* **Usage**: Requires restart: No Value Type: INTEGER	`1800`
`smp.cluster.enabled`	*Define if application is set in cluster. In not cluster environment, properties are updated on set Property.* **Usage**: Requires restart: No Value Type: BOOLEAN	`false`

<table>
<tr><th>Configuration Property</th><th>Description and Usage</th><th>Default</th></tr>
<tr><td>smp.credentials.reset_request.url</td><td><i>The URL address for resetting the password in DomiSMP. This URL can be used when DomiSMP is behind a reverse proxy. If the property value is not set, the reset URL is created using reverse proxy headers such as Host and X-Forwarded-Host.</i>
<b>Usage</b>:
Requires restart: No
Value Type: URL</td><td>-</td></tr>
<tr><td>smp.credentials.reset_request.url.validMinutes</td><td><i>The number of minutes for reset token to be valid.</i>
<b>Usage</b>:
Requires restart: No
Value Type: INTEGER</td><td>90</td></tr>
<tr><td>smp.passwordPolicy.validationRegex</td><td><i>Password minimum complexity rules.</i>
<b>Usage</b>:
Requires restart: No
Value Type: REGEXP
<b>Default</b>:
^(?=.*[0-9])(?=.*[a-z])(?=.*[A-Z])(?=.*[~`!@#$%^&+=\-_<>.,?:;*/()|\[\]\{}'"\\]).\{16,32}$</td><td>See
Description</td></tr>
</table>

Configuration Property	Description and Usage	Default
`smp.passwordPolicy.validationMessage`	*The error message shown to the user in case the password does not follow the regex put in the* `smp.passwordPolicy.pattern property`. **Usage**: Requires restart: No Value Type: STRING Must have: • Minimum length: 16 characters. • Maximum length: 32 characters. • At least one letter in lowercase; • At least one letter in uppercase; • At least one digit; • At least one special character.	-
`smp.passwordPolicy.validDays`	*Number of days password is valid.* **Usage**: Requires restart: No Value Type: INTEGER	`90`
`smp.passwordPolicy.warning.beforeExpiration`	*How many days before expiration should the UI warn users at login.* **Usage**: Requires restart: No Value Type: INTEGER	`15`
`smp.passwordPolicy.expired.forceChange`	*Force change password at UI login if expired.* **Usage**: Requires restart: No Value Type: BOOLEAN	`true`

Configuration Property	Description and Usage	Default
`smp.user.login.fail.delay`	*Delay response in ms on invalid username or password.* **Usage**: Requires restart: No Value Type: INTEGER	1000

Configuration Property	Description and Usage	Default
`smp.user.login.maximum.attempt`	*Number of console login attempt before the user is deactivated.* **Usage**: Requires restart: No Value Type: INTEGER	`5`
`smp.user.login.suspension.time`	*Time in seconds for a suspended user to be reactivated.* **Usage**: Requires restart: No Value Type: INTEGER - If set to `0`, the user will not be reactivated.	`3600`

Configuration Property	Description and Usage	Default
`smp.accessToken.validDays`	*Number of days access token is valid.* **Usage**: Requires restart: No Value Type: INTEGER	`60`
`smp.accessToken.login.maximum.attempt`	*Number of accessToken login attempt before the accessToken is deactivated.* **Usage**: Requires restart: No Value Type: INTEGER	`10`
`smp.accessToken.login.suspension.time`	*Time in seconds for a suspended accessToken to be reactivated.* **Usage**: Requires restart: No Value Type: INTEGER - If set to `0`, the user will not be reactivated.	`3600`
`smp.accessToken.login.fail.delay`	*Delay in ms on invalid token id or token.* **Usage**: Requires restart: No Value Type: INTEGER	`1000`
`smp.ui.authentication.types`	*Set list of '|' separated authentication types: PASSWORD|SSO.* **Usage**: Requires restart: No Value Type: LIST_STRING	`PASSWORD`

<table>
<tr><th>Configuration Property</th><th>Description and Usage</th><th>Default</th></tr>
<tr><td><code>smp.automation.authentication.types</code></td><td><em>Set list of "|" separated application-automation authentication types (Web-Service integration).</em>
<strong>Usage</strong>:
Requires restart: No
Value Type: LIST_STRING
Supported Values: TOKEN, CERTIFICATE.
<strong>Default</strong>:
<code>TOKEN|CERTIFICATE</code></td><td>See Description</td></tr>
<tr><td><code>smp.automation.authentication.external.tls.clientCert.enabled</code></td><td><em>Authentication with external module as: reverse proxy. Authenticated data are sent to application using 'Client-Cert' HTTP header. Do not enable this feature without a properly configured reverse-proxy.</em>
<strong>Usage</strong>:
Requires restart: No
Value Type: BOOLEAN</td><td><code>false</code></td></tr>
<tr><td><code>smp.automation.authentication.external.tls.SSLClientCert.enabled</code></td><td><em>Authentication with external module as: reverse proxy. Authenticated certificate is sent to application using <code>SSLClientCert</code> HTTP header. Do not enable this feature without properly a configured reverse-proxy.</em>
<strong>Usage</strong>:
Requires restart: No
Value Type: BOOLEAN</td><td><code>false</code></td></tr>
</table>

Configuration Property	Description and Usage	Default
`smp.sso.cas.ui.label`	*The SSO service provider label.* **Usage**: Requires restart: Yes Value Type: STRING	`EU Login`

Configuration Property	Description and Usage	Default
`smp.sso.cas.url`	*The SSO CAS URL endpoint.* **Usage**: Requires restart: Yes Value Type: URL **Default**: `http://localhost:8080/cas/`	See Description
`smp.sso.cas.urlPath.login`	*The CAS URL path for login.* *The complete URL is composed by parameters:* `${smp.sso.cas.url}/${smp.sso.cas.urlpath.login}.` **Usage**: Requires restart: Yes Value Type: STRING	`login`
`smp.sso.cas.callback.url`	*The URL is the callback URL belonging to the local SMP Security System. If using RP,make sure it target SMP path* `/ui/public/rest/security/cas`. **Usage**: Requires restart: Yes Value Type: URL **Default**: `http://localhost:8080/smp/ui/public/rest/security/cas`	See Description
`smp.sso.cas.smp.urlPath`	*SMP relative path which triggers CAS authentication.* **Usage**: Requires restart: Yes Value Type: STRING **Default**: `/smp/ui/public/rest/security/cas`	See Description

Configuration Property	Description and Usage	Default

`smp.sso.cas.smp.user.data.urlPaths`	*Relative path for CAS user data. Complete URL is composed by parameters:*
	`${smp.sso.cas.url}/${smp.sso.cas.smp.user.data.urlpath}`
	Usage:
	Requires restart: Yes
	Value Type: STRING
	Default:
	`userdata/myAccount.cgi`
		See Description

`smp.sso.cas.token.validation.urlPath`	The CAS URL path for login. Complete URL is composed of parameters: `${smp.sso.cas.url}/${smp.sso.cas.token.validation.urlpath}` **Usage**: Requires restart: Yes Value Type: STRING **Default**: `laxValidate`	See Description
`smp.sso.cas.token.validation.params`	*The CAS token validation key:value properties separated with a pipe (`|`).* **Usage**: Requires restart: Yes Value Type: MAP_STRING **Default**: `acceptStrengths:BASIC,CLIENT_CERT|assuranceLevel:TOP`	See Description

Configuration Property	Description and Usage	Default
`smp.sso.cas.token.validation.groups`	*Pipe-separated (`|`) CAS groups user must belong to.* **Usage**: Requires restart: Yes Value Type: LIST_STRING **Default**: `DIGIT_SMP|DIGIT_ADMIN`	See Description
`smp.sso.cas.registration.enabled`	*If the value is set to true, the user is automatically registered to DomiSMP the first time they use the external CAS. The CAS server provides the necessary user data, which is then mapped to the DomiSMP user entity according to the smp.sso.cas.registration.mapping.* **Usage**: Requires restart: No Value Type: BOOLEAN	`true`
`smp.sso.cas.registration.confirmation.mandatory`	*The value determines whether the CAS-automatically created user is activated immediately or if the System admin must activate the user before they can log in to DomiSMP.* **Usage**: Requires restart: No Value Type: BOOLEAN	`false`
`smp.sso.cas.registration.mapping`	*Pipe-separated (`|`) key:value list of mapping defining how CAS user data is mapped to DomiSMP user entity. Currently supported values are: `EMAIL` and `FULL_NAME`. The username of the newly created user is the CAS principal name/identifier* **Usage**: Requires restart: No Value Type: MAP_STRING **Default**: `EMAIL:${email}\|FULL_NAME:${firstName} ${lastName}`	See Description

Configuration Property	Description and Usage	Default
Configuration Property	**Description and Usage**	**Default**
`mail.smtp.host`	Email configuration: *Email server.* **Usage**: Requires restart: No Value Type: STRING	-
`mail.smtp.port`	Email configuration: *SMTP mail port.* **Usage**: Requires restart: No Value Type: INTEGER	`25`
`mail.smtp.protocol`	Email configuration: *SMTP mail protocol.* **Usage**: Requires restart: No Value Type: STRING	smtp
`mail.smtp.username`	Email configuration: *SMTP mail protocol; mail sender’s username.* **Usage**: Requires restart: No Value Type: STRING	-
`mail.smtp.password`	Email configuration: _SMTP mail protocol; mail sender’s encrypted password. **Usage**: Requires restart: No Value Type: STRING	-
`mail.smtp.properties`	*Pipe-separated (`|`) key:value properties list.* **Usage**: Requires restart: No Value Type: MAP_STRING **Example**: `mail.smtp.auth:true|mail.smtp.starttls.enable:true|mail.smtp.quitwait:false`	-
Configuration Property	**Description and Usage**	**Default**

Configuration Property	Description and Usage	Default
`smp.alert.user.created.enabled`	*Enable or disable notifications for user creation events.* **Usage**: Requires restart: No Value Type: BOOLEAN	`true`
`smp.alert.user.created.level`	*User creation event notification alert level.* **Usage**: Requires restart: No Value Type: STRING Possible values: LOW, MEDIUM, HIGH	`HIGH`
`smp.alert.user.updated.enabled`	*Enable or disable notifications when user data is changed.* **Usage**: Requires restart: No Value Type: BOOLEAN	`true`
`smp.alert.user.updated.level`	*User update data event notification alert level.* **Usage**: Requires restart: No Value Type: STRING Possible values: LOW, MEDIUM, HIGH	`HIGH`
`smp.alert.user.login_failure.enabled`	*Enable/disable the login failure alert of the authentication module.* **Usage**: Requires restart: No Value Type: BOOLEAN	`false`
`smp.alert.user.login_failure.level`	*Login failure alert level.* **Usage**: Requires restart: No Value Type: STRING Possible values: LOW, MEDIUM, HIGH	`LOW`
`smp.alert.user.suspended.enabled`	*Enable/disable the login suspended alert of the authentication module.* **Usage**: Requires restart: No Value Type: BOOLEAN	`true`

Configuration Property	Description and Usage	Default
`smp.alert.user.suspended.level`	*Suspended login alert level.* **Usage**: Requires restart: No Value Type: STRING Possible values: LOW, MEDIUM, HIGH	`HIGH`
`smp.alert.user.suspended.mail.moment`	*When should the account disabled alert be triggered.* **Usage**: Requires restart: No Value Type: STRING Possible Values: • `AT_LOGON`: if set, an alert is triggered each time a user tries to log into a disabled account. • `WHEN_BLOCKED`: if set, an alert is triggered when the account is suspended.	`WHEN_BLOCKED`

Configuration Property	Description and Usage	Default
`smp.alert.password.imminent_expiration.enabled`	*Enable/disable the "Password about to expire" alert.* **Usage**: Requires restart: No Value Type: BOOLEAN	`true`
`smp.alert.password.imminent_expiration.delay_days`	*Number of days before password expiration the system is to send alerts.* **Usage**: Requires restart: No Value Type: INTEGER	`15`
`smp.alert.password.imminent_expiration.frequency_days`	*Frequency in days for (re)sending the "Password about to expire" alert.* **Usage**: Requires restart: No Value Type: INTEGER	`5`
`smp.alert.password.imminent_expiration.level`	*"Password about to expire" alert's level.* **Usage**: Requires restart: No Value Type: STRING Possible values: LOW, MEDIUM, HIGH	`LOW`

Configuration Property	Description and Usage	Default
Configuration Property	**Description and Usage**	**Default**
`smp.alert.password.expired.enabled`	*Enable/disable the "Password expired" alert.* **Usage**: Requires restart: No Value Type: BOOLEAN	`true`
`smp.alert.password.expired.delay_days`	*Period in days after password expiration the system is to send "password expiration" alerts.* **Usage**: Requires restart: No Value Type: INTEGER	`30`
`smp.alert.password.expired.frequency_days`	*Frequency in days between "Password expired" alerts.* **Usage**: Requires restart: No Value Type: INTEGER	`5`
`smp.alert.password.expired.level`	*"Password expired" alert's level.* **Usage**: Requires restart: No Value Type: STRING Possible values: LOW, MEDIUM, HIGH.	`LOW`

Configuration Property	Description and Usage	Default
`smp.alert.accessToken.imminent_expiration.enabled`	*Enable/disable the "accessToken about to expire" alert.* **Usage**: Requires restart: No Value Type: BOOLEAN	`true`
`smp.alert.accessToken.imminent_expiration.delay_days`	*Number of days before password expiration the system is to send alerts.* **Usage**: Requires restart: No Value Type: INTEGER	15
`smp.alert.accessToken.imminent_expiration.frequency_days`	*Frequency in days between alerts.* **Usage**: Requires restart: No Value Type: INTEGER	`5`

Configuration Property	Description and Usage	Default
`smp.alert.accessToken.imminent_expiration.level`	*AccessToken imminent expiration alert level.* **Usage**: Requires restart: No Value Type: STRING Possible values: LOW, MEDIUM, HIGH.	`LOW`

Configuration Property	Description and Usage	Default
`smp.alert.accessToken.expired.enabled`	*Enable/disable the accessToken expiration alert.* **Usage**: Requires restart: No Value Type: BOOLEAN	`true`
`smp.alert.accessToken.expired.delay_days`	*Number of days after expiration as for how long the system should send alerts.* **Usage**: Requires restart: No Value Type: INTEGER	`30`
`smp.alert.accessToken.expired.frequency_days`	*Frequency in days between alerts.* **Usage**: Requires restart: No Value Type: INTEGER	`30`
`smp.alert.accessToken.expired.level`	*Access Token expiration alert level.* **Usage**: Requires restart: No Value Type: STRING Possible values: LOW, MEDIUM, HIGH.	`LOW`

Configuration Property	Description and Usage	Default
`smp.alert.certificate.imminent_expiration.enabled`	*Enable/disable the imminent certificate expiration alert.* **Usage**: Requires restart: No Value Type: BOOLEAN	`true`
`smp.alert.certificate.imminent_expiration.delay_days`	*Number of days before expiration as for how long before expiration the system should send alerts.* **Usage**: Requires restart: No Value Type: INTEGER	`15`

Configuration Property	Description and Usage	Default
`smp.alert.certificate.imminent_expiration.frequency_days`	*Frequency in days between alerts.* **Usage**: Requires restart: No Value Type: INTEGER	`5`
`smp.alert.certificate.imminent_expiration.level`	*Certificate imminent expiration alert level. Values: |{LOW, MEDIUM, HIGH}* **Usage**: Requires restart: No Value Type: STRING	`LOW`

Configuration Property	Description and Usage	Default
`smp.alert.certificate.expired.enabled`	*Enable/disable the certificate expiration alert.* **Usage**: Requires restart: No Value Type: BOOLEAN	`true`
`smp.alert.certificate.expired.delay_days`	*Number of days after expiration as for how long the system should send alerts.* **Usage**: Requires restart: No Value Type: INTEGER	`30`
`smp.alert.certificate.expired.frequency_days`	*Frequency in days between alerts.* **Usage**: Requires restart: No Value Type: INTEGER	`5`
`smp.alert.certificate.expired.level`	*Certificate expiration alert level.* **Usage**: Requires restart: No Value Type: STRING Possible values: LOW, MEDIUM, HIGH	`LOW`

Configuration Property	Description and Usage	Default
`smp.alert.credentials.cronJobExpression`	*Property CRON expression for triggering alert messages.* **Usage**: Requires restart: No Value Type: CRON_EXPRESSION **Default**: `0 52 4 */1 * *`	See Description

Configuration Property	Description and Usage	Default
`smp.alert.credentials.serverInstance`	*Which instance (hostname) to generates a report when `smp.cluster.enabled` is set to true.* **Usage**: Requires restart: No Value Type: STRING	`localhost`
`smp.alert.credentials.batch.size`	Max alerts generated in a batch for the type. **Usage**: Requires restart: No Value Type: INTEGER	`200`

Configuration Property	Description and Usage	Default
`smp.alert.mail.from`	*Alert send mail.* **Usage**: Requires restart: No Value Type: EMAIL **Default**: `test@alert-send-mail.eu`	See Description
`smp.domain.default`	*Default domain code.* **Usage**: Requires restart: No Value Type: STRING • If the domain cannot be determined from the request, the default domain is used.	-

<table>
<tr><th>Configuration Property</th><th>Description and Usage</th><th>Default</th></tr>
<tr><td><code>smp.certificate.validation.allowed.certificate.types</code></td><td><em>Allowed user certificate types.</em>
<b>Usage</b>:
Requires restart: No
Value Type: LIST_STRING
<b>Example</b>:
<code>RSA|EC|Ed25519|Ed448</code>
• If empty no restrictions are imposed.
• For other values see the java KeyFactory Algorithms.</td><td>-</td></tr>
</table>

Configuration Property	Description and Usage	Default
authentication.blueCoat.enabled	NOTE Property was replaced by property: smp.automation.authentication.external.tls.clientCert.enabled **Usage**: Requires restart: No Value Type: BOOLEAN	false
smp.domain.default	*Default domain code. If the domain cannot be determined from the request, the default domain is used.* **Usage**: Requires restart: No Value Type: STRING	-
smp.certificate.validation.allowed.certificate.types	*Allowed user certificate types.* **Usage**: Requires restart: No Value Type: LIST_STRING **Example**: `RSA|EC|Ed25519|Ed448` - If empty no restrictions are imposed. - For other values see the java KeyFactory Algorithms.	-

2.5.3. Configuration Table

The smp_domain table is used to support the multitenancy feature of the SMP. Its parameters/fields are:

SML_SMP_ID

This is the SMP ID that must match the SMP ID registered within the SML.

SML_CLIENT_CERT_HEADER

The SMP’s certificate - needed only when accessing BDMSL directly through HTTP. The configured "Client-Cert" HTTP header will be added to each BDMSL request (bypassing SSL certificate verification made normally by SSL terminator).

SML_CLIENT_KEY_ALIAS

This is the Domain scoped alias of the keystore private key used for authentication with the SML.

The password is the same as `xmldsig.keystore.password` defined in the SMP configuration file.

SIGNATURE_KEY_ALIAS

This field points to the **Domain scoped** alias of the Keystore private key certificate, used by the SMP to sign GET Signed Service Metadata responses.

SML_SUBDOMAIN

This is the informative identifier of SML domain code (eHealth, Peppol, etc). Since SML subdomain is part of DNS domain it must be a valid DNS domain part.

DOMAIN_CODE

The unique domain code that is used as HTTP domain parameter when adding participants true REST service API to particular a domain. Domain code can be alphanumeric and up to 63 characters long.

Example

Update the default single domain smp_domain table record:

```
UPDATE smp_domain SET SML_SMP_ID='SMP-MCB-ID14',
SML_CLIENT_KEY_ALIAS= 'smp_mock';
```

or

```
UPDATE smp_domain SET SML_SMP_ID='SMP-MCB-ID14',
SML_CLIENT_CERT_HEADER=
'serial=000000000000000000009A195D2DD88C&subject=CN=SMP_1000000000,O=DG-
DIGIT,C=BE&validFrom=Oct 21 02:00:00 2014 CEST&validTo=Oct 21 01:59:59 2016
CEST&issuer=CN=Issuer
Common Name,OU=Issuer Organization Unit,O=Issuer Organization,C=BE'
WHERE domainId='default';
```

2.5.4. Database configuration

The eDelivery SMP database back-end configuration is performed within the eDelivery SMP configuration file (**smp.config.properties** file).

Depending on the selected database back-end, modify the **smp.config.properties** files as indicated below. SMP database's connection can be configured in the properties file, or it can use application server datasource configuration by JNDI.

Oracle Database

- Datasource configured from property file:

```
../

## Sample for Oracle
```

```
jdbc.driver=oracle.jdbc.driver.OracleDriver
jdbc.url=jdbc:oracle:thin:@[.mark]#localhost:1521/xe
jdbc.user=smp
jdbc.password=secret123
hibernate.dialect=org.hibernate.dialect.Oracle10gDialect

/..

* Datasource (connection pool) configured on the application server
using the JNDI (recommended):

../

hibernate.dialect=org.hibernate.dialect.Oracle10gDialect

# weblogic datasource JNDI example
# datasource.jndi=jdbc/edeliverySmpDS

# tomcat datasource JNDI example
datasource.jndi=java:comp/env/jdbc/edeliverySmpDS
```

MySQL

- Datasource configured from property file:

```
../
# For mysql connector v8
jdbc.driver=com.mysql.cj.jdbc.Driver

# For mysql connector v5
jdbc.driver=com.mysql.jdbc.Driver
jdbc.url=jdbc:mysql://localhost:3306/smp
jdbc.user=smp
jdbc.password=secret123

hibernate.dialect=org.hibernate.dialect.MySQL5InnoDBDialect

/..
Datasource (connection pool) configured on the application server
using the JNDI (recommended)
../

hibernate.dialect=**org.hibernate.dialect.Oracle10gDialect

# weblogic datasource JNDI example
# datasource.jndi=**jdbc/edeliverySmpDS
# tomcat datasource JNDI example
datasource.jndi=**java:comp/env/jdbc/edeliverySmpDS
```

```
/..
```

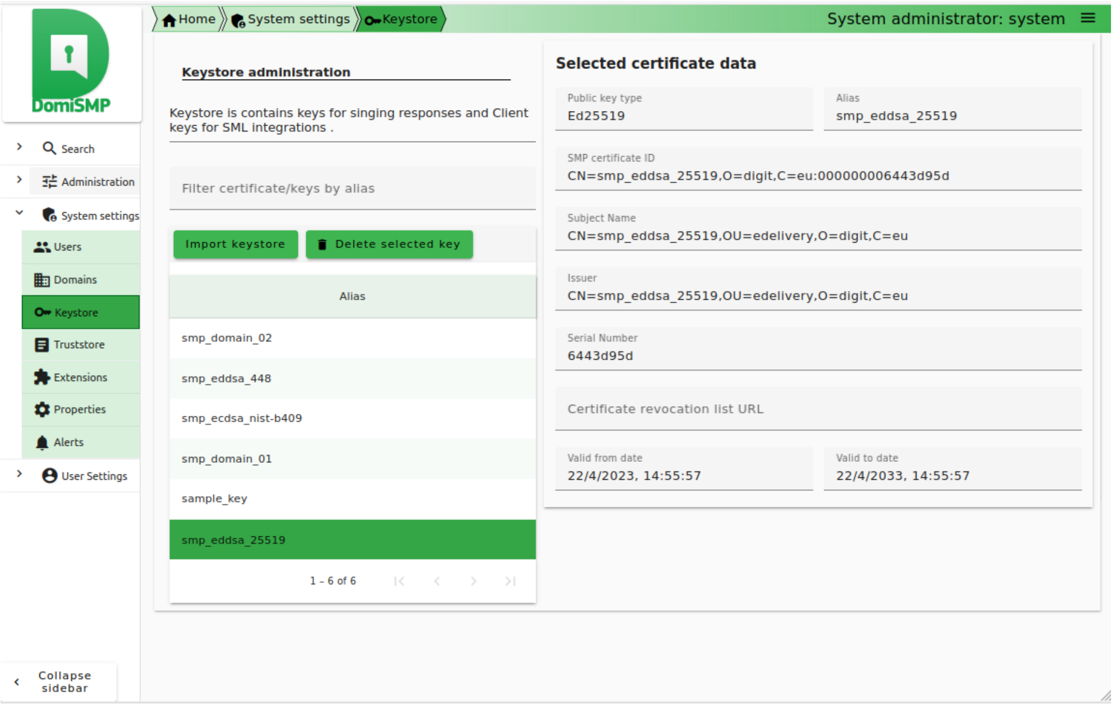
2.5.5. SMP Keystore

eDelivery SMP uses keystore for storing keys for the two different purposes:

- One **mandatory** key is used for signing the responses to **GET** requests (XMLDSIG response signing).
- One **optional** key is used to authenticate SMP using 2-way-SSL when it is calling SML via HTTPS.

If the Keystore does not exist when the SMP is started for the first time, it is automatically created with a sample key/certificate 'sample_key'.

The user with a system administrator role can update/manage the Keystore entries using the user interface on the
System settings/Keystore page:



Keystore administration

Keystore is contains keys for singing responses and Client keys for SML integrations .

Filter certificate/keys by alias

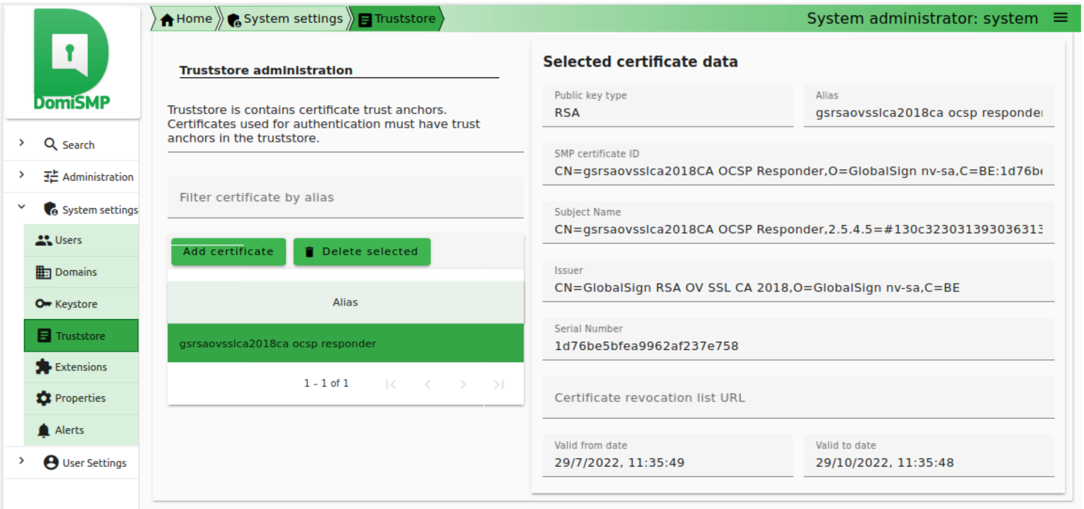
smp_domain_02
smp_eddsa_448
smp_ecdsa_nist-b409
smp_domain_01
sample_key
smp_eddsa_25519

Selected certificate data

Public key type: Ed25519
Alias: smp_eddsa_25519
SMP certificate ID: CN=smp_eddsa_25519,O=digit,C=eu:000000006443d95d
Subject Name: CN=smp_eddsa_25519,OU=edelivery,O=digit,C=eu
Issuer: CN=smp_eddsa_25519,OU=edelivery,O=digit,C=eu
Serial Number: 6443d95d
Certificate revocation list URL
Valid from date: 22/4/2023, 14:55:57
Valid to date: 22/4/2033, 14:55:57

2.5.6. SMP Truststore

eDelivery SMP uses truststore for storing trusted X509Certificates for the WebService 2-way-SSL authentication and for storing the SML server certificate. The truststore is automatically created at the initial SMP start-up. The truststore can be managed with a System admin account using the UI tools under the page System settings / Truststore.



Truststore administration

Truststore is contains certificate trust anchors. Certificates used for authentication must have trust anchors in the truststore.

Filter certificate by alias

Alias
gsrsaovsslca2018ca ocsp responder

Selected certificate data

Public key type: RSA

Alias: gsrsaovsslca2018ca ocsp responde

SMP certificate ID: CN=gsrsaovsslca2018CA OCSP Responder,O=GlobalSign nv-sa,C=BE:1d76b

Subject Name: CN=gsrsaovsslca2018CA OCSP Responder,2.5.4.5=#130c32303139303631

Issuer: CN=GlobalSign RSA OV SSL CA 2018,O=GlobalSign nv-sa,C=BE

Serial Number: 1d76be5bfea9962af237e758

Certificate revocation list URL

Valid from date: 29/7/2022, 11:35:49

Valid to date: 29/10/2022, 11:35:48

2.5.7. Custom Keystore and Truststore

On some systems, generating new passwords and keys can take a long time. To speed up the initial startup, consider the following option:

- Install faster system random generators.
- In case of development or local testing, set property: **smp.mode.development**=**true** to **smp.config.properties.** To ensure high security, this option MUST NOT be enabled in production.
- Use custom/preprepared keystore and truststore as described below.

Users can configure eDelivery SMP to use prepared keystores at initial startup. To achieve this, the Keystore must be generated manually and saved in the SMP security folder. If the Keystore already contains the keys/certificates, they must have the same Key password as it was set for accessing the Keystore.

The following properties must be set in `smp.config.properties`:

- `smp.security.folder`: The security folder of the SMP where the keystore must be located.
- `smp.keystore.filename`: The keystore's filename.
- `smp. truststore.password`: Password for accessing the keystore and keys.

NOTE Decrypted passwords must be wrapped in {DEC}\{[PASSWORD]};. Example: {DEC}{testPASSkeystore1234}.

```
../

smp.security.folder=/opt/tomcat/security/
smp.keystore.filename=smp-keystore.p12
smp.keystore.password=\{DEC}\{testPASSkeystore1234}
smp.keystore.type=PKCS12

smp.truststore.filename=smp-keystore.p12
smp.truststore.password=\{DEC}\{testPASStruststore1234}
smp.truststore.type=PKCS12

/..
```

- After initial startup, the properties are stored (and the password encrypted) inside the SMP_CONFIGURATION table, and they should be removed from the smp.config.properties file.

2.6. STEP4 Deploying SMP Application

SMP .war file Deployment

The eDelivery SMP is deployed using the steps described in the next sections.

2.6.1. Tomcat

Download and copy smp-X.war file in the Tomcat /webapps directory (AS_HOME/webapps/smp.war).

NOTE The application context path is the same as the first part of the smp.war filename. For example, if we deploy the file smp.war, then the application will be accessible on http://localhost:8080/smp/.
If the deployed file is smp-X.war, the application URL will be:
http://localhost:8080/smp-X/.

2.6.2. WebLogic/Oracle

Deploy the .war file within WebLogic using the Oracle Weblogic deployer feature or using the Weblogic Administration Console.

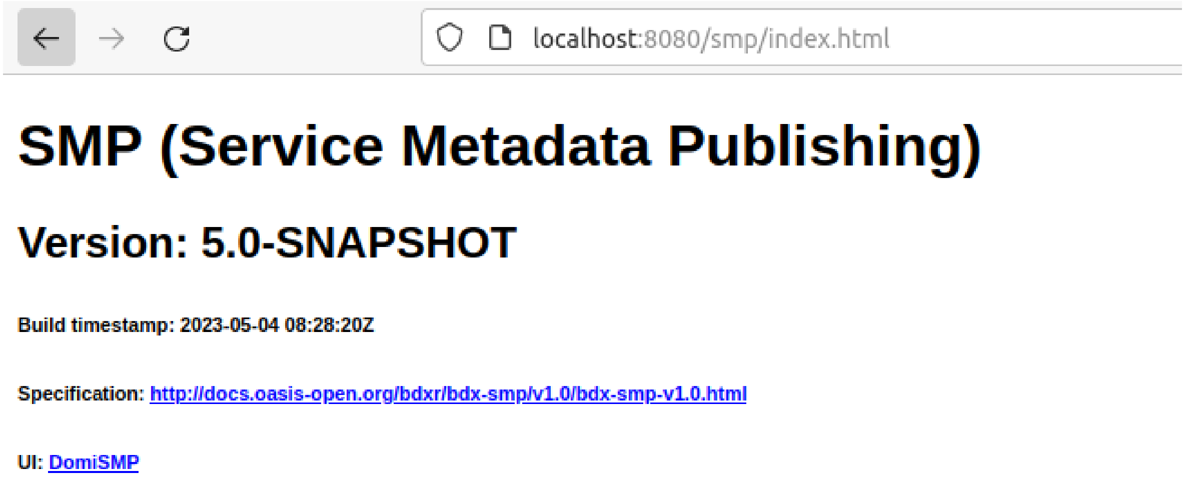
An example of using the Oracle ,the weblogic.deployer, is shown below:

```
java weblogic.Deployer -adminurl
t3://$\{WebLogicAdminServerListenAddress}:$\{WebLogicAdminServerPort} \
-username $\{WebLogicAdminUserName} \
-password $\{WebLogicAdminUserPassword} \
-deploy -name smp.war \
-targets $\{SMP_ManagedServer} \
-source $TEMP_DIR/* smp.war
```

Installation Verification

Verify the installation by navigating in your browser:
http://<hostname>:<port>/smp.

If the deployment was successful, the following page is displayed:



SMP (Service Metadata Publishing)

Version: 5.0-SNAPSHOT

Build timestamp: 2023-05-04 08:28:20Z

Specification: http://docs.oasis-open.org/bdxr/bdx-smp/v1.0/bdx-smp-v1.0.html

UI: DomiSMP

2.7. Configuring SMP/BDMSL integration

Configuring the eDelivery SMP for use with an BDMSL

The eDelivery SMP can establish an BDMSL integration using two identification mechanisms:

- Using HTTP and plain text with metadata embedded into the HTTP header Client-Cert of the REST request Using HTTP and plain text with metadata embedded into the HTTP header Client-Cert of the REST request.

IMPORTANT	This approach should **be used only for testing purposes** and only if both BDMSL and eDelivery SMP are located in the same network where the BDMSL web services are **not** exposed to the internet.

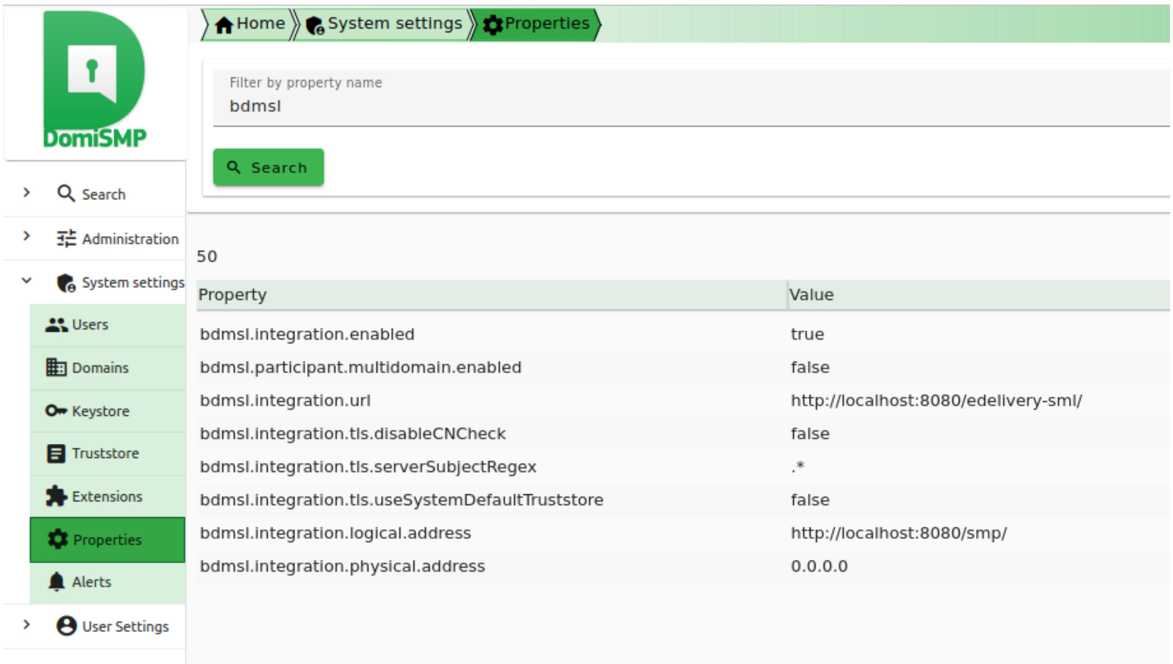
- Using 2-way HTTPS/TLS (**recommended**).

The BDMSL integration configuration has two parts:

- Configuration of the BDMSL integration data as: BDMLS URL, SMPs URL, etc.
- Configuration of the SMP domain credentials/X509Certificate and unique SMP identifier.

2.7.1. Configuring BDSML

The BDMSL integration data can be set using the UI Property tool:



Filter by property name
bdmsl

50

Property	Value
bdmsl.integration.enabled	true
bdmsl.participant.multidomain.enabled	false
bdmsl.integration.url	http://localhost:8080/edelivery-sml/
bdmsl.integration.tls.disableCNCheck	false
bdmsl.integration.tls.serverSubjectRegex	.*
bdmsl.integration.tls.useSystemDefaultTruststore	false
bdmsl.integration.logical.address	http://localhost:8080/smp/
bdmsl.integration.physical.address	0.0.0.0

▼ **BDMSL Configuration Properties**

To configure BDMSL, set the following properties:

- `bdmsl.integration.enabled`: set value to true to enable BDMSL (SML) integration.
- `bdmsl.integration.url`: set the URL where BDMSL is located.
  Example: https://acc.edelivery.tech.ec.europa.eu/edelivery-sml/
- `bdmsl.integration.logical.address`: set the public SMP URL address. The URL is used by the BDMSL when generating DNS records for the SMP. Do not change this property once the SMP domain is registered to BDMSL. Example: `https://smp.domain.eu/smp`.
- `bdmsl.integration.physical.address`: IP4 address of the SMP server. The value is informative and can be `0.0.0.0`.

▼ **2-Way TLS Authentication Configuration Properties**

If using the 2-Way TLS authentication, configure:

- `bdmsl.integration.tls.disableCNCheck`: if set to `true`, the BDMSL server domain and Certificate CN value must match with the BDMSL certificate to be trusted.
- `bdmsl.integration.tls.useSystemDefaultTruststore`: if set to `true`, the system default truststore is used to verify the BDMSL truststore. The system default truststore usually points to the `$JAVA_HOME/lib/security/cacerts` truststore, or is configured on the application server using the `javax.net.ssl.trustStore` system parameter. If the property is set to false, the SMP truststore is used to verify the BDMSL server certificate trust.
- `bdmsl.integration.tls.serverSubjectRegex`: regular expression for BDMSL server TLS certificate subject verification.
  Example: CertEx. .`CN=acc.edelivery.tech.ec.europa.eu.`.

2.7.2. Configuring SMP domain credentials

Once BDMSL integration data is configured, the next step is to configure the SMP client certificate and ID for the BDMSL authentication. Because SMP 4.2 can handle multiple domains, each domain can have its X509Certificate to log into the correct BDMSL DNS domain.

To configure the SMP domain credentials:

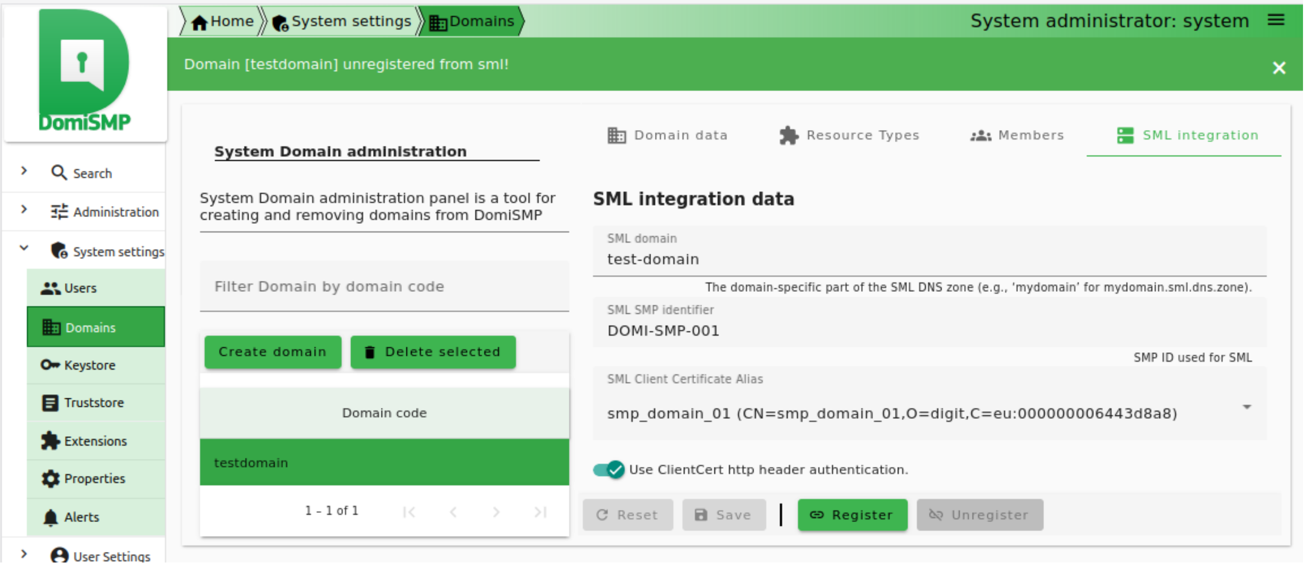
1. Register BDMSL client key/certificate to the SMP Keystore.
2. Create or edit Domain in the UI/Domain tool, enter the SMP ID and choose the client certificate.
3. Choose the authentication type. SML supports two ways of authentication.

   ClientCert

   HTTP Client-Cert certificate header. This must be used only behind a reverse proxy. The BDMSL should NOT allow this type of authentication from the internet. In practice, the HTTP Client-Cert should be generated only by the reverse proxy.

   HTTPS/TLS*

   standard mutual TLS authentication (**recommended*).



Domain [testdomain] unregistered from sml!

System Domain administration

System Domain administration panel is a tool for creating and removing domains from DomiSMP

Filter Domain by domain code

Domain code
testdomain

SML integration data

SML domain
test-domain

The domain-specific part of the SML DNS zone (e.g., 'mydomain' for mydomain.sml.dns.zone).

SML SMP identifier
DOMI-SMP-001

SMP ID used for SML

SML Client Certificate Alias
smp_domain_01 (CN=smp_domain_01,O=digit,C=eu:000000006443d8a8)

☑ Use ClientCert http header authentication.

2.8. SMP User Management

The DomiSMP has two user application roles:

- System Admin: the role allows user to modify DomiSMP system management and settings such as: Domain management, User management, Truststore management, Key management, DomiSMP configuration, etc.
- User: this role allows the user to log into the DomiSMP.

The user can have additional permissions on editing DomiSMP entities when they are assigned to the Resource, Groups, and Domains. Please read the following chapter for more details.

2.8.1. Domain, Group and Resources

The DomiSMP supports 3-layer security realms.

Resource

the most basic unit.

The Resource is identified by the unique ID, which is part of the URL of the resource as example:

`http://localhost/smp/resource-identifier`

A Resource example is the *Service Group* document from the Oasis SMP specification. The user can be a Resource member with **Admin** or **Viewer** membership roles. If the user has an Admin membership role, it can modify resource document(s) and manage the resource memberships. If the user has role Viewer, it can view/read the Resource if the Resource has visibility set to: `Private`.

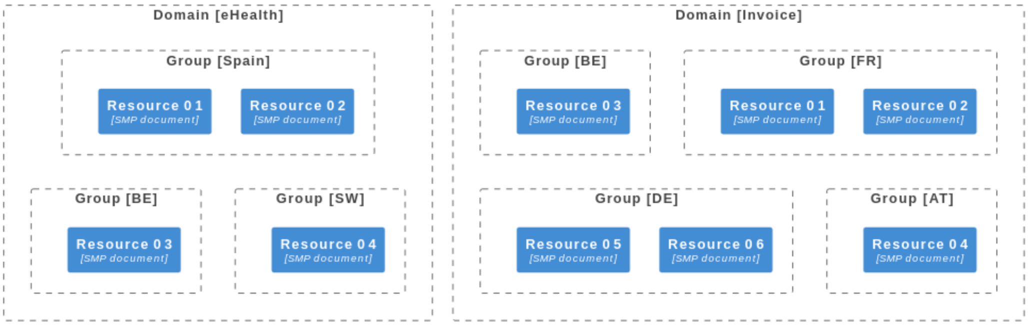
Group

a cluster of resources managed by the dedicated group administrators.
The group admin(s) can create and delete the resource, but *only* the resource admins can modify data/documents for the resource. The user can be a Group member with **Admin** or **Viewer** membership roles. With Admin group membership, the user can create and delete group resources. If the user has group role Viewer, it can view/read the Resources if the Group has visibility set to: “Private”.

Domain

the top layer.
It indicates the business purpose of the network of participants, such as invoice exchange, Health Records message exchanges, etc. The Domain usually has a domain owner who handles participant interoperability, defining message types, network authentication, and authorization methods such as Certificate PKI, Identity Service providers, etc.

In DomiSMP 5.0, the user with a Domain Admin role can create domain groups and assign users to them.


Domain [eHealth]
Group [Spain]
Resource 01 [SMP document]
Resource 02 [SMP document]
Group [BE]
Resource 03 [SMP document]
Group [SW]
Resource 04 [SMP document]
Domain [Invoice]
Group [BE]
Resource 03 [SMP document]
Group [FR]
Resource 01 [SMP document]
Resource 02 [SMP document]
Group [DE]
Resource 05 [SMP document]
Resource 06 [SMP document]
Group [AT]
Resource 04 [SMP document]

The provided database script creates the following users:

User name	Role	Default password [1]
system	SYSTEM_ADMIN	123456
user	USER	123456

2.8.2. User Roles

The following DomiSMP users can be of three types, as briefly described below:

Actor	UC	Short description	Oper.	Data
Group Admin	Create or Update resource: 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` the resource identifier. Information is stored into Resource table. This same service is used to create and update a ServiceGroup.	PUT	ServiceGroup
Group SMP	Erase Service Group	Erases the resource (service group definition) AND the list of sub-resources such as servicemetada for the specified receiver participant.	DELETE	ServiceGroup
Resource Admin	Create or Update Resource such as: Service group document and subresources: 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
Resource Admin	Erase Service Metadata	Remove all information about one specific service (i.e. all related processes and endpoints definitions).	DELETE	ServiceMetadata
Anonymous User	Retrieve Service Group	Obtain the list of public 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`.	GET	ServiceGroup
Anonymous User	Retrieve Service Metadata	Obtain detailed definition about one specific service of a specific participant for all supported transports. This service retrieves the SignedServiceMetadata according to the input quadruplet `participantIdentifier` `participantIndentifierScheme` `documentIdentifier` `documentIdentifierScheme`.	GET	SignedServiceMetadata

Actor	UC	Short description	Oper.	Data
System admin		Create, modify, and delete users and domains. System admin can be only used in the DomiSMP UI.		

NOTE For a complete description of the SMP user management, please consult the SMP Interface Control Document (ICD) document available at https://ec.europa.eu/digital-building-blocks/wikis/display/DIGITAL/SMP.

+ Users can be added, modified and deleted using the SMP Admin console or directly by executing sql commands. Below are instructions on how to modify users in the database.

2.8.3. BCRYPT password generation

To manage DomiSMP users, you can use the DomiSMP UI console.

Use the procedure below to create the first system admin user.
An alternative is to use the provided SQL init scripts and replace passwords by first login.

The DomiSMP uses the BCRYPT algorithm to hash users' passwords. A BCRYPT-hashing tool is bundled with the SMP WAR file.

To get the hashing code:

1. Place a copy of the `smp-X.war` file into a temporary directory.
2. Extract the war file using the `jar` command:

```
jar -xvf smp-X.war
```

Obtain one or multiple hashes at once, using the following command:

```
java -cp "WEB-INF/lib/" eu.europa.ec.edelivery.smp.utils.BCryptPasswordHash*
<password_to_be_hashed>
```

The result is a BCRYPT hash of the specified password (in this example, `123456`)

```
java -cp "WEB-INF/lib/" eu.europa.ec.edelivery.smp.utils.BCryptPasswordHash 123456
```

Result:

```
2a$10$6nYTSUSh2BQfbOLIyCXn8eUViBcnn.WcjUrWOtJlMNDOdAtI85zMa
```

The next command shows the hashing of several passwords at once, separated by a space in the command.

```
java -cp "WEB-INF/lib/" eu.europa.ec.edelivery.smp.BCryptPasswordHash*
<password_to_be_hashed_1> <password_to_be_hashed_2>
2a$10$6nYTSUSh2BQfbOLIyCXn8eUViBcnn.WcjUrWOtJlMNDOdAtI85zMa
2a$107zNzSeZpxiHeqY2BRKkHE.HknfIe3aiu6XzU.qHHnnPbUHKtfcmDG
```

2.8.4. SMP Database User Creation

Adding an SMP user is done by adding a new entry in the SMP database **SMP_USER** table either directly or via the Administration console.

The User role is set in the SMP_USER table APPLICATION_ROLE column as follows:

User Role	Role value
System Administrator	SYSTEM_ADMIN
Domi SMP user	USER
AnonymousUser (Not defined in the SMP User database)	N/A

In the following examples, a **System Admin** user is created.

SYSTEM_ADMIN SMP User creation

NOTE

To log in the Administration Console (for the first time), it is necessary to, create a user with SYSTEM_ADMIN privileges by entering the details directly into the SMP_USER table.
This initial user's password is generated using the BCRYPT utility described previously.
If PASSWORD_CHANGED is not set, the user will be asked to change the password at first logon.
Example of a SYSTEM_ADMIN user creation:
Username:
smp_admin
Password (Hashed):
$2a$10$6nYTSUSh2BQfbOLIyCXn8eUViBcnn.WcjUrWOtJlMNDOdAtI85zMa
Role: SYSTEM_ADMIN

Execute the following database command using the database user/password created in the Database Configuration section of this guide.

MySql example:

```
insert into SMP_USER (USERNAME, ACTIVE, APPLICATION_ROLE, EMAIL,
CREATED_ON, LAST_UPDATED_ON) values

1,'smp_admin', 1, 'SYSTEM_ADMIN',
'system@mail-example.local', NOW(), NOW());

insert into SMP_CREDENTIAL (FK_USER_ID, CREDENTIAL_ACTIVE,
CREDENTIAL_NAME, CREDENTIAL_VALUE, CREDENTIAL_TYPE, CREDENTIAL_TARGET,
CREATED_ON, LAST_UPDATED_ON) values

((select id from SMP_USER where USERNAME='smp_admin'),1,
'smp_admin',
'$2a$10$olcGeWKGEoRia2DPuFqRNeca0IEdRSmOrljLz57BAjf1jlC9SohrS',
'USERNAME_PASSWORD','UI', NOW(), NOW());
```

Oracle example:

```
insert into SMP_USER (ID, USERNAME, ACTIVE, APPLICATION_ROLE, EMAIL,
CREATED_ON, LAST_UPDATED_ON) values

(SMP_USER_SEQ.NEXTVAL,'smp_admin', 1, 'SYSTEM_ADMIN', 'system@mail-example.local',
sysdate, sysdate);

insert into SMP_CREDENTIAL (FK_USER_ID, CREDENTIAL_ACTIVE,
CREDENTIAL_NAME, CREDENTIAL_VALUE, CREDENTIAL_TYPE, CREDENTIAL_TARGET,
CREATED_ON, LAST_UPDATED_ON) values

((select id from SMP_USER where USERNAME='smp_admin'),1,'smp_admin',
'$2a$10$olcGeWKGEoRia2DPuFqRNeca0IEdRSmOrljLz57BAjf1jlC9SohrS',
'USERNAME_PASSWORD','UI', sysdate, sysdate);
```

NOTE The username/password credential is stored in the `SMP_CREDENTIAL` table.

The record must have the following values set to:

- `CREDENTIAL_VALUE`: the BCrypted password
- `CREDENTIAL_TYPE`: value must be set to: 'USERNAME_PASSWORD'
- `CREDENTIAL_TARGET`: value must be set to: 'UI'
- `FK_USER_ID`: value must be set to user id.

2.9. Logging Configuration

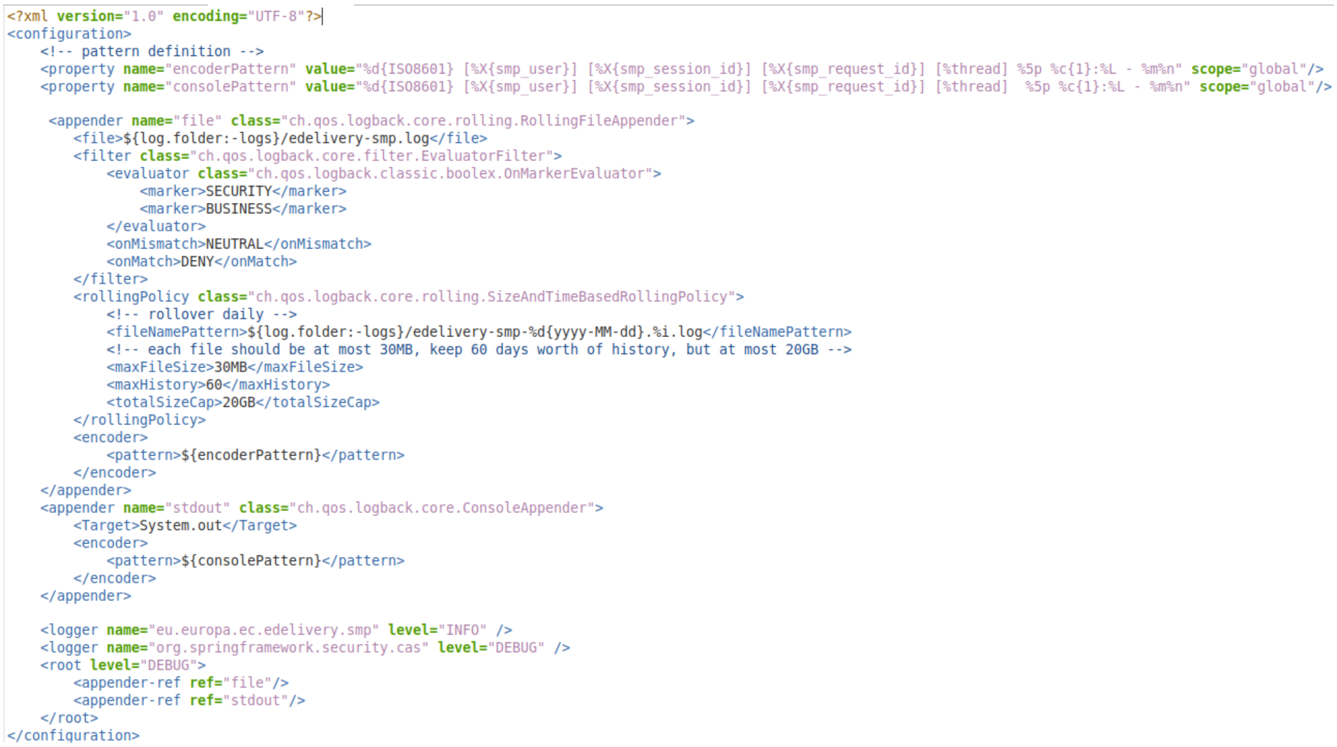
2.9.1. Logging properties

The SMP logging properties are defined in the ./WEB-INF/classes/logback.xml file embedded in the SMP `.war` file.

It is possible to modify the configuration of the logs by editing the embedded `logback.xml` or by defining new logback file in **smp.config.properties** file as example:

`log.configuration.file=/opt/apache-tomcat-8.5.30/smp/logback.xml`

In the example below, a `logback.xml` file is shown:



```
<?xml version="1.0" encoding="UTF-8"?>
<configuration>
    <!-- pattern definition -->
    <property name="encoderPattern" value="%d{ISO8601} [%X{smp_user}] [%X{smp_session_id}] [%X{smp_request_id}] [%thread] %5p %c{1}:%L - %m%n" scope="global"/>
    <property name="consolePattern" value="%d{ISO8601} [%X{smp_user}] [%X{smp_session_id}] [%X{smp_request_id}] [%thread]  %5p %c{1}:%L - %m%n" scope="global"/>

     <appender name="file" class="ch.qos.logback.core.rolling.RollingFileAppender">
        <file>${log.folder:-logs}/edelivery-smp.log</file>
        <filter class="ch.qos.logback.core.filter.EvaluatorFilter">
            <evaluator class="ch.qos.logback.classic.boolex.OnMarkerEvaluator">
                <marker>SECURITY</marker>
                <marker>BUSINESS</marker>
            </evaluator>
            <onMismatch>NEUTRAL</onMismatch>
            <onMatch>DENY</onMatch>
        </filter>
        <rollingPolicy class="ch.qos.logback.core.rolling.SizeAndTimeBasedRollingPolicy">
            <!-- rollover daily -->
            <fileNamePattern>${log.folder:-logs}/edelivery-smp-%d{yyyy-MM-dd}.%i.log</fileNamePattern>
            <!-- each file should be at most 30MB, keep 60 days worth of history, but at most 20GB -->
            <maxFileSize>30MB</maxFileSize>
            <maxHistory>60</maxHistory>
            <totalSizeCap>20GB</totalSizeCap>
        </rollingPolicy>
        <encoder>
            <pattern>${encoderPattern}</pattern>
        </encoder>
    </appender>
    <appender name="stdout" class="ch.qos.logback.core.ConsoleAppender">
        <Target>System.out</Target>
        <encoder>
            <pattern>${consolePattern}</pattern>
        </encoder>
    </appender>

    <logger name="eu.europa.ec.edelivery.smp" level="INFO" />
    <logger name="org.springframework.security.cas" level="DEBUG" />
    <root level="DEBUG">
        <appender-ref ref="file"/>
        <appender-ref ref="stdout"/>
    </root>
</configuration>
```

More details on how to configure logback can be found at:

https://logback.qos.ch/documentation.html

2.10. Capability Documents

SMP's primary function is to store and provide access to participant capability documents. Once stored in DomiSMP, these documents can be accessed via the SMP REST API during the dynamic discovery process. Examples of capability documents are:

- Oasis SMP 1.0 Service Groups Document
- Oasis SMP 1.0 ServiceMetadata document
- Oasis CPPA3-CPP document

The Oasis SMP 1.0 ServiceMetadata document example is shown below:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<ServiceMetadata xmlns="http://docs.oasis-open.org/bdxr/ns/SMP/2016/05"
xmlns:ns2="http://www.w3.org/2000/09/xmldsig#">
    <ServiceInformation>
        <ParticipantIdentifier scheme="iso6523-scheme-type">
0088:123456</ParticipantIdentifier>
        <DocumentIdentifier
            scheme="my-service-document">document-to-exchang</DocumentIdentifier>
        <ProcessList>
            <Process>
                <ProcessIdentifier scheme="as4-conformance-scheme">conformance-
service</ProcessIdentifier>
```

```
                <ServiceEndpointList>
                    <Endpoint transportProfile="bdxr-transport-ebms3-as4-v1p0">
                        <EndpointURI>http://access-point.eu/msh</EndpointURI>
                        <Certificate>Q2VydGlmaWNhdGUgZGF0YSA=</Certificate>
                        <ServiceDescription>Service description for partners
</ServiceDescription>
                        <TechnicalContactUrl>www.contact-data-
page.eu</TechnicalContactUrl>
                    </Endpoint>
                </ServiceEndpointList>
            </Process>
        </ProcessList>
    </ServiceInformation>
</ServiceMetadata>
```

DomiSMP categorizes these capability documents into two main levels:

Resources

Resources are the main documents of the participant. An example of a resource document is the Oasis SMP 1.0 Service Groups Document.

Sub-resources

The Sub-resources are the documents that provide additional information about a particular participant’s service. An example of a Sub-resource document is the Oasis SMP 1.0 Service Metadata document. .

2.10.1. Referencing Document Properties

In DomiSMP capability documents have a set of default properties which users can extend by creating custom properties.

DomiSMP also provides the possibility of referencing the values of these user-defined properties by using the following notation:

`${custom_prop_name}`

This allows users to reference these property values dynamically use them in the content of the document’s XML. When properties' values are modified, their references are updated transparently. DomiSMP replaces all these references before returning the resource via an API call.

Below is an example where `ParticipantIdentifier`, `DocumentIdentifier`, `EndpointURI`, and the `Certificate` endpoint are referenced.

Example Referencing Document Capabilities Properties

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ServiceMetadata xmlns="http://docs.oasis-open.org/bdxr/ns/SMP/2016/05"
xmlns:ns2="http://www.w3.org/2000/09/xmldsig#">
    <ServiceInformation>
        <ParticipantIdentifier scheme="${resource.identifier.scheme}"
```

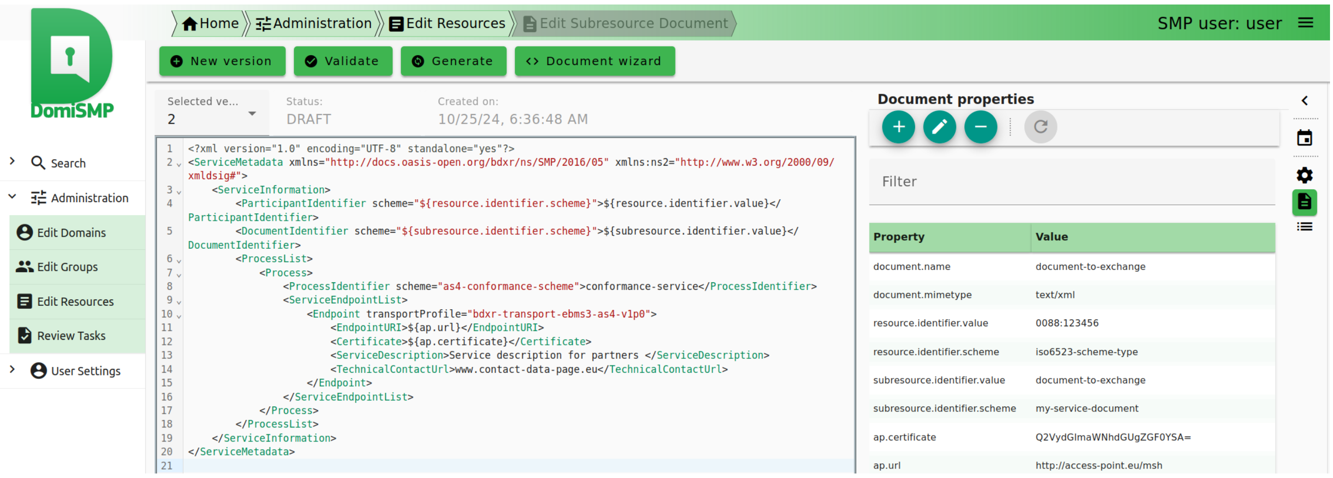
```
>${resource.identifier.value}</ParticipantIdentifier> ①
        <DocumentIdentifier scheme="${subresource.identifier.scheme}"
>${subresource.identifier.value}</DocumentIdentifier> ②
        <ProcessList>
            <Process>
                <ProcessIdentifier scheme="as4-conformance-scheme">conformance-
service</ProcessIdentifier>
                <ServiceEndpointList>
                    <Endpoint transportProfile="bdxr-transport-ebms3-as4-v1p0">
                        <EndpointURI>${ap.url}</EndpointURI>
                        <Certificate>${ap.certificate}</Certificate>
                        <ServiceDescription>Service description for partners
</ServiceDescription>
                        <TechnicalContactUrl>www.contact-data-
page.eu</TechnicalContactUrl>
                    </Endpoint>
                </ServiceEndpointList>
            </Process>
        </ProcessList>
    </ServiceInformation>
</ServiceMetadata>
```

Where:

① Is resolved as: `<ParticipantIdentifier scheme="iso6523-scheme-type">0088:123456</ParticipantIdentifier>`.

② Is resolved as: `<DocumentIdentifier scheme="my-service-document">document-to-exchang</DocumentIdentifier>`.

Document properties are managed in the console's document editor under the **Document editor/Document properties** page.



Selected ve... 2

Status: DRAFT

Created on: 10/25/24, 6:36:48 AM

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ServiceMetadata xmlns="http://docs.oasis-open.org/bdxr/ns/SMP/2016/05" xmlns:ns2="http://www.w3.org/2000/09/
xmldsig#">
    <ServiceInformation>
        <ParticipantIdentifier scheme="${resource.identifier.scheme}">${resource.identifier.value}</
ParticipantIdentifier>
        <DocumentIdentifier scheme="${subresource.identifier.scheme}">${subresource.identifier.value}</
DocumentIdentifier>
        <ProcessList>
            <Process>
                <ProcessIdentifier scheme="as4-conformance-scheme">conformance-service</ProcessIdentifier>
                <ServiceEndpointList>
                    <Endpoint transportProfile="bdxr-transport-ebms3-as4-v1p0">
                        <EndpointURI>${ap.url}</EndpointURI>
                        <Certificate>${ap.certificate}</Certificate>
                        <ServiceDescription>Service description for partners </ServiceDescription>
                        <TechnicalContactUrl>www.contact-data-page.eu</TechnicalContactUrl>
                    </Endpoint>
                </ServiceEndpointList>
            </Process>
        </ProcessList>
    </ServiceInformation>
</ServiceMetadata>
```

Document properties

Property	Value
document.name	document-to-exchange
document.mimetype	text/xml
resource.identifier.value	0088:123456
resource.identifier.scheme	iso6523-scheme-type
subresource.identifier.value	document-to-exchange
subresource.identifier.scheme	my-service-document
ap.certificate	Q2VydGlmaWNhdGUgZGF0YSA=
ap.url	http://access-point.eu/msh

Document's default properties are:

- Resources and sub-resources properties:
    - `document.name` - The name of the document.

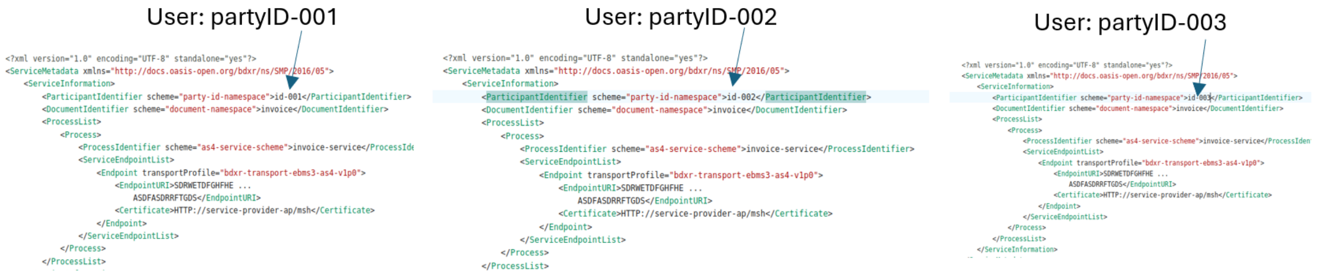
- `document.mimetype` - The version of the document.
- `resource.identifier.scheme` - the scheme of the resource identifier.
- `resource.identifier.value` - the value of the resource identifier.

- Sub-resource documents properties:
  - `subresource.identifier.scheme` - the scheme of the subresource identifier.
  - `subresource.identifier.value` - the value of the subresource identifier.

Other document properties can be defined and managed by the user within the **Document properties** tool.

2.10.2. Referencing Documents

The document referencing feature enables users to reuse the content of a single document across multiple resources. Prior to DomiSMP 5.1, each resource had its own document where users published their message exchange capabilities.



```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ServiceMetadata xmlns="http://docs.oasis-open.org/bdxr/ns/SMP/2016/05">
    <ServiceInformation>
        <ParticipantIdentifier scheme="party-id-namespace">id-001</ParticipantIdentifier>
        <DocumentIdentifier scheme="document-namespace">invoice</DocumentIdentifier>
        <ProcessList>
            <Process>
                <ProcessIdentifier scheme="as4-service-scheme">invoice-service</ProcessIde
                <ServiceEndpointList>
                    <Endpoint transportProfile="bdxr-transport-ebms3-as4-v1p0">
                        <EndpointURI>SDRWETDFGHFHE ...
                            ASDFASDRRFTGDS</EndpointURI>
                        <Certificate>HTTP://service-provider-ap/msh</Certificate>
                    </Endpoint>
                </ServiceEndpointList>
            </Process>
        </ProcessList>
```

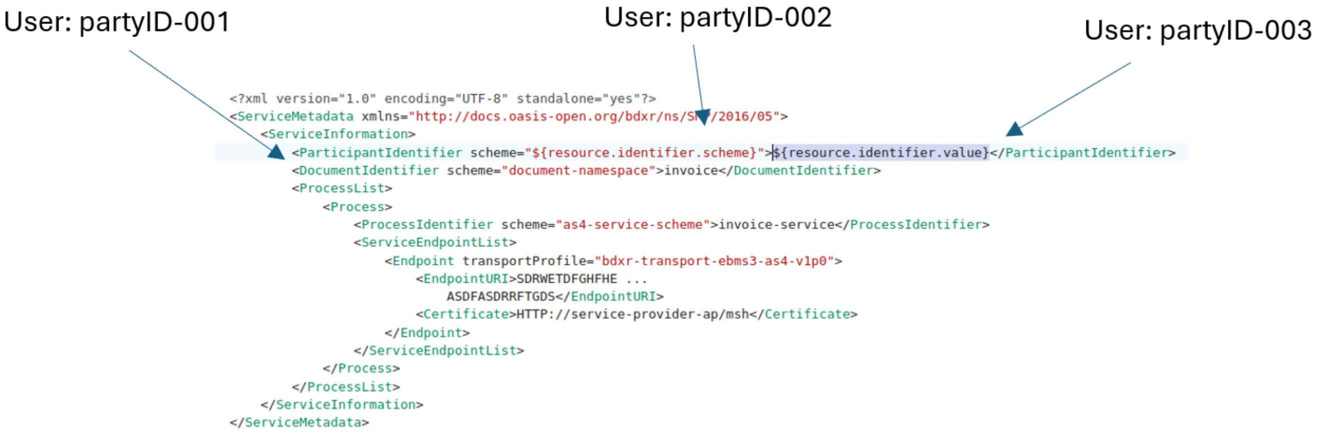
```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ServiceMetadata xmlns="http://docs.oasis-open.org/bdxr/ns/SMP/2016/05">
    <ServiceInformation>
        <ParticipantIdentifier scheme="party-id-namespace">id-002</ParticipantIdentifier>
        <DocumentIdentifier scheme="document-namespace">invoice</DocumentIdentifier>
        <ProcessList>
            <Process>
                <ProcessIdentifier scheme="as4-service-scheme">invoice-service</ProcessIdentifier>
                <ServiceEndpointList>
                    <Endpoint transportProfile="bdxr-transport-ebms3-as4-v1p0">
                        <EndpointURI>SDRWETDFGHFHE ...
                            ASDFASDRRFTGDS</EndpointURI>
                        <Certificate>HTTP://service-provider-ap/msh</Certificate>
                    </Endpoint>
                </ServiceEndpointList>
            </Process>
        </ProcessList>
```

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ServiceMetadata xmlns="http://docs.oasis-open.org/bdxr/ns/SMP/2016/05">
    <ServiceInformation>
        <ParticipantIdentifier scheme="party-id-namespace">id-003</ParticipantIdentifier>
        <DocumentIdentifier scheme="document-namespace">invoice</DocumentIdentifier>
        <ProcessList>
            <Process>
                <ProcessIdentifier scheme="as4-service-scheme">invoice-service</ProcessIden
                <ServiceEndpointList>
                    <Endpoint transportProfile="bdxr-transport-ebms3-as4-v1p0">
                        <EndpointURI>SDRWETDFGHFHE ...
                            ASDFASDRRFTGDS</EndpointURI>
                        <Certificate>HTTP://service-provider-ap/msh</Certificate>
                    </Endpoint>
                </ServiceEndpointList>
            </Process>
        </ProcessList>
    </ServiceInformation>
```

When multiple participants shared the same Access Point service provider, they all published within their service capabilities the same Access Point data: such as Access Point URL and Certificates. This resulted in multiple documents with the same content. To avoid this redundancy, DomiSMP 5.1 introduced the document referencing feature so that users can reference the same document across multiple resources.

This feature simplifies the maintenance of user's documents by reusing the same document template data multiple times.

Document referencing combined with placeholders allows users to override specific values with their own (e. g. participant identifier).

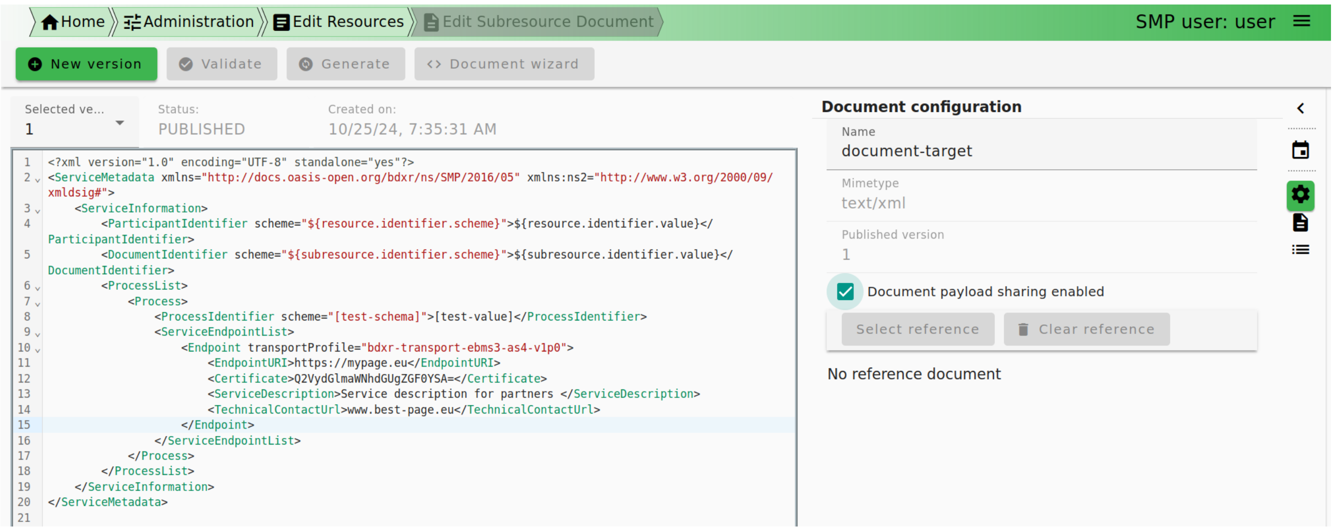


```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ServiceMetadata xmlns="http://docs.oasis-open.org/bdxr/ns/SMP/2016/05">
    <ServiceInformation>
        <ParticipantIdentifier scheme="${resource.identifier.scheme}">${resource.identifier.value}</ParticipantIdentifier>
        <DocumentIdentifier scheme="document-namespace">invoice</DocumentIdentifier>
        <ProcessList>
            <Process>
                <ProcessIdentifier scheme="as4-service-scheme">invoice-service</ProcessIdentifier>
                <ServiceEndpointList>
                    <Endpoint transportProfile="bdxr-transport-ebms3-as4-v1p0">
                        <EndpointURI>SDRWETDFGHFHE ...
                            ASDFASDRRFTGDS</EndpointURI>
                        <Certificate>HTTP://service-provider-ap/msh</Certificate>
                    </Endpoint>
                </ServiceEndpointList>
            </Process>
        </ProcessList>
    </ServiceInformation>
</ServiceMetadata>
```

When placeholder values are not defined in the final document, the values from the referenced document are used.

Making Document Available for Referencing

To allow documents to be referenced, they must have the **Document payload sharing enabled** option set in the **Document configuration** tool. Once the checkbox is selected, the documents can be referenced from other documents.



Selected ve... 1
Status: PUBLISHED
Created on: 10/25/24, 7:35:31 AM

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ServiceMetadata xmlns="http://docs.oasis-open.org/bdxr/ns/SMP/2016/05" xmlns:ns2="http://www.w3.org/2000/09/xmldsig#">
    <ServiceInformation>
        <ParticipantIdentifier scheme="${resource.identifier.scheme}">${resource.identifier.value}</ParticipantIdentifier>
        <DocumentIdentifier scheme="${subresource.identifier.scheme}">${subresource.identifier.value}</DocumentIdentifier>
        <ProcessList>
            <Process>
                <ProcessIdentifier scheme="[test-schema]">[test-value]</ProcessIdentifier>
                <ServiceEndpointList>
                    <Endpoint transportProfile="bdxr-transport-ebms3-as4-v1p0">
                        <EndpointURI>https://mypage.eu</EndpointURI>
                        <Certificate>Q2VydGlmaWNhdGUgZGF0YSA=</Certificate>
                        <ServiceDescription>Service description for partners </ServiceDescription>
                        <TechnicalContactUrl>www.best-page.eu</TechnicalContactUrl>
                    </Endpoint>
                </ServiceEndpointList>
            </Process>
        </ProcessList>
    </ServiceInformation>
</ServiceMetadata>
```

Document configuration

Name: document-target
Mimetype: text/xml
Published version: 1

☑ Document payload sharing enabled

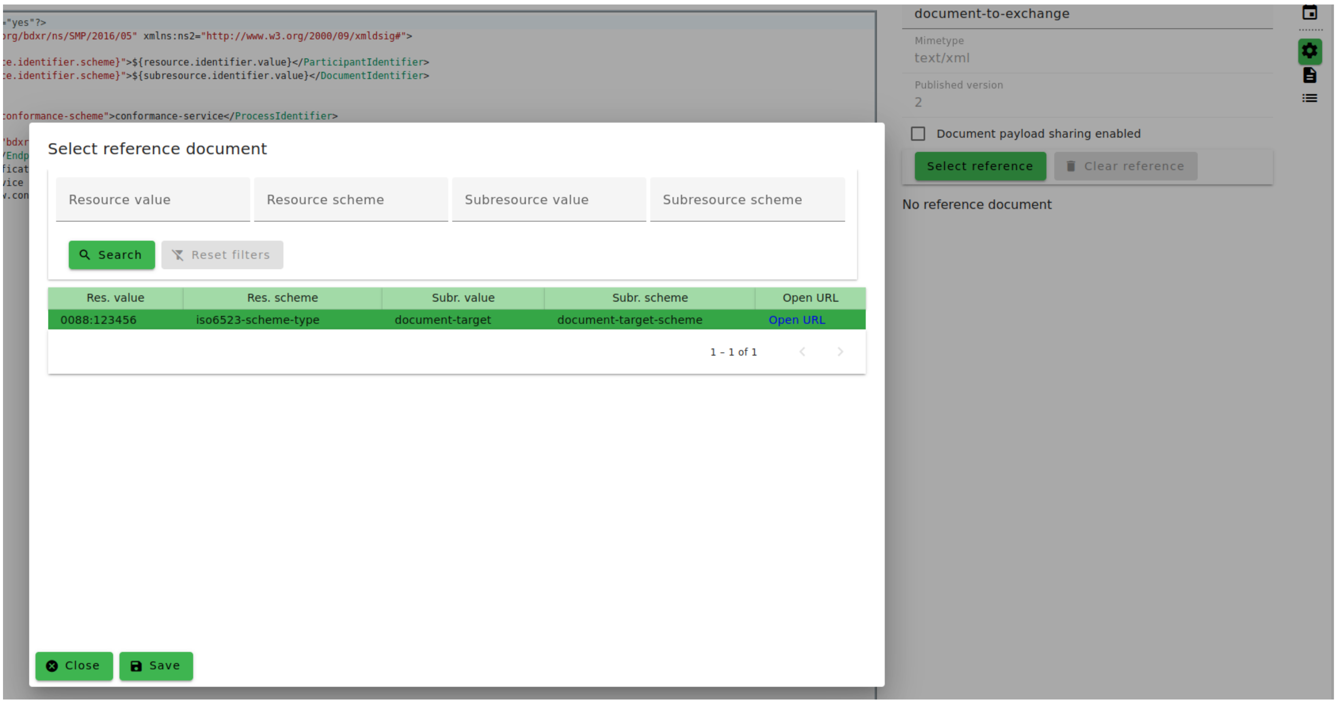
No reference document

How to Reference a Document

To reference a document, follow these steps in the **Document Configuration** tool:

1. Open the document where the reference will be used and choose the **Document Configuration** tab on the left side of the document editor.
2. Click the **Document References** button to open the **Document Reference** dialog.
3. From the Document Reference Table, select the document to be referenced. If the target document is not listed, use the filter fields to narrow the results.
4. Select the document and click the **Save** button to add the reference.
5. Save the changes to the document.

When retrieving the document via REST API, the referenced document is automatically included in the response.



```
="yes"?>
org/bdxr/ns/SMP/2016/05" xmlns:ns2="http://www.w3.org/2000/09/xmldsig#">

ce.identifier.scheme}">${resource.identifier.value}</ParticipantIdentifier>
ce.identifier.scheme}">${subresource.identifier.value}</DocumentIdentifier>


conformance-scheme">conformance-service</ProcessIdentifier>
```

Select reference document

Res. value	Res. scheme	Subr. value	Subr. scheme	Open URL
0088:123456	iso6523-scheme-type	document-target	document-target-scheme	Open URL

1 - 1 of 1

document-to-exchange

Mimetype
text/xml

Published version
2

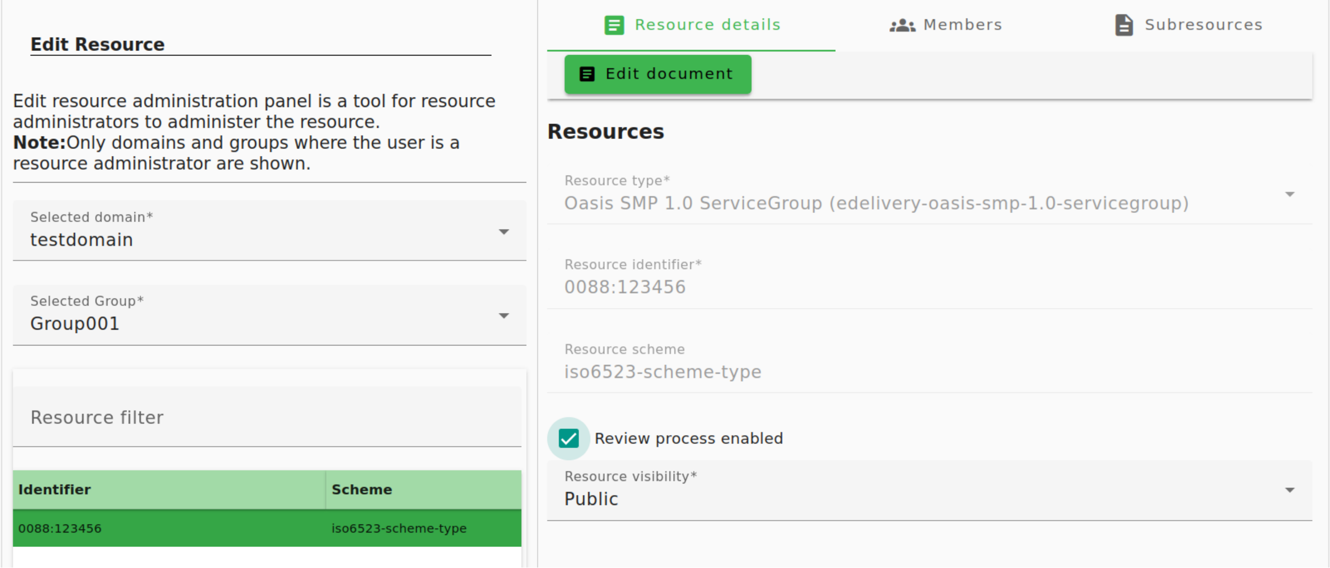
☐ Document payload sharing enabled

No reference document

IMPORTANT	If the document is already referencing another document, it cannot be set as "Shared enabled" as "Shared enabled" documents cannot reference other documents. == Document Review

DomiSMP offers a basic document review and approval feature, allowing users to review and approve documents before they are published. Disabled by default, this feature can be enabled for each Resource and Subresource via the DomiSMP Console, in the **Resource Details** page.

When the review process is enabled for a Resource, this option is applied to all related Sub-resources. The same is true for when disabling the review process.



Edit Resource

Edit resource administration panel is a tool for resource administrators to administer the resource.
Note: Only domains and groups where the user is a resource administrator are shown.

Selected domain*
testdomain

Selected Group*
Group001

Identifier	Scheme
0088:123456	iso6523-scheme-type

Resources

Resource type*
Oasis SMP 1.0 ServiceGroup (edelivery-oasis-smp-1.0-servicegroup)

Resource identifier*
0088:123456

Resource scheme
iso6523-scheme-type

☑ Review process enabled

Resource visibility*
Public

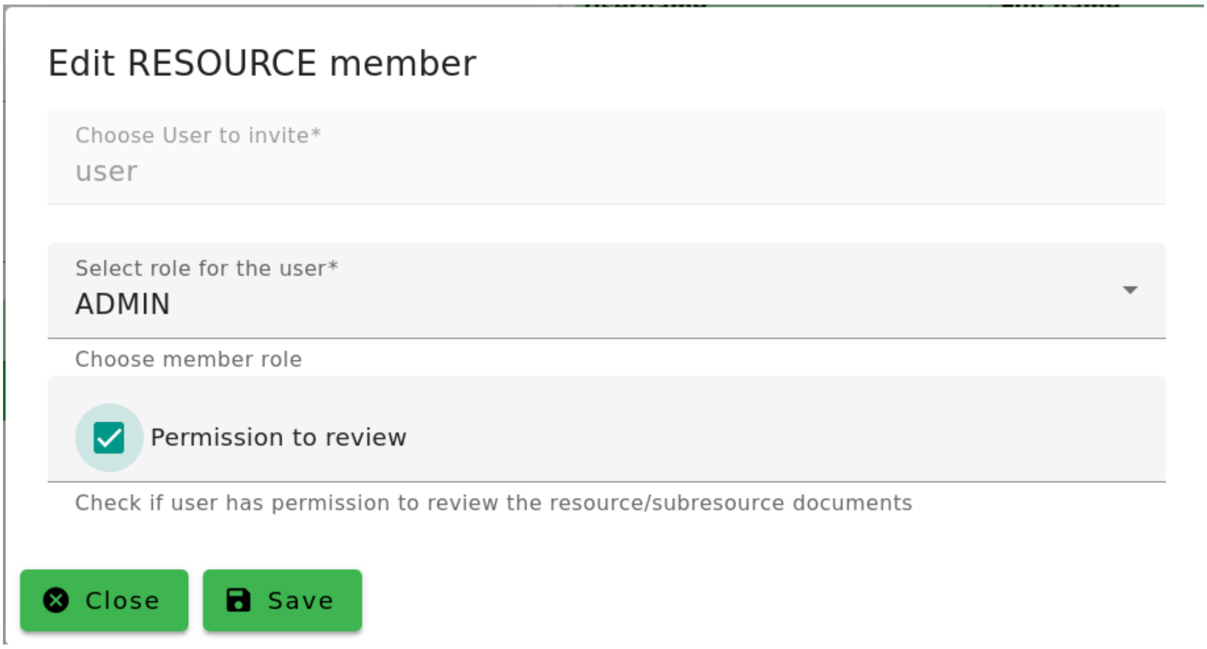
When the Review process is enabled, the following actions become available to the user in the DomiSMP Document editor page:

- `Submit for review` - allows the user to submit the document for review.
- `Approve` - allows the user to approve the document. Visible only for the users with review

permissions.

- `Reject` - allows the user to reject the document. Resource administrators can always reject documents under review even if when they do not have review permissions for the specific document in review. This ensures that the administrator can remove documents from the review queue.

The documents can be reviewed/approved only by the resource administrators that have the review permission enabled. The permissions can be set in the DomiSMP Console in the **Edit Resource** page. Once the document is approved, it can be published.



Edit RESOURCE member

Choose User to invite*
user

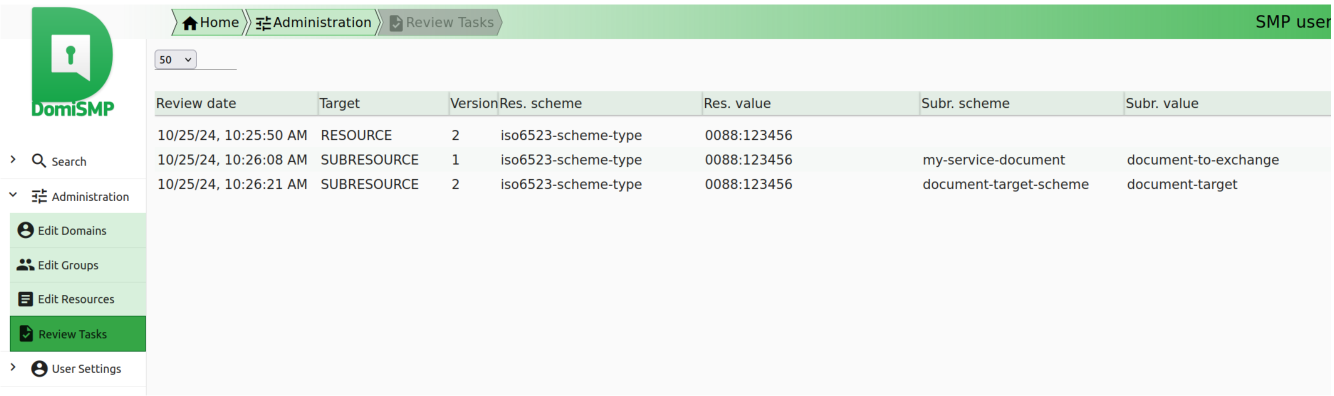
Select role for the user*
ADMIN

Choose member role

☑ Permission to review

Check if user has permission to review the resource/subresource documents

The reviewers can find documents under review in the DomiSMP Console in the **Review page**.



Review date	Target	Version	Res. scheme	Res. value	Subr. scheme	Subr. value
10/25/24, 10:25:50 AM	RESOURCE	2	iso6523-scheme-type	0088:123456		
10/25/24, 10:26:08 AM	SUBRESOURCE	1	iso6523-scheme-type	0088:123456	my-service-document	document-to-exchange
10/25/24, 10:26:21 AM	SUBRESOURCE	2	iso6523-scheme-type	0088:123456	document-target-scheme	document-target

by double-clicking on the document, the reviewer can see the document details, and approve or reject it. == Localization

Currently, DomiSMP provides language support and date/time format per locale.

2.10.3. Translations

DomiSMP UI Language Support

By default, DomiSMP UI ships in English, but it provides support for multiple languages. This feature allows users to add custom translations.

2.10.4. Setting Custom Translations

DomiSMP loads user provided translations from a pre-configured folder in its installation location. The location of this folder is set via the `smp.locale.folder=locales` configuration property.

This property takes as value either a relative or an absolute path to a directory accessible from the machine where DomiSMP is running on.

Relative Paths to Translation Files

If relative, the provided path describes the file's location relative to the working directory of the server into which DomiSMP is deployed. If the directory does not yet exist, DomiSMP creates it along with all its parents when at startup.

Multiple Translation Files

Multiple custom translations files can be deployed at the configured location.
Translation files are `.json` files with names that are:

- prefixed with `ui_`;
- followed by the ISO 639 language code (`en`, `fr`, etc.).

DomiSMP automatically loads new valid translation files placed at the configured location.

Creating a New Translation File

See below a part of the default `ui_en.json` file's:

`ui_en.json`

~~~
```json
{
  "column.selection.link.all": "All",
  "column.selection.link.none": "None",
  "column.selection.link.show": "Show columns",
  "column.selection.link.hide": "Hide columns",

  "cancel.dialog.text": "Do you want to cancel all unsaved operations?",
  "cancel.dialog.title": "Unsaved data",

  ...
}
```
~~~

When creating a new translation of the UI, copy the default `ui_en.json` file in order to reuse the labels referencing UI items, such as `column.selection.link.show` and replace the language values with its equivalents in the new language.

This new file should be named using the correct language code. See below a partial example for a French translation file.
See also **DomiSMP Supported Locales**.

ui_fr.json

~~~
```json
{
"column.selection.link.all": "Tout",
"column.selection.link.none": "Aucun",
"column.selection.link.show": "Afficher les colonnes",
"column.selection.link.hide": "Masquer les colonnes",

"cancel.dialog.text": "Voulez-vous annuler toutes les opérations non enregistrées ?",
"cancel.dialog.title": "Données non enregistrées",

  ...
}
```
~~~

▼ DomiSMP Supported Locales

File	Locale
ui_bg.json	Bulgarian
ui_cs.json	`Czech
ui_da.json	Danish
ui_de.json	German
ui_el.json	Greek
ui_en.json	English
ui_es.json	Spanish
ui_et.json	Estonian
ui_fi.json	Finnish
ui_fr.json	French
ui_hr.json	Croatian
ui_hu.json	Hungarian
ui_it.json	Italian
ui_lt.json	Lithuanian
ui_lv.json	Latvian
ui_mt.json	Maltese
ui_nl.json	Dutch
ui_pl.json	Polish
ui_pt.json	Portuguese
ui_ro.json	Romanian
ui_sk.json	Slovak

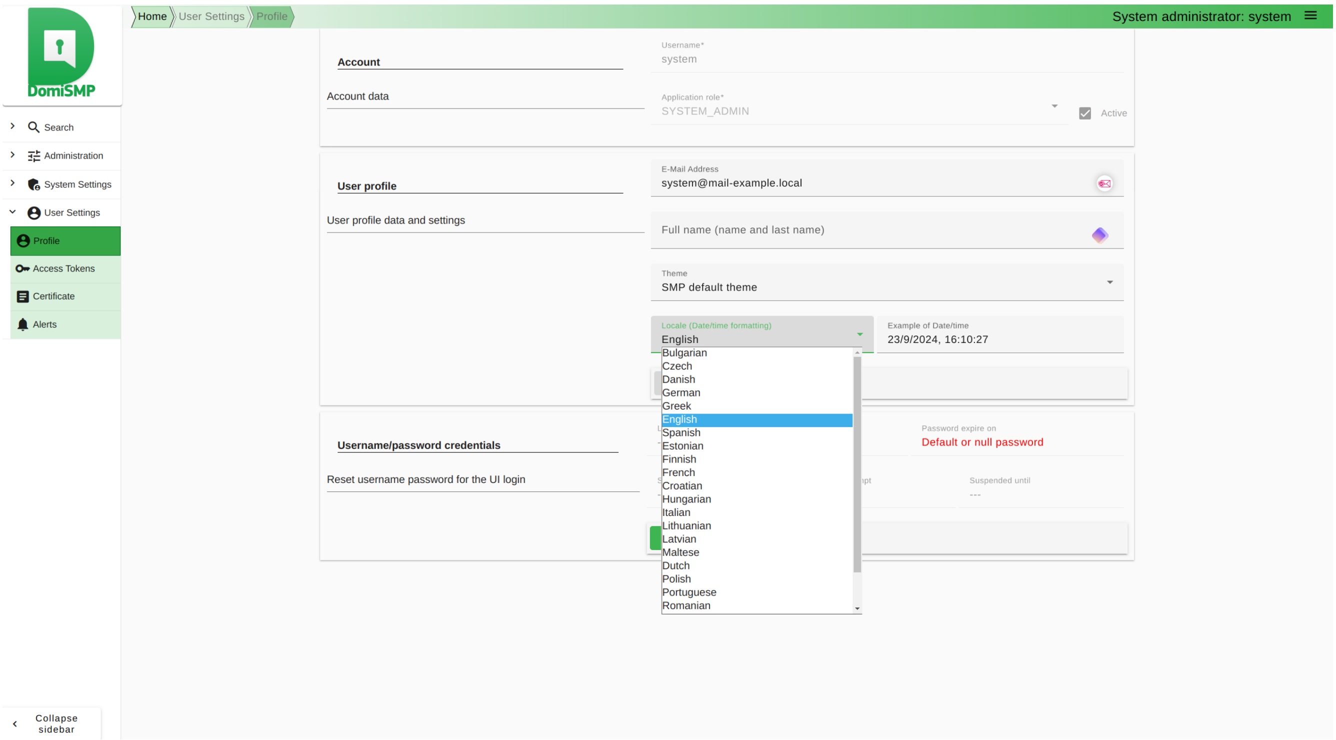
File	Locale
`ui_sl.json`	Slovenian
`ui_sv.json`	Swedish

2.10.5. Setting User’s Language Preference

Once a translation file is created in the locales folder (see Setting Custom Translations) and loaded in DomiSMP, users can then set their preferred language by navigating to the **User Profile** page in the DomiSMP UI and selecting the corresponding value from the dropdown.

> **NOTE**
>
> The DomiSMP Console’s Language preference option in **User Profile** is currently listing all European countries official languages.
> If you would like to see other languages added on the dropdown, please contact eDelivery Support.

If a user selects a language for which a translation file cannot be found, DomiSMP then loads English, the default translation file.
This user preference is remembered between visits.



System administrator: system

Account

Account data

Username*: system

Application role*: SYSTEM_ADMIN

☑ Active

User profile

User profile data and settings

E-Mail Address: system@mail-example.local

Full name (name and last name)

Theme: SMP default theme

Locale (Date/time formatting): English

Example of Date/time: 23/9/2024, 16:10:27

Username/password credentials

Reset username password for the UI login

Password expire on: Default or null password

Suspended until: ---

2.11. SMP SOAP UI

SOAP UI Testing

The SOAP UI can be used to create, update and delete Service Groups and Metadata.

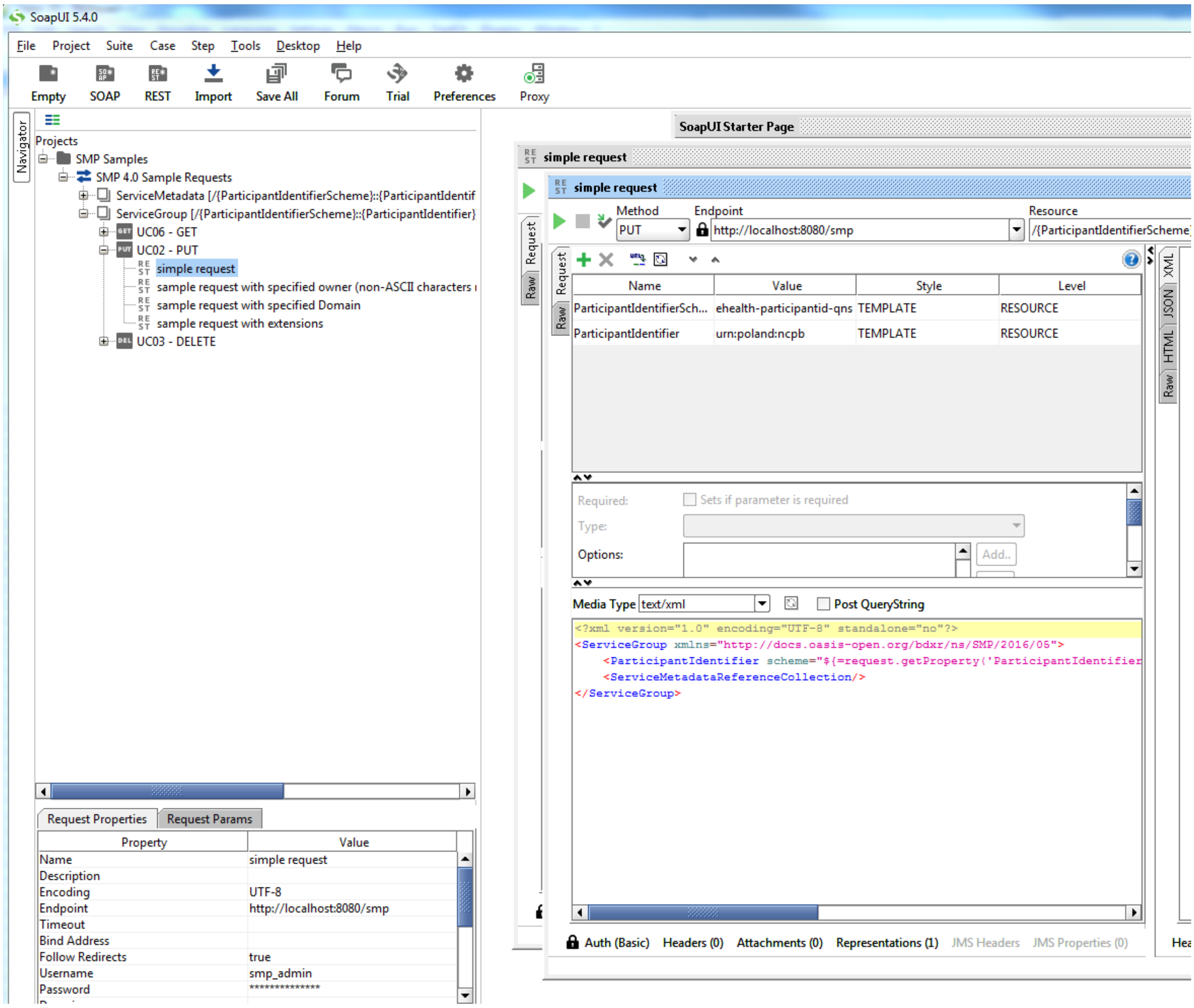
An SMP SoapUI project contains sample requests and is included in the zip file already downloaded.

The procedure to create, update or delete a Service Group is described in the next steps.

2.11.1. Service Groups CRUD Operations

▼ *CREATE Service Group*

In the left navigation pane of the SoapUI interface, browse to the REST PUT method as shown below:


SoapUI 5.4.0
File Project Suite Case Step Tools Desktop Help
Empty SOAP REST Import Save All Forum Trial Preferences Proxy
Navigator
Projects
SMP Samples
SMP 4.0 Sample Requests
ServiceMetadata [/{ParticipantIdentifierScheme}::{ParticipantIdentif
ServiceGroup [/{ParticipantIdentifierScheme}::{ParticipantIdentifier}
UC06 - GET
UC02 - PUT
simple request
sample request with specified owner (non-ASCII characters i
sample request with specified Domain
sample request with extensions
UC03 - DELETE
SoapUI Starter Page
simple request
Request
Raw
XML
JSON
HTML
Auth (Basic) Headers (0) Attachments (0) Representations (1) JMS Headers JMS Properties (0)
Hea

Method	Endpoint	Resource
PUT	http://localhost:8080/smp	/{ParticipantIdentifierScheme}

Name	Value	Style	Level
ParticipantIdentifierSch...	ehealth-participantid-qns	TEMPLATE	RESOURCE
ParticipantIdentifier	urn:poland:ncpb	TEMPLATE	RESOURCE

Required: □ Sets if parameter is required

Type:

Options:

Media Type text/xml □ Post QueryString

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<ServiceGroup xmlns="http://docs.oasis-open.org/bdxr/ns/SMP/2016/05">
    <ParticipantIdentifier scheme="${=request.getProperty('ParticipantIdentifier
    <ServiceMetadataReferenceCollection/>
</ServiceGroup>
```

Request Properties | Request Params

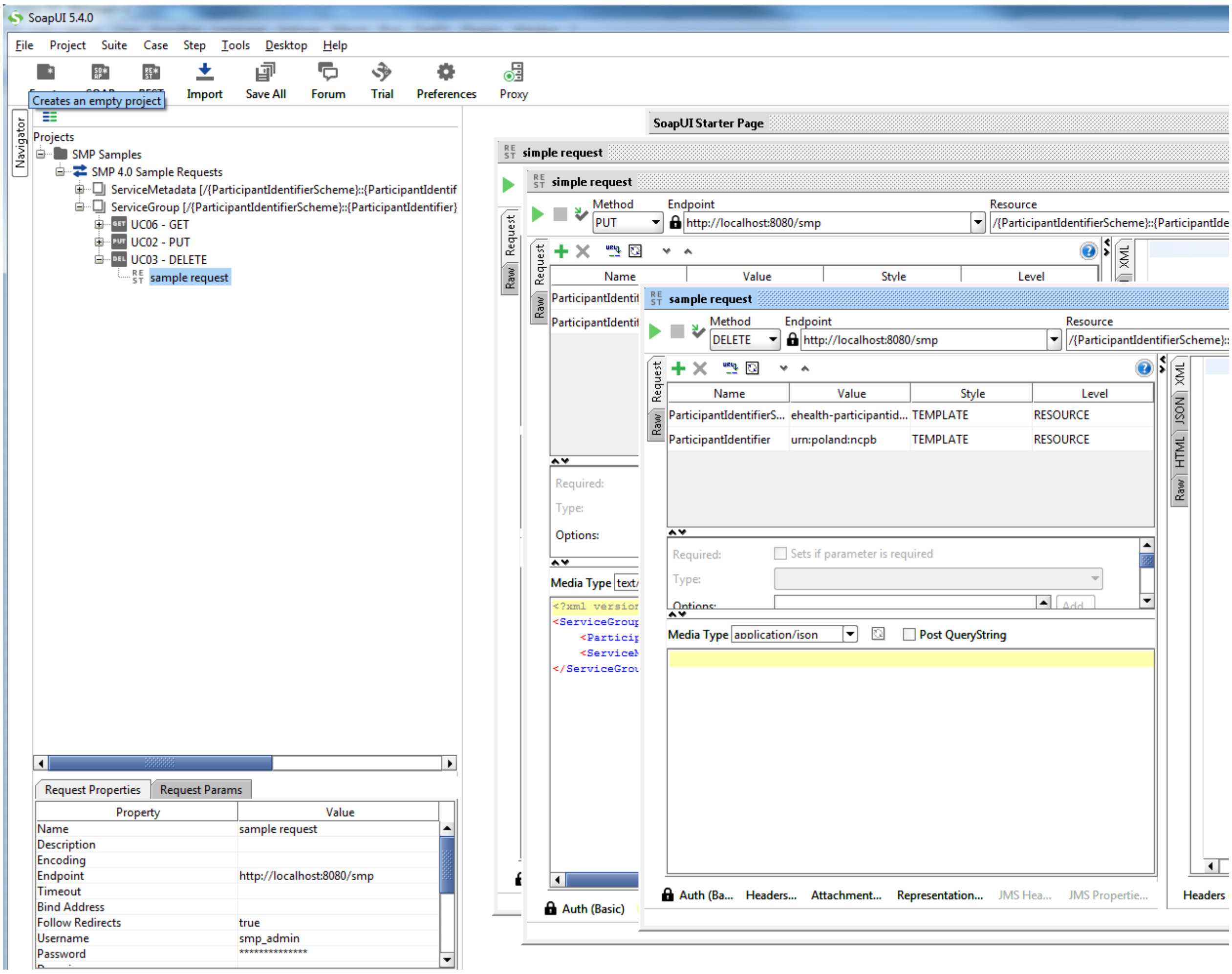
Property	Value
Name	simple request
Description	
Encoding	UTF-8
Endpoint	http://localhost:8080/smp
Timeout	
Bind Address	
Follow Redirects	true
Username	smp_admin
Password	**************

▼ *UPDATE Service Group*

The REST method to update the **ServiceGroup** is the same as the one used for creating **ServiceGroup** described in the previous section.

▼ *DELETE ServiceGroup*

On the SoapUI interface on the left navigation panel, browse to the REST DELETE method as indicated below:



Projects
- SMP Samples
  - SMP 4.0 Sample Requests
    - ServiceMetadata [/{ParticipantIdentifierScheme}::{ParticipantIdentif
    - ServiceGroup [/{ParticipantIdentifierScheme}::{ParticipantIdentifier}
      - UC06 - GET
      - UC02 - PUT
      - UC03 - DELETE
        - sample request

Request Properties | Request Params

Property	Value
Name	sample request
Description	
Encoding	
Endpoint	http://localhost:8080/smp
Timeout	
Bind Address	
Follow Redirects	true
Username	smp_admin
Password	**************

simple request

Method: PUT Endpoint: http://localhost:8080/smp Resource: /{ParticipantIdentifierScheme}::{ParticipantIde

Name	Value	Style	Level
ParticipantIdentif			
ParticipantIdentif			

Required:
Type:
Options:
Media Type text/

```
<?xml versior
<ServiceGroup
    <Particip
    <Service
</ServiceGrou
```

sample request

Method: DELETE Endpoint: http://localhost:8080/smp Resource: /{ParticipantIdentifierScheme}::

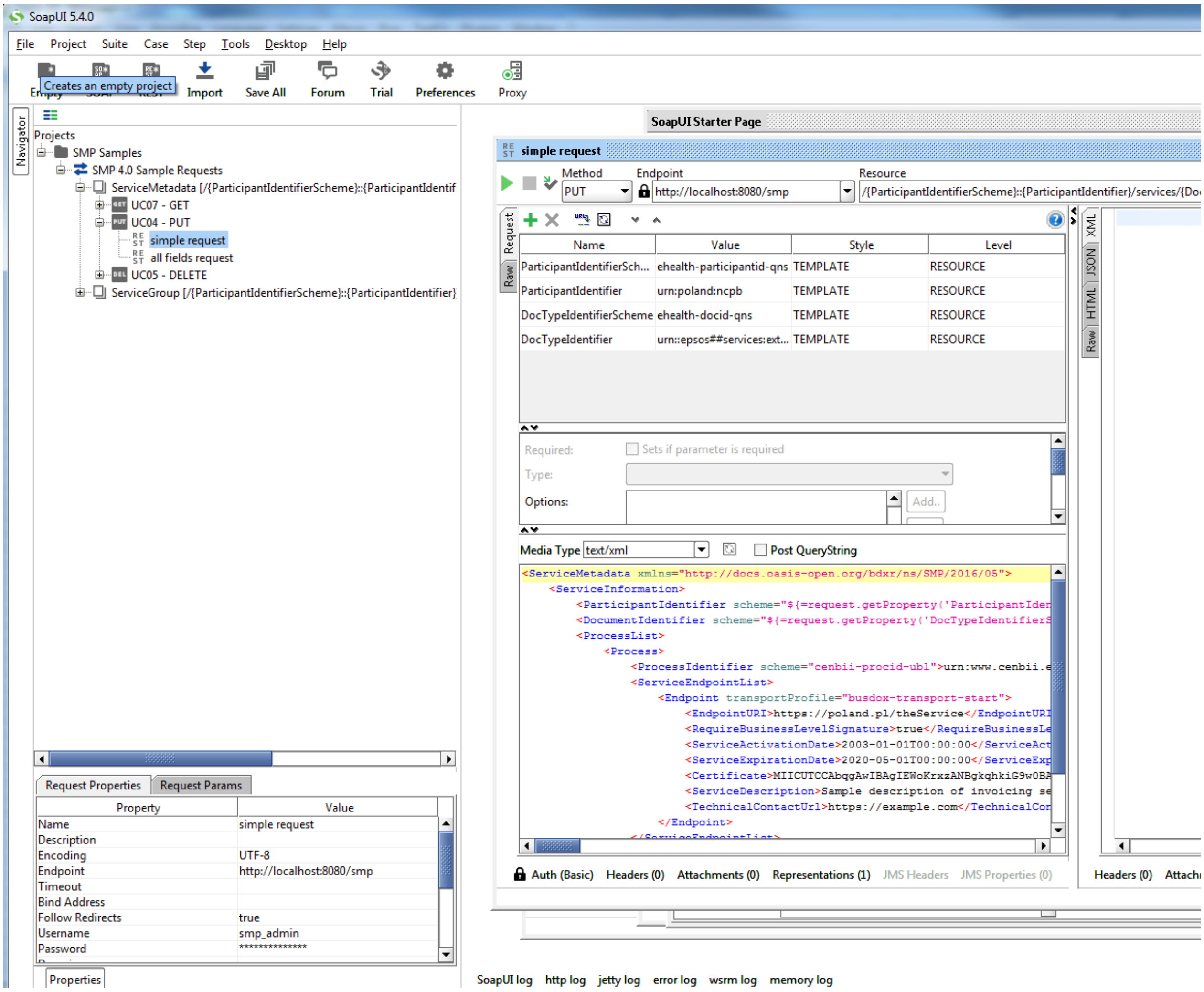
Name	Value	Style	Level
ParticipantIdentifierS...	ehealth-participantid...	TEMPLATE	RESOURCE
ParticipantIdentifier	urn:poland:ncpb	TEMPLATE	RESOURCE

Required: □ Sets if parameter is required
Type:
Options:
Media Type application/json □ Post QueryString

2.11.2. Service Metadata CRUD Operations

▼ *CREATE Service Metadata*

In the left navigation pane of the SoapUI interface, browse to the REST PUT method as shown below:



Method: PUT

Endpoint: http://localhost:8080/smp

Resource: /{ParticipantIdentifierScheme}::{ParticipantIdentifier}/services/{Do

Name	Value	Style	Level
ParticipantIdentifierSch...	ehealth-participantid-qns	TEMPLATE	RESOURCE
ParticipantIdentifier	urn:poland:ncpb	TEMPLATE	RESOURCE
DocTypeIdentifierScheme	ehealth-docid-qns	TEMPLATE	RESOURCE
DocTypeIdentifier	urn::epsos##services:ext...	TEMPLATE	RESOURCE

Required: □ Sets if parameter is required

Type:

Options:

Media Type: text/xml □ Post QueryString

```
<ServiceMetadata xmlns="http://docs.oasis-open.org/bdxr/ns/SMP/2016/05">
    <ServiceInformation>
        <ParticipantIdentifier scheme="${=request.getProperty('ParticipantIden
        <DocumentIdentifier scheme="${=request.getProperty('DocTypeIdentifierS
        <ProcessList>
            <Process>
                <ProcessIdentifier scheme="cenbii-procid-ubl">urn:www.cenbii.e
                <ServiceEndpointList>
                    <Endpoint transportProfile="busdox-transport-start">
                        <EndpointURI>https://poland.pl/theService</EndpointURI
                        <RequireBusinessLevelSignature>true</RequireBusinessLe
                        <ServiceActivationDate>2003-01-01T00:00:00</ServiceAct
                        <ServiceExpirationDate>2020-05-01T00:00:00</ServiceExp
                        <Certificate>MIICUTCCAbqgAwIBAgIEWoKrxzANBgkqhkiG9w0BA
                        <ServiceDescription>Sample description of invoicing se
                        <TechnicalContactUrl>https://example.com</TechnicalCon
                    </Endpoint>
```

Request Properties | Request Params

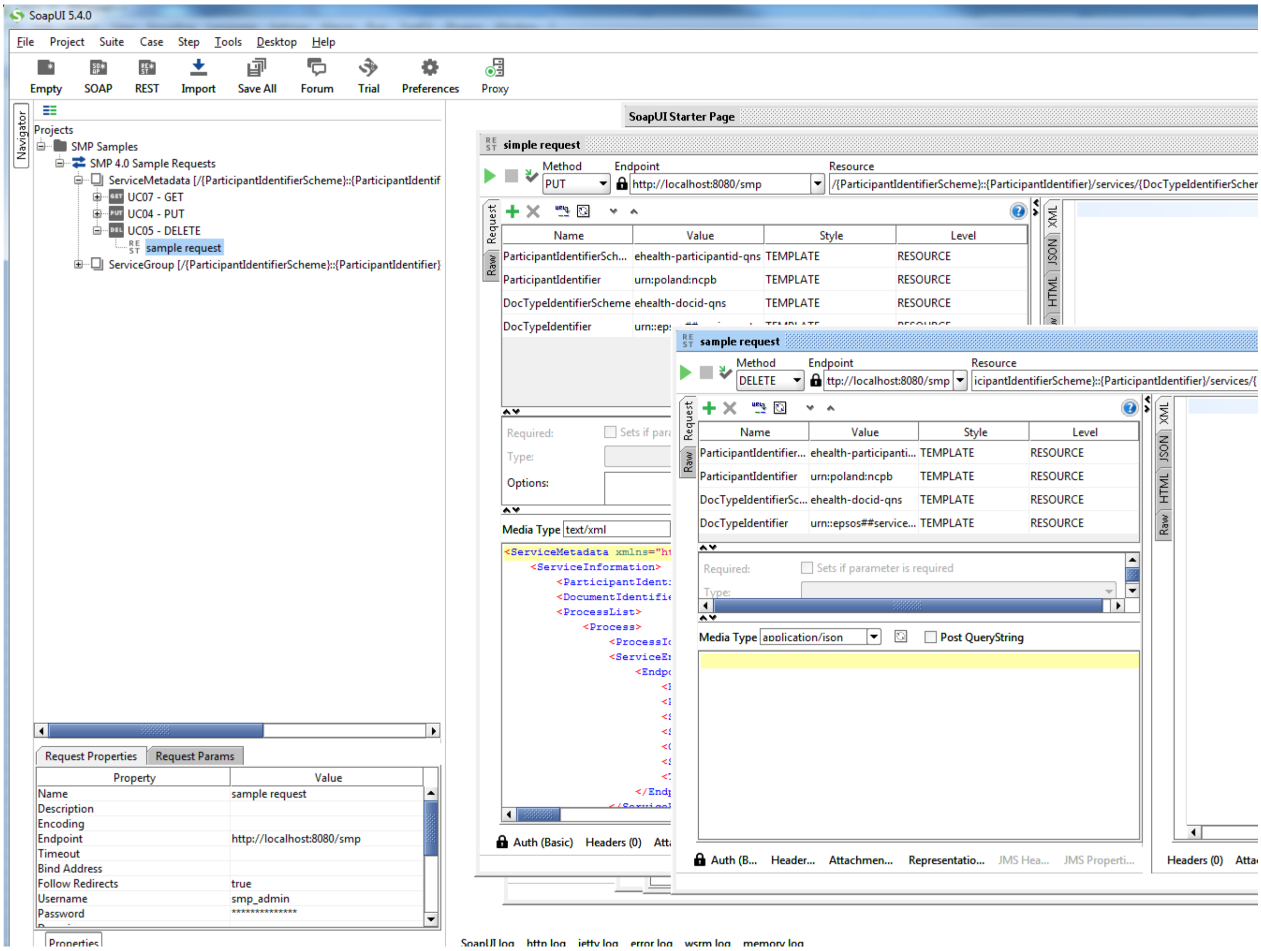
Property	Value
Name	simple request
Description	
Encoding	UTF-8
Endpoint	http://localhost:8080/smp
Timeout	
Bind Address	
Follow Redirects	true
Username	smp_admin
Password	**************

▼ *UPDATE Service Metadata*

The REST method to update `ServiceMetadata` is the same as the one use for creating `ServiceMetadata` as described in the previous section.

▼ *DELETE Service Metadata*

In the left navigation pane of the SoapUI interface, browse to the `REST DELETE` method as indicated below:


SoapUI 5.4.0
File Project Suite Case Step Tools Desktop Help
Empty SOAP REST Import Save All Forum Trial Preferences Proxy
Navigator
Projects
SMP Samples
SMP 4.0 Sample Requests
ServiceMetadata [/{ParticipantIdentifierScheme}::{ParticipantIdentif
UC07 - GET
UC04 - PUT
UC05 - DELETE
sample request
ServiceGroup [/{ParticipantIdentifierScheme}::{ParticipantIdentifier}
SoapUI Starter Page
simple request
sample request
Request
Raw
XML
JSON
HTML
Auth (Basic) Headers (0) Atta
Auth (B... Header... Attachmen... Representatio... JMS Hea... JMS Properti...
Headers (0) Atta
Request Properties Request Params
Properties
SoapUI log http log jetty log error log wsrm log memory log

Method: PUT | Endpoint: http://localhost:8080/smp | Resource: /{ParticipantIdentifierScheme}::{ParticipantIdentifier}/services/{DocTypeIdentifierSche

Name	Value	Style	Level
ParticipantIdentifierSch...	ehealth-participantid-qns	TEMPLATE	RESOURCE
ParticipantIdentifier	urn:poland:ncpb	TEMPLATE	RESOURCE
DocTypeIdentifierScheme	ehealth-docid-qns	TEMPLATE	RESOURCE
DocTypeIdentifier	urn::eps		

Required: □ Sets if para

Type:

Options:

Media Type: text/xml

```
<ServiceMetadata xmlns="ht
   <ServiceInformation>
      <ParticipantIdent
      <DocumentIdentifi
      <ProcessList>
         <Process>
            <ProcessI
            <ServiceE
               <Endp
               </Endp
```

Method: DELETE | Endpoint: ttp://localhost:8080/smp | Resource: icipantIdentifierScheme}::{ParticipantIdentifier}/services/{

Name	Value	Style	Level
ParticipantIdentifier...	ehealth-participanti...	TEMPLATE	RESOURCE
ParticipantIdentifier	urn:poland:ncpb	TEMPLATE	RESOURCE
DocTypeIdentifierSc...	ehealth-docid-qns	TEMPLATE	RESOURCE
DocTypeIdentifier	urn::epsos##service...	TEMPLATE	RESOURCE

Required: □ Sets if parameter is required

Type:

Media Type: application/json □ Post QueryString

Property	Value
Name	sample request
Description	
Encoding	
Endpoint	http://localhost:8080/smp
Timeout	
Bind Address	
Follow Redirects	true
Username	smp_admin
Password	**************

2.12. Compiling SMP

2.12.1. Compilation Prerequisites

Supported Operating System Platform

The eDelivery SMP can be built on the following OS platforms:

- Windows Workstation & Server
- Linux platform

Software Requirements

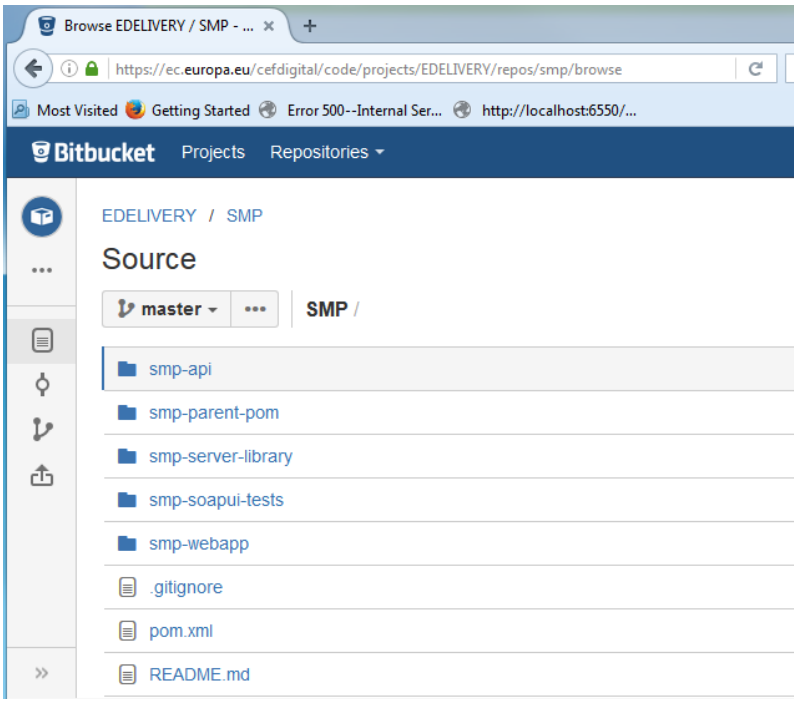
The following software components are required on the target system:

- Java Development Kit environment (JDK), version 8: http://www.oracle.com/technetwork/java/javase/downloads/index.html
- Maven 3.6 and above (https://maven.apache.org/download.cgi)
- GIT (optional: Git is only used to download the project sources but these sources can be downloaded from any system having Git installed and then just copied manually on the compilation platform).

2.12.2. Downloading the Source Code

The source code of SMP is freely available and can be downloaded from the following location:

https://ec.europa.eu/digital-building-blocks/code/projects/EDELIVERY/repos/smp/browse


Browse EDELIVERY / SMP - ...
https://ec.europa.eu/cefdigital/code/projects/EDELIVERY/repos/smp/browse
Most Visited
Getting Started
Error 500--Internal Ser...
http://localhost:6550/...
Bitbucket
Projects
Repositories
EDELIVERY / SMP
Source
master
SMP /

smp-api
smp-parent-pom
smp-server-library
smp-soapui-tests
smp-webapp
.gitignore
pom.xml
README.md

2.12.3. Compiling SMP Source Code

1. Create a new `/comp_dir` temporary directory.
2. From `/comp_dir`, execute:

```
git clone https://ec.europa.eu/digital-building-blocks/code/scm/edelivery/smp.git
```

3. Go to the newly created `/smp` directory.

   Using a ls command in the above-mentioned directory renders the following contents:

```
pom.xml README.md smp-api smp-parent-pom smp-server-library smp-soapui-tests smp-webapp
```

4. Start the compilation by executing the following command:

```
mvn clean install -DskipTests
```

5. A successful compilation will result with the following:

```
mvn clean install -DskipTests
```

Expected Console Output

```
[INFO] Scanning for projects…
```

```
/..
../

[INFO] Installing /home/smpcomp/smp/smp/pom.xml to
/home/smpcomp/.m2/repository/eu/europa/ec/smp/3.X/smp-3.X.pom

[INFO]
------------------------------------------------------------------------
[INFO] Reactor Summary:
[INFO]
[INFO] smp-angular ........................................ SUCCESS [132.375 s]
[INFO] smp-api ............................................ SUCCESS [32.375 s]
[INFO] smp-server-library ................................. SUCCESS [02:01 min]
[INFO] smp-webapp ......................................... SUCCESS [23.314 s]
[INFO] SMP Builder POM .................................... SUCCESS [2.222 s]
[INFO]
------------------------------------------------------------------------


[INFO] BUILD SUCCESS

[INFO]
------------------------------------------------------------------------
[INFO] Total time: 03:00 min
[INFO] Finished at: 2017-06-08T11:35:27+02:00
[INFO] Final Memory: 61M/726M
[INFO]
------------------------------------------------------------------------
```

As a result the web application, `smp.war` , is created in the `/smp-webapp/target/` directory.
Using a ls command in the above-mentioned directory renders the following contents:

```
smp-X smp.war classes generated-sources generated-test-sources maven-status test-
classes webapp-classes
```

2.13. SMP Admin Console

The SMP Admin console enables:

- Anonymous users to search and explore published data in the SMP.
  Anonymous users can search for participants by participant ID, schema, or domain.
- Service Group administrators to manage owned Service groups; SMP administrators to manage Service groups registered on SMP, and System Administrators to manage users and domains

Admin Console URL

The Admin console dashboard is reachable via the following URLs:

```
http://[host]:[port]/smp%5b-version%5d/iu/[http://[host]:[port]/smp[-version]/iu/]
```

If the deployment package (war file) filename changed in order to simply upgrade the old SMP version as for example `smp-4.0.0.war` to `cipa-smp-full-webapp.war`, then the application root context might change as well.

Example:

```
http://wlal0079a.cc.cec.eu.int:1043/cipa-smp-full-webapp/ui/[http://
[host]:[port]/cipa-smp-full-webapp/ui/].
```

Roles

Two types of application roles are defined in the SMP Admin Console:

- **System Administrator**: this is a *super admin* who can manage SMP users and domains.
- **User**: a regular user of the DomiSMP: the user can administer Domains, Groups and Resources according to membership roles described in § 11-SMP User Management.

When users are logged, their role is displayed in read-only mode (as a label). Only the System Administrator can change the role of another user.

[1] to change immediately for security reasons

Chapter 3. Interface Description

3.1. Introduction

eDelivery building blocks helps public administrations exchange electronic data and documents with other public administrations, businesses and citizens, in an interoperable, secure, reliable and trusted way. By using 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.

▼ *Guide’s Purpose*

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 the Digital Europe Programme on the other hand.

▼ *Scope*

This document is a high-level functional definition of the services provided by SML. 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.

▼ *Audience*

This document is intended for:

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

▼ *Concept Definitions*

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

- [REF3]
- [REF4]
- [REF7]

▼ *References*

#	Document	Contents outline
[REF1]	What is eDelivery?	Overview of eDelivery
[REF2]	Using HTTP Methods for RESTful Services	Short description of HTTP Methods for RESTful Services

#	Document	Contents outline
[REF3]	***OpenPEPPOL AISBL - Policy for use of Identifiers***	
[REF4]	***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.
[REF5]	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.
[REF6]	***eSens Building Blocks - PR - SMP***	e-SENS will use SML (Simple Metadata Publisher) specification originally developed by PEPPOL and generalised and standardised by OASIS.TheSMPspecification usually complements the Location LookUp ABB.
[REF7]	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).
[REF8]	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.
[REF9]	Business Document Exchange Network - Common Definitions, CommonDefinitions.pdf	This document contains the definitions and terms that are common between the Business 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.

#	Document	Contents outline
[REF10]	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.
[REF11]	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.
[REF12]	***OASIS - Service Metadata Publishing (SMP) Version 2.0***	This document describes the version 2.0 of the Oasis SMP standard.

IMPORTANT

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

3.2. Positioning SMP in eDelivery

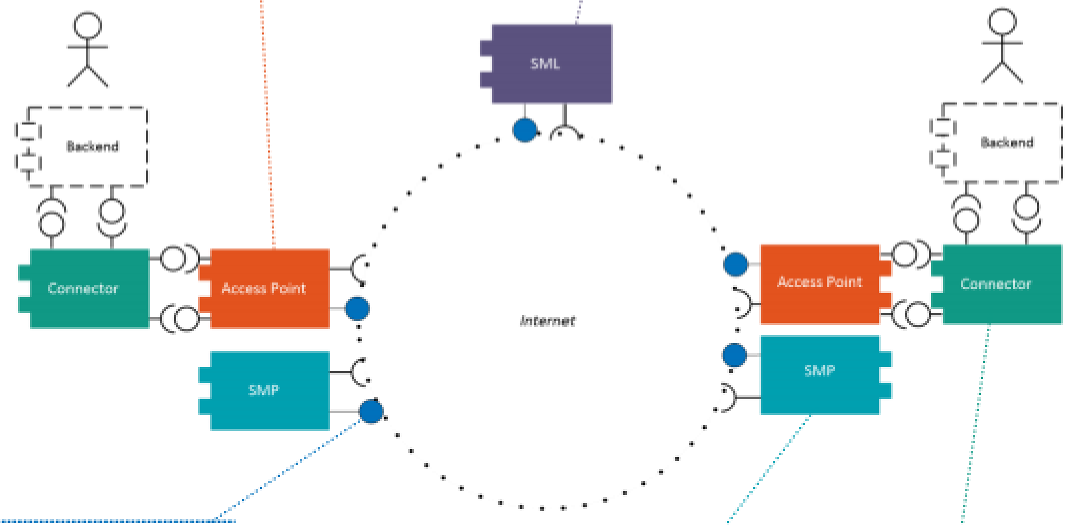
▼ **eDelivery in a nutshell**

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

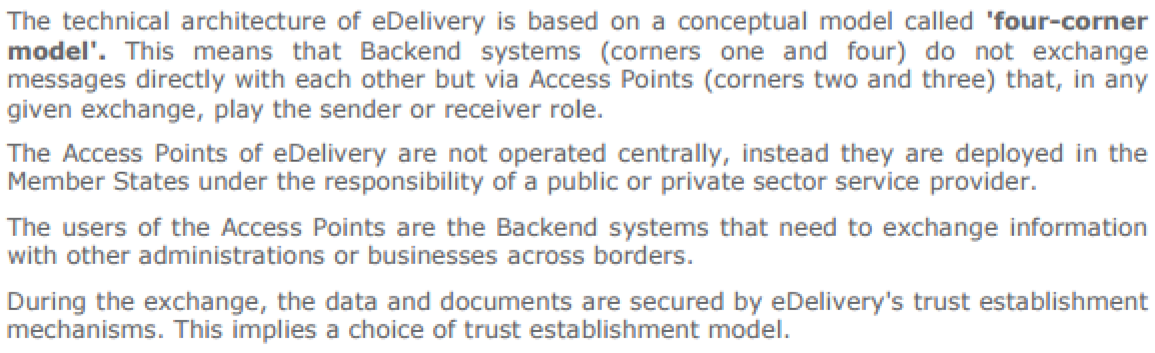
In order to activate this exchange, two public administrations' Access Points need to establish trust between each other. This is done through digital certificates.

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.

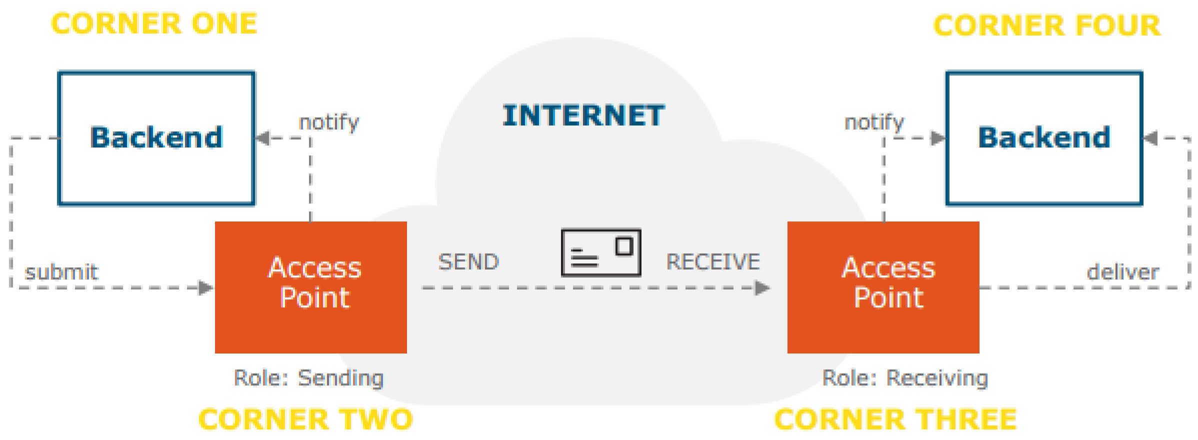


The technical architecture of eDelivery is based on a conceptual model called **'four-corner model'.** This means that Backend systems (corners one and four) do not exchange messages directly with each other but via Access Points (corners two and three) that, in any given exchange, play the sender or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



3.2.1. SMP role

The role of SML in the Four Corner Model is 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,
- **clients** (*senders*) to find out the definitions of those services.

To that end, SMP provides services for:

- **receivers** to register services definitions (such as “put metadata”);
- **senders** to consult those definitions (“retrieve metadata”).

3.2.2. SMP/SML Interactions

To promote the consistency of the whole process SMP sends location information to SML of:

- **SMP**'s own location to allow senders to discover SMP;
- **all Access Points** providing access to declared `ServiceGroups` of the participants SMP is managing.

SML Management Services

SML exposes multiple management services allowing SML to declare new location information or any related changes. They are:

Manage participant identifiers interface

the interface for Service Metadata publishers for managing the metadata relating to specific participant identifiers that they make available.

Manage service metadata interface

the interface for Service Metadata publishers for managing the metadata about their services, e.g., binding, interface profile and key information.

NOTE

TManage participant identifiers interface and Manage service metadata interface are not detailed here but the document will refer to these when they are invoked from SML REST services.
Refer to the `Execution` sections of the REST Services definitions below for further details on these interactions.

See also [REF11].

SML also 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.

NOTE

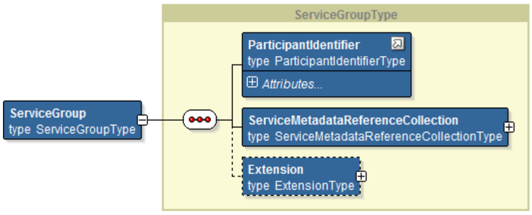
This functionality isn’t addressed currently, but is planned for a future release.
The use cases envisioned for this functionality are:
UC08 - Register SMP
UC09 - Change SMP Location
UC10 - Unregister SMP
UC11 - Migrate Metadata SMP

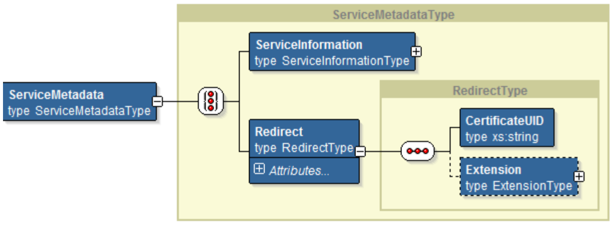
3.3. Data model

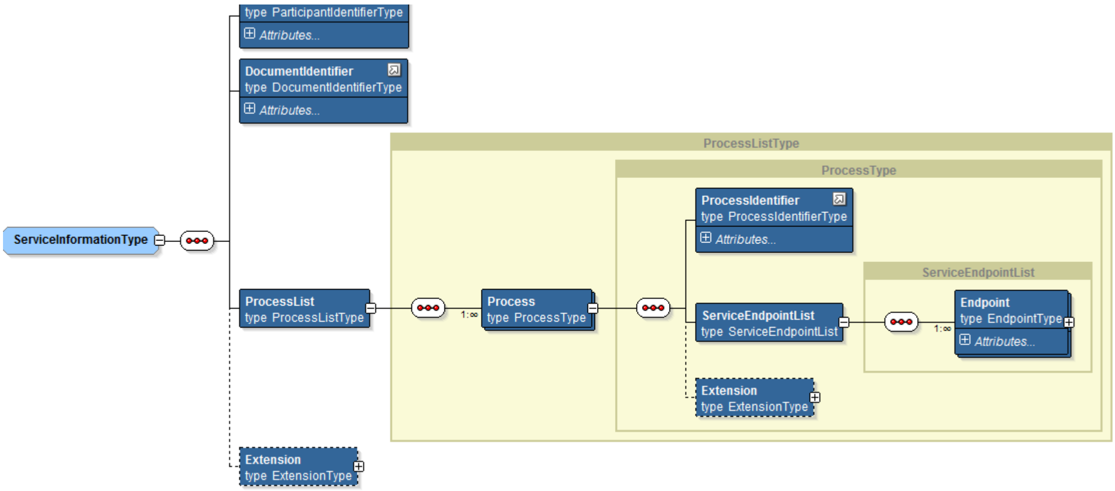
The SMP interface is built around the data it is intended to manage. Therefore, this document starts by defining the data itself.

3.3.1. Logical data model

The diagram below depicts the major parts of the data model describing the configuration held by SML and managed through the interface described in this document. This model is another view of the XSD definition that can be found in the XSD Files section.







▼ *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. [REF7], Data model).
Refer to [REF4] § 2.4 “Identifiers” for Oasis SMP 1.0 or [REF11] § 3.0 “Identifiers” for Oasis SMP 2.0 to find more details and additional references about identifiers of participants (/businesses), documents and processes.

▼ *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 [REF4] § 2.4 “Identifiers” for Oasis SMP 1.0 or [REF11] § 3.0 “Identifiers” for Oasis SMP 2.0 to find more details and additional references about identifiers of participants (/businesses), documents and processes.

▼ *Process*

You can find the following definitions for `ServiceMetadata`:

```
"a structure that reresents 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 [⋯]□
```

```
"⋯ 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)."
"It also describes which business processes a document can participate [⋯]"
```

▼ *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 [REF4] and [REF11])

XSD element	Description
endpointURI Oasis SMP 1.0 Element: /ServiceEndpointList/ Endpoint/EndpointURI Oasis SMP 2.0 Element: /sma:ProcessMetadata/sma:Endpoint/smb:AddressURI	The address of an endpoint, as a URL.
transportProfile Oasis SMP 1.0 Element: ServiceInformation/ ProcessList/../Endpoint/ @transportProfile Oasis SMP 2.0 Element: /ServiceMetadata/sma:ProcessMetadata/sma:Endpoint/smb:TransportProfileID	Indicates the type of transport method that is being used between access points
requireBusinessLevelSignature Oasis SMP 1.0 Element: ServiceInformation/ProcessList/../Endpoint/	Indicates the type of transport method that is being used between access points.

XSD element	Description
requireBusinessLevelSignature Oasis SMP 1.0 Element: ServiceInformation/ ProcessList/../Endpoint/ RequireBusinessLevelSignature Oasis SMP 2.0 Element: /	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 Oasis SMP 1.0 Element: ServiceInformation/ ProcessList/../Endpoint/ MinimumAuthenticationLevel Oasis SMP 2.0 Element: /	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.

XSD element	Description
serviceActivationDate Oasis SMP 1.0 Element: ServiceInformation/ ProcessList/../Endpoint/ ServiceActivationDate Oasis SMP 2.0 Element: sma:ProcessMetadata/sma:Endpoint/smb:ActivationDate	Activation date of the service. Senders should ignore services that are not yet activated. Format of ServiceActivationDate date is xs: dateTime.
serviceExpirationDate Oasis SMP 1.0 Element: /ProcessList/../Endpoint/ ServiceExpirationDate Oasis SMP 2.0 Element: sma:ProcessMetadata/sma:Endpoint/smb:ExpirationDate	Expiration date of the service. Senders should ignore services that are expired. Format of `ServiceExpirationDate` date is `xs:dateTime`.
certificate Oasis SMP 1.0 Element: /ProcessList/../Endpoint/ Certificate Oasis SMP 2.0 Element: /ServiceMetadata/sma:ProcessMetadata/sma:Endpoint/sma:Certificate	Holds the complete **[X509v3]** signing certificate of the recipient gateway, as a PEM base 64 encoded DER formatted value.
serviceDescription Oasis SMP 1.0 Element: /ProcessList/../Endpoint/ ServiceDescription Oasis SMP 2.0 Element: /	A human-readable description of the service

XSD element	Description
technicalContactUrl Oasis SMP 1.0 Element: /ProcessList/../Endpoint/ TechnicalContactUrl Oasis SMP 2.0 Element: /ServiceMetadata/sma:ProcessMetadata/sma:Endpoint/smb:Contact	Represents a link to human-readable contact information. This might also be an email address.
technicalInformationUrl Oasis SMP 1.0 Element: /ProcessList/../Endpoint/ TechnicalInformationUrl Oasis SMP 2.0 Element:	A URL to human-readable documentation of the service format. This could for example be a website containing links to XML Schemas, WSDLs, Schematrons and other relevant resources.
extension Oasis SMP 1.0 Element: /Process/Extension Oasis SMP 2.0 Element: /ServiceMetadata/sma:ProcessMetadata/ext:SMPExtensions	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 Oasis SMP 1.0 Element: /ServiceInformation/ Extension Oasis SMP 2.0 Element: /	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.

3.3.2. 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 storage and the retrieval of participant's information (cf. §3.1.1 – "Original official OASIS SMP XSD").

   [image8] | *image8.emf*

2. The second one, defined in this document (cf. 3.1.2 – "Extended SMP XSD ") defines the structure of error messages.

   [image29] | *image29.emf*

   Resource Admin
   Group Admin
   SMP
   Participant

   [image8]

3. The second one, defined in this document (cf. 3.1.2 – "Extended SMP XSD ") defines the structure of error messages.

   [image29]

Resource Admin
Group Admin
SMP
Participant

1 2

Use cases summary

Table 1. Actors

Actor	Definition
System Admin	A user granted rights to administer the Domain Admin type of users. This role is symbolised by 4 stars (it has the highest authority). The system admin is application role with permissions to configure SMP systems settings.
Domain Admin [image32] | *image32.emf*	A user granted to administer the Domain groups. The domain groups are group of users in the domain/network. The user can create/delete groups and manage the memberships of the groups. The role is symbolised by 3 stars (it has the authority to create/delete and assign the Group admin users to the groups)

Actor	Definition
Group Admin [image33] | *image33.emf*	A user granted rights to administer the participants for the group. The group admin administers the resources (Service groups) and assigns the memberships to the resources. This role is symbolised by 2 stars (it has the authority to create/delete resources (ServiceGroups) and to assign the admin users to the resources)
Resource Admin [image34] | *image34.emf*	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 symbolised by 1 single stars (it has the authority to administer (update) the resource (service groups) and subresources (ServiceMetadata), but cannot create or delete the resource)
User	Any participant sending documents to any other receiver participant and consulting SML in that purpose This role is symbolised by no single star since he has only public read accesses

In addition to the above described roles, two additional terms are used:

Sender

to refer to an actor who uses the system (SML) 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 (SML) on the right hand side of the same model. In the present use cases, the receiver will behave as “Resource Admin.

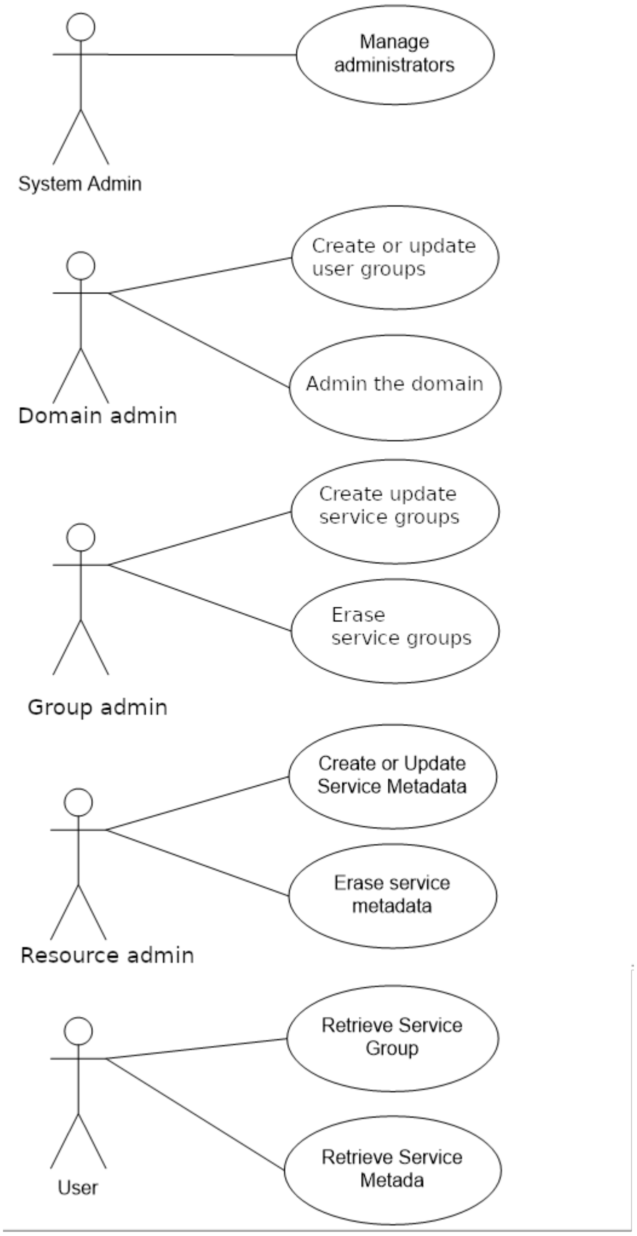
The System Admin being neither on the left nor on the right of that model, but rather on top of it, will never be referred to as ‘sender’ nor ‘receiver’.

Use cases diagram

[image35] | *image35.emf*

[image30] | *image30.emf*

[image36] | *image36.emf*



ID	Actor	UC	Description	Operation	Data
UC01	System Admin	**Manage Administrators**	Create and modify user information i‘ SMP table 'A’ministrator'	n/a	User (table)
UC02	Group Admin	**Create or Update Resources/Service Group**	Create a new ServiceGroup for a new receiver participant. This service stores the *Service Group* and links it to the specified pair participantIdentifier + participantIndentifierScheme. Information is store into ServiceGroup table. This same service is used to create and update a ServiceGroup.	PUT	ServiceGroup

ID	Actor	UC	Description	Operation	Data
UC03	Group Admin	**Erase Resource/Service Group**	Erases the service group definition AND the list of services for the specified receiver participant.	DELETE	ServiceGroup
UC04	Resource Admin	**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	Resource Admin	**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
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

Stories

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 SML (the System) is involved. If 'N/A' is mentioned, some action part of the 'story' happens without any involvement of SML.

- UC01: As a `System Admin`, I create a new 'Group Admin' to allow the creation and the management of a new ServiceGroup for a participant.
- UC01: As "System Admin", I create a new 'Resource Admin' to allow the creation and the

management of the metadata of that new ServiceGroup.

- UC02: As "Group Admin", I create a new ServiceGroup and link it to the administrator “'Resource Admin” to allow the management of ServiceMetadata for the related participant.
- UC04: As "Resource Admin", I define ALL the ServiceMetadata for the participant that I administer.
- N/A: As" User", I ask the DNS to resolve the address of SML 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.

3.4. 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`).

UC01 Manage Administrators

This use case introduces the foundation for an administration console: creating an ‘Group Admin’ 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.

▼ *Use Case Description*

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	

Brief description	
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 ServiceGroups 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

▼ *REST Service: None*

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

UC02 Create/Update Service Group

▼ *Use Case Description*

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 stored into ServiceGroup table. This same service is used to create and update a ServiceGroup.						
Actors						
Group Admin						
Preconditions						

Brief description						
The authenticated user has the role of " Group Admin"						
If the ServiceGroup is managed remotely, the "Resource Admin" must have been created before in the "Administrator" table.						
Identifier and scheme of the service group provided in the request must comply to the policy defined in [REF3]						

Brief description						
If SML is serving multiple domains, the header field "Domain" must be populated and refer to one of the domains served by SML.						
Basic flow event						
Step						
1	The receiver declares its service group and the related Administrator (Resource Admin) to SML					

Brief description						
2	The SMP authenticates the user, validates the request, and add or replace the information into its configuration database and passes the information to SML.					
3	The receiver receives the confirmation that the definitions were stored properly with HTTP response "201 Created".					
4	Use case ends with success					

Brief description						
Alternative flows						
3a	**Service Group 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					

Brief description						
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)**					

Brief description						
2b1	The SMP replies with HTTP error "400 Bad request" or "500 Internal server 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					

Brief description						
2c	**SMP is serving multiple domains and the Domain field is not specified in the header**					

Brief description						
2c1	The SMP replies with HTTP error "400 Bad request" (business code WRONG_FIELD) with message: "SMP is configured to use multiple domains, but no Domain is specified in request. Please specify Domain in request."					
2c2	The receiver receives the error message					
2c3	Use case ends					

Brief description						
2d	**Domain field refers to a domain that is not served by SML**					
2d1	The SMP replies with HTTP error "400 Bad request" (business code WRONG_FIELD) with message: Requested domain does not exist " (followed by the domain field value)					
2d2	The receiver receives the error message					
2d3	Use case ends					

Brief description						
Post conditions						
Successful conditions						
	Service Group is either created or updated, and the corresponding "'Resource Admin" 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					

▼ *REST Service: PutServiceGroup*

Input: * **In the URL**: **The participant’s identifier and identifier’s scheme**

(ParticipantIdentifier) * In the header (optional fields): the Certificate Identifier required for authenticating the remote user as "Resource Admin" for this service group.

NB: if the Certificate Identifier is not provided, the "Group Admin " will manage the metadata of that Service Group – the username of the basic authentication is used to identify the "Group Admin " to link him with the Service Group.

- The "Domain" for which the ServiceGroup must be created.

NB: this field is optional and relevant only if SML is serving multiple domains. In that case this field must be provided.

- **In the TEXT:** a ServiceGroup structure as defined in the standard OASIS XSD (cf. 3.1.1 - "Original official OASIS SMP XSD") containing:
  - The Participant's identifier and scheme that uniquely identifies this service group; These must be identical to the ones provided in the URL.
  - Optionally, the Extension information in the HTTP TEXT

Details on the *ServiceGroup-Owner* structure:

- The following attributes of the certificate will be used in this order:
- CN,
- O,
- C, and
- Serial number
- As an example, the following certificate:

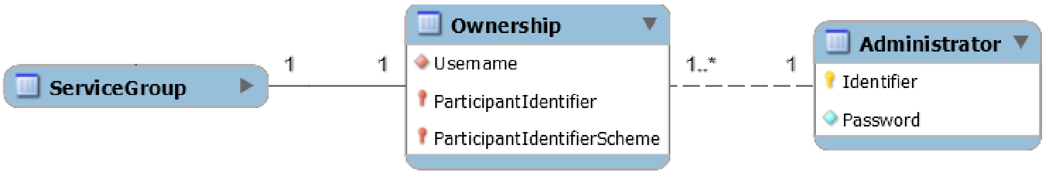
```
sno=0001&subject=EMAILADDRESS=receiver@test.be, CN=SMP_receiverCN,
OU=B4, O=DIGIT, L=Brussels, ST=BE, C=BE:df48f09389f034&validfrom=Jun 1
10:37:53 2015 CEST&validto=Jun 1 10:37:53 2035
CEST&issuer=EMAILADDRESS=root@test.be,CN=rootCN,OU=B4,O=DIGIT,L=Brussels,ST=BE,C=BE
```

will be provided as such in the HTTP header:

ServiceGroup-Owner: CN=SMP_receiverCN, O=DIGIT, C=BE:df48f09389f034

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 attribute *ServiceGroup-Owner* is present in the HTTP Header, then use this as information to store as "Identifier"
- If not, store instead the basic authentication information provided in the HTTP header.
- If it is a newly created ServiceGroup, invoke SML's Create Business Identifier service.
- If it is an existing ServiceGroup, invoke SML's Delete Business Identifier" and Create Business Identifier services.
- if the service invocation succeeds, commit the transaction.
- if the service invocation fails:
    - rollback the transaction;
    - if necessary (delete succeeded), try to reinvoke the Create Business Identifier service with the old information to restore SML properly;
    - The Response to this service is "failure".

Output:

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

Sample Request

HTTP Header (No AdminServiceGroup authentication information – Group Admin creates or updates the ServiceGroup)

PUT http://smp-digit-mock.publisher.ehealth.acc.edelivery.tech.ec.europa.eu/ehealth-actorid-qns::urn:poland:ncpb HTTP/1.1

Host: smp-digit-mock.publisher.ehealth.acc.edelivery.tech.ec.europa.eu Accept: application/xml Content-Type: application/xml Authorization: Basic dXNlcjpwYXNz Content-Length: 278

HTTP Header (Resource Admin authentication information is certificate)

Host: smp-digit-mock.publisher.ehealth.acc.edelivery.tech.ec.europa.eu Accept: application/xml Content-Type: application/xml ServiceGroup-Owner:CN=SMP_1000000181,O=DIGIT,C=DK:406b2abf0bd1d46ac4292efee597d414 Authorization: Basic dXNlcjpwYXNz Content-Length: 278

Text

```xml
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<ServiceGroup xmlns="http://docs.oasis-open.org/bdxr/ns/SMP/2016/05">
<ParticipantIdentifier
scheme="ehealth-actorid-qns">urn:poland:ncpb</ParticipantIdentifier>
<ServiceMetadataReferenceCollection/>
</ServiceGroup>
```

Sample Response

HTTP header

HTTP/1.1 [.mark]#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 showed in the above example.

Text N/A

Table 2. Error Codes

HTTP code	HTTP Message	Business code	Meaning
`200`	`OK`	n/a	The request was completed successfully.
`201`	`Created`	n/a	The PUT operation completed successfully.
`400`	`Bad Request`	`XSD_INVALID`	The XML included in the request is not validate against the XSD defining the input structure.
		`MISSING_FIELD`	Some field that is optional in the XSD but mandatory for this invocation is missing (missing field name in description).
		`WRONG_FIELD` aa	Some field is valid against XSD definition, but the more specific content is invalid (erroneous field name in description). Or Some header field is either missing or invalid.
		`FORMAT_ERROR`	Some field is expected to have a specific format is not valid (erroneous field name in description).
		`USER_NOT_FOUND`	The referenced " Resource Admin" was not found as Administrator.
`401`	`Unauthorized`	`UNAUTHORIZED`	The user is not granted the right to issue this request.
`404`	`Resource not found`	`NOT_FOUND`	The requested information was not found.

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

NOTE	The business code and the description are in the response and compliant to the ErrorResponseType as described in §3.3 – "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`

UC03 Erase Service Group

▼ *Use Case Description*

Brief description						
Erases the service group definition AND the list of services for the specified receiver participant.						
Actors						
Group Admin						
Preconditions						
The authenticated user has the role of " Group Admin".						
Referenced service group was previously defined.						
Basic flow event						
Step						

Brief description						
1	The receiver requests its service group to be removed from SML.					
2	The SMP authenticates the user, validates the request, and removes all the information on the service group from its configuration and from SML.					

Brief description						
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.					

Brief description						
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)**					

Brief description						
2b1	The SMP replies with HTTP error "400 Bad request" or "500 Internal server 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	**Service Group is not defined**					

Brief description						
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						

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

▼ *REST Service: DeleteServiceGroup*

<u>Input</u>: ServiceGroup identifier: ParticipantIdentifier, ParticipantIdentifierScheme in the HTTP header

<u>Execution</u>:

The username or the certificate from the HTTP header 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 “<u>Delete</u> Business Identifier”.

If SML service invocation succeeded, commit the transaction.

If SML service invocation failed:

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

<u>Output</u>: 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://smp-digit-mock.publisher.ehealth.acc.edelivery.tech.ec.europa.eu/ehealth-actorid-qns::urn:poland:ncpb HTTP/1.1

Host: smp-digit-mock.publisher.ehealth.acc.edelivery.tech.ec.europa.eu

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:45.0) Gecko/20100101 Firefox/45.0 Accept: application/xml Accept-Language: en-GB,en;q=0.8,de;q=0.5,fr;q=0.3 Accept-Encoding: gzip, deflate DNT: 1 Referer: http://130.206.118.4/smp-swagger-ui/ Origin: http://130.206.118.4 Proxy-Authorization: Basic dXNlcm5hbWU6cGFzc3dvcmQ= Connection: keep-alive Text

N/A

Sample Response

```
HTTP header
HTTP/1.1 200 OK
Connection: Keep-Alive
Content-Encoding: gzip
Content-Length: 20
Content-Type: application/xml
Date: Thu, 22 Dec 2016 10:47:56 GMT
Server: Jetty(6.1.26)
Set-Cookie: BCIDSLB=PS1LUX-56; domain=europa.eu; path=/; HttpOnly
access-control-allow-origin:*
```

Text N/A

Table 3. UC05 Error codes

HTTP code	HTTP Message	Business code	Meaning
200	OK	n/a	The request was completed successfully
400	Bad Request	FORMAT_ERROR	Some field is expected to have a specific format is not valid (erroneous field name in description)
401	Unauthorized	UNAUTHORIZED	The user is not granted the right to issue this request
404	Resource not found	NOT_FOUND	The requested information was not found
500	Internal Server 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`

UC04 Create/Update Service Metadata

▼ *Use Case Description*

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. [REF7] §2.1) A sender (ed. "user") may want to discover what docu	

Brief description	
Actors	
Resource Admin	
Preconditions	
• The authenticated user has the role of "Resource Admin"	

Brief description	
• Resource Admin user initiating the request is linked to the specified Service Group	

Brief description	
- The certificate of the "Resource Admin" is valid	

Brief description	
- The certificate information of the "Resource Admin" was previously stored in the configuration	

Brief description	
Identifier and scheme of the service group and documents provided in the request must comply to the policy defined in [REF3]	
Basic flow event	
Step	
1	The receiver requests its service metadata to be put into SML.
2	The SMP verifies the certificate of the " Resource Admin" against its information in the database, validates the request, and either create or update all the information into its configuration database.
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**

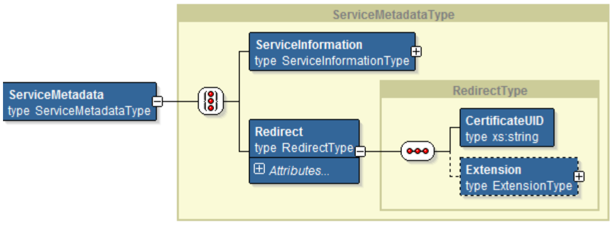
Brief description	
3a1	The receiver receives the confirmation that the definitions were updated properly with HTTP response "200 OK".
3a2	Use case ends with success.
Exception flows	
1a	**SMP is not reachable**
1a1	The user receives a network connection error.
1a2	Use case ends with success.
2a	**Authentication / 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 "400 Bad request" or "500 Internal server 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 defined0

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

▼ *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.

Insert or replace the all the ServiceInformation for that ServiceGroup Document.

In case of error:

- 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 if the authenticated user matches the “Resource Admin” user linked to the ServiceGroup.

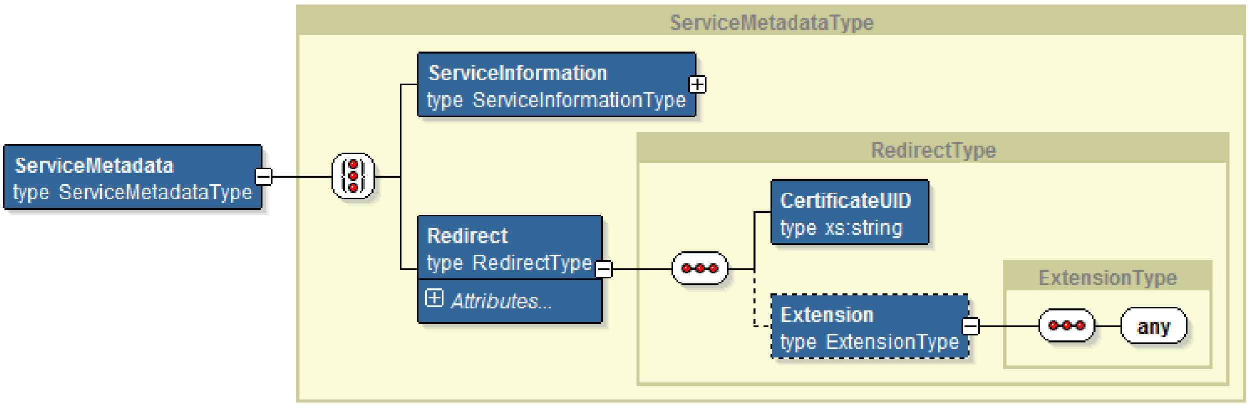
For this user to be the eligible “Resource Admin” it must have been referenced as such in the ServiceGroup definition (cf. PutServiceGroup) by an “Group Admin” user via service “PutServiceGroup” (by the “Group Admin” who was previously defined by the “Domain Admin”).

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 400 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 in the header.

HTTP Header (with certificate)

PUT http://smp-digit-mock.publisher.ehealth.acc.edelivery.tech.ec.europa.eu/ehealth-actorid-qns::urn:poland:ncpb/services/ehealth-resid-qns::urn::epsos##services:extended:epsos::107 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

Host: smp-digit-mock.publisher.ehealth.acc.edelivery.tech.ec.europa.eu

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:45.0) Gecko/20100101 Firefox/45.0

Accept: application/xml

Accept-Language: en-GB,en;q=0.8,de;q=0.5,fr;q=0.3

Accept-Encoding: gzip, deflate

DNT: 1

Content-Type: application/xml

Referer: http://130.206.118.4/smp-swagger-ui/

Content-Length: 4741

Origin: http://130.206.118.4

Proxy-Authorization: Basic dXNlcm5hbWU6cGFzc3dvcmQ=

Connection: keep-alive

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)

<?xml version="1.0" encoding="UTF-8" standalone="no"?><ServiceMetadata xmlns="http://docs.oasis-open.org/bdxr/ns/SMP/2016/05">

<ServiceInformation>

<ParticipantIdentifier scheme="ehealth-actorid-qns">urn:poland:ncpb</ParticipantIdentifier>

<DocumentIdentifier scheme="ehealth-resid-qns">urn::epsos##services:extended:epsos::107</DocumentIdentifier>

<ProcessList>

<Process>

<ProcessIdentifier scheme="ehealth-procid-qns">urn:epsosPatientService::List</ProcessIdentifier>

<ServiceEndpointList>

<Endpoint transportProfile="urn:ihe:iti:2013:xcpd">

&lt;EndpointURI&gt;<a href="http://poland.pl/ncp/patient/list&lt;/EndpointURI&gt" class="bare">http://poland.pl/ncp/patient/list&lt;/EndpointURI&gt</a>;

<RequireBusinessLevelSignature>false</RequireBusinessLevelSignature>

<MinimumAuthenticationLevel>urn:epSOS:loa:1</MinimumAuthenticationLevel>

<ServiceActivationDate>2016-06-06T11:06:02.000+02:00</ServiceActivationDate>

<ServiceExpirationDate>2026-06-06T11:06:02+02:00</ServiceExpirationDate>

<Certificate>MIID7jCCA1egAwIBAgICA+YwDQYJKoZIhvcNAQENBQAwOjELMAkGA1UEBhMCRlIx
EzARBgNVBAoMCklIRSBFdXJvcGUxFjAUBgNVBAMMDUlIRSBFdXJvcGUgQ0EwHhcNMTYwNjAxM
TQzNTUzWhcNMjYwNjAxMTQzNTUzWjCBgzELMAkGA1UEBhMCUFQxDDAKBgNVBAoMA01vSD
ENMAsGA1UECwwEU1BNUzENMAsGA1UEKgwESm9hbzEOMAwGA1UEBRMFQ3VuaGExHTAbBg
NVBAMMFHFhZXBzb3MubWluLXNhdWRlLnB0MRkwFwYDVQQMDBBTZXJ2aWNlIFByb3ZpZGV
yMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA1eN4qPSSRZqjVFG9TlcPlxf2WiSimQK
9L1nf9Z/s0ezeGQjCukDeDq/Wzqd9fpHhaMMq+XSSOtyEtIr5K/As4kFrViONUUkG12J6UllSWogp0N
YFwA4wIqKSFiTnQS5/nRTs05oONCCGILCyJNNeO53JzPlaq3/QbPLssuSAr6XucPE8wBBGM8b/TsB2
G/zjG8yuSTgGbhaZekq/Vnf9ftj1fr/vJDDAQgH6Yvzd88Z0DACJPHfW1p4F/OWLI386Bq7g/bo1DUPA
yEwlf+CkLgJWRKki3yJlOCIZ9enMA5O7rfeG3rXdgYGmWS7tNEgKXxgC+heiYvi7ZWd7M+/SUwIDA
QABo4IBMzCCAS8wPgYDVR0fBDcwNTAzoDGgL4YtaHR0cHM6Ly9nYXplbGxlLmloZS5uZXQvcGtp
L2NybC82NDMvY2FjcmwuY3JsMDwGCWCGSAGG+EIBBAQvFi1odHRwczovL2dhemVsbGUuaWhl
Lm5ldC9wa2kvY3JsLzY0My9jYWNybC5jcmwwPAYJYIZIAYb4QgEDBC8WLWh0dHBzOi8vZ2F6ZW
xsZS5paGUubmV0L3BraS9jcmwvNjQzL2NhY3JsLmNybDAfBgNVHSMEGDAWgBTsMw4TyCJeouF
rr0N7el3Sd3MdfjAdBgNVHQ4EFgQU1GQ/K1ykIwWFgiONzWJLQzufF/8wDAYDVR0TAQH/BAIwAD
AOBgNVHQ8BAf8EBAMCBSAwEwYDVR0lBAwwCgYIKwYBBQUHAwEwDQYJKoZIhvcNAQENBQA
DgYEAZ7t1Qkr9wz3q6+WcF6p/YX7Jr0CzVe7w58FvJFk2AsHeYkSlOyO5hxNpQbs1L1v6JrcqziNFrh
2QKGT2v6iPdWtdCT8HBLjmuvVWxxnfzYjdQ0J+kdKMAEV6EtWU78OqL60CCtUZKXE/NKJUq7TTU
CFP2fwiARy/t1dTD2NZo8c=</Certificate>

<ServiceDescription>This is the epSOS Patient Service List for the Polish NCP</ServiceDescription>

&lt;TechnicalContactUrl&gt;<a href="http://poland.pl/contact&lt;/TechnicalContactUrl&gt" class="bare">http://poland.pl/contact&lt;/TechnicalContactUrl&gt</a>;

&lt;TechnicalInformationUrl&gt;<a
href="http://poland.pl/contact&lt;/TechnicalInformationUrl&gt" class="bare">http://poland.pl/
contact&lt;/TechnicalInformationUrl&gt</a>;

</Endpoint>

</ServiceEndpointList>

</Process>

</ProcessList>

<Extension><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/2001/04/xmldsig-more#rsa-sha256"/><Reference
URI=""><Transforms><Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"/></Transforms><DigestMethod
Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/><DigestValue>CJeDJ72nQkwsZ2XWc8e
put8pcBzfHSwO6uHr77/xbQo=</DigestValue></Reference></SignedInfo><SignatureValue>WlCU
wlHJy9sehansEjFXSPkAobodbeM8OxXfLjQVYs7Vh085dESYaAbcDoDZ6t8IaHbsRtkiCgZG

yVRvOwB42EVRkhyWu0zVnlowfieBgvMqtZdYMbx6Z7Npwvo0UDcYI/HnHnzsyHhklKKNGPymXJ
XH

waEt4QJw+ne2n7Tb0Qg=</SignatureValue><KeyInfo><X509Data><X509SubjectName>CN=Sampl
e National
Infrastructure,OU=Sante,C=PT</X509SubjectName><X509Certificate>MIICAzCCAWygAwIBAgIEW
CRzHjANBgkqhkiG9w0BAQsFADBGMQswCQYDVQQGEwJQVDEOMAwGA1UE

CwwFU2FudGUxJzAlBgNVBAMMHlNhbXBsZSBOYXRpb25hbCBJbmZyYXN0cnVjdHVyZTAeFw0xN
jEx
MTAxMzE2NTBaFw0yNjExMTAxMzE2NTBaMEYxCzAJBgNVBAYTAlBUMQ4wDAYDVQQLDAVTY
W50ZTEn
MCUGA1UEAwweU2FtcGxlIE5hdGlvbmFsIEluZnJhc3RydWN0dXJlMIGfMA0GCSqGSIb3DQEBAQU
A
A4GNADCBiQKBgQCywt50WXEWIiWytRGcMqzeMM/EyxruNthPdiUEUTbs9un7lzGGjpfFMTgd83
wJ
haB6FgpaVd8V2w/JBdkim5Ltuhu2vA0d6hHOsa58neIfe4z1ZhswwNmB0+mDTjwnd/gg8IJyQhhY
c5G4x7m0ZGdDKZDizjtDTEPTsl8D4FzBFwIDAQABMA0GCSqGSIb3DQEBCwUAA4GBACKxUpAx0P
Ym
ZZi4DfAzBkQ0+CvQw/l6Yo8wonVdpcQXO3khpWIcXhgYhTLHwm8IwJLEyFatmMyCKklSA3CLebJU
L4XH1GcdCg6oPKPUc+ovbgN7/iR265Elp4qHfpVteBijBTyZReH4oAK9hRhK1gLwtjI7vpjVaPXv
vkV1fbrz</X509Certificate></X509Data></KeyInfo></Signature></Extension>

</ServiceInformation> </ServiceMetadata>

Sample Response (applicable for both examples request 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
```

NOTE 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

Table 4. UC06 Error codes

HTTP code	HTTP Message	Business code	Meaning
`200`	`OK`	n/a	The request was completed successfully.
`201`	`Created`	n/a	The PUT operation completed successfully.

HTTP code	HTTP Message	Business code	Meaning
400	Bad Request	XSD_INVALID	The XML included in the request is not validate against the XSD defining the input structure.
		MISSING_FIELD	Some field that is optional in the XSD but mandatory for this invocation is missing (missing field name in description).
		WRONG_FIELD	Some field is valid against XSD definition, but the more specific content is invalid (erroneous field name in description).
		OUT_OF_RANGE	Some numeric (or date field) is out of the valid range (erroneous field name in description).
		UNAUTHOR_FIELD	Some field that is optional in the XSD but forbidden for this invocation is present (unauthorized field name in description).
		FORMAT_ERROR	Some field is expected to have a specific format is not valid (erroneous field name in description).
		OTHER_ERROR	Some other specific error was encountered processing the request (more information in the ErrorDescription field).
401	Unauthorized	UNAUTHORIZED	The user is not granted the right to issue this request.
404	Resource not found	NOT_FOUND	The requested information was not found.
500	Internal Server 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
- DocumentIdentifier
- DocumentIdentifierScheme

- `IpAddress`
- `RequestHeader`
- `RequestText`
- `ResponseHeader`
- `HTTP code`
- `Business code`
- `ErrorDescription`

UC05 Erase Service Metadata

▼ *Use Case Description*

Brief description						
Remove all information about one specific service (i.e., all related processes and endpoints definitions).						
Actors						
Resource Admin (or Group Admin).						
Preconditions						
The user knows the address of SML.						

Brief description						
Resource Admin initiating the request is linked to the specified ServiceGroup.						
The authenticated user has the role of " Resource Admin”.						
The referenced ServiceMetadata exists.						
Basic flow event						
Step						
1	The receiver requests its service metadata to be removed from the SMP.					

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

Brief description						
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					

Brief description						
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)**					

Brief description						
2b1	The SMP replies with HTTP error "400 Bad request" or "500 Internal server 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					

Brief description						
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						
	Service Metadata are absent					

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

▼ *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 “Resource Admin” user linked to the Service Group.

For this user to be the eligible “Resource Admin” it must have been referenced as such in the ServiceGroup definition (cf. PutServiceGroup) by an “Group Admin” user 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 “Group Admin” user to issue the required operation (DeleteServiceGroup) if necessary.

Delete in one single transaction any information related to that service where participant and documents identifiers match the provided ServiceMetadata identifier.

In case of abort the deletion to undo what was previously done:

- Rollback the transaction
- 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

Table 5. ##Figure 10- SignedServiceMetadata data model

HTTP code	HTTP Message	Business code	Meaning
200	OK	n/a	The request was completed successfully
400	Bad Request	OTHER_ERROR	Some other specific error was encountered processing the request (more information in the ErrorDescription field)
401	Unauthorized	UNAUTHORIZED	The user is not granted the right to issue this request
404	Resource not found	NOT_FOUND	The requested information was not found
500	Internal Server 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
- DocumentIdentifier
- DocumentIdentifierScheme
- IpAddress

- RequestHeader
- ResponseHeader
- HTTP code
- Business code
- ErrorDescription

3.5. 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. §3.1.1 – "Original official OASIS SMP XSD”).

UC06 - Retrieve Service Group

▼ *Use Case Description*

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 participant Identifier+ participant Indentifier Scheme. Returns information from the ServiceMetadata table only (references to actual MetaData) (Cf. [REF7] §2.1). A sender (ed. "user") may want to discover what document types can						

Brief description						
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 references to SML					

Brief description						
2	The SMP validates the request and retrieves the information from its configuration database (into table Service Group and Service Metadata tables).					
3	The user receives the participant’s service group information					
4	Use case ends with success					
Exception flows						

Brief description						
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)**					

Brief description						
2a1	The SMP replies with HTTP error "400 Bad request" or "500 Internal server 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					

Brief description						
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						

Brief description						
	The user receives Service Group information for the requested receiver participant.					
Failure conditions						
	The user received no Service Group information about the requested receiver participant.					

▼ *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:

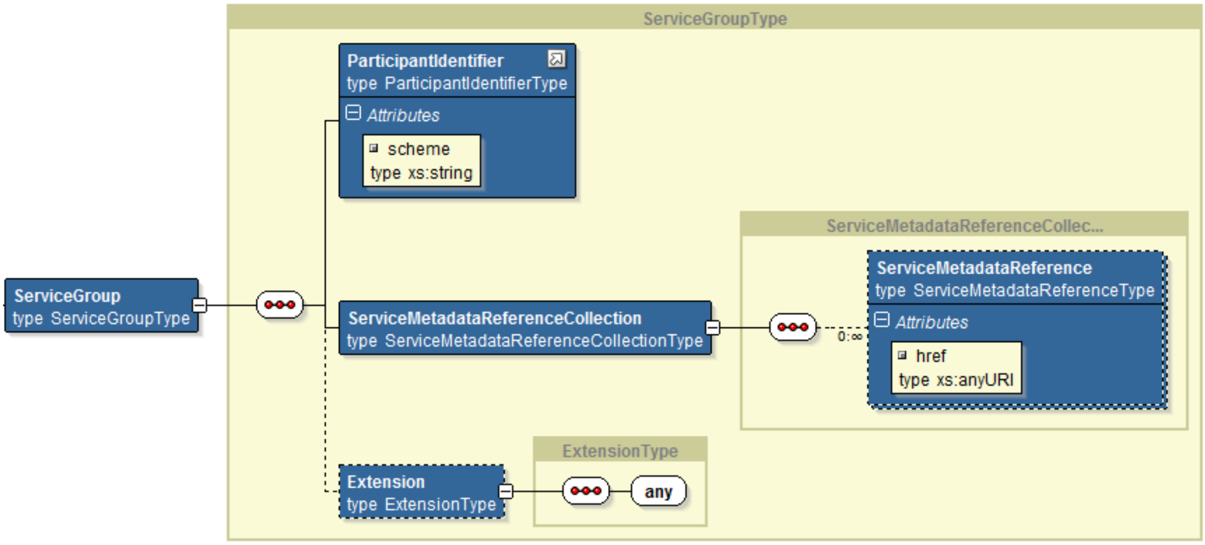
Selects all service Metadata related to the ServiceGroup specified by the provided

ParticipantIdentifier and build the corresponding URI from it.

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

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 metadata of the specified participant. To obtain the details on those services, the ServiceMetadata can be obtained from SML 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/2016/05 ">

<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

Table 6. ##Table 9 – UC07 Error codes

HTTP code	HTTP Message	Business code	Meaning
200	OK	n/a	The request was completed successfully
400	Bad Request	OTHER_ERROR	Some other specific error was encountered processing the request (more information in the ErrorDescription field)
404	Resource not found	NOT_FOUND	The requested information was not found

HTTP code	HTTP Message	Business code	Meaning
500	Internal Server 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

UC07 - Retrieve Service Metadata

▼ *Use Case Description*

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+documentIdentifi	

Brief description	
Actors	
User	
Preconditions	
The user application has previously resolved the address of SML from the DNS.	
Referenced service group and required Service Meta data were previously defined by the receiver.	

Brief description	
Basic flow event	
Step	
1	The user requests the detailed information of a receiver’s service to SML
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 reinitiates 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 "400 Bad request" or "500 Internal server 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

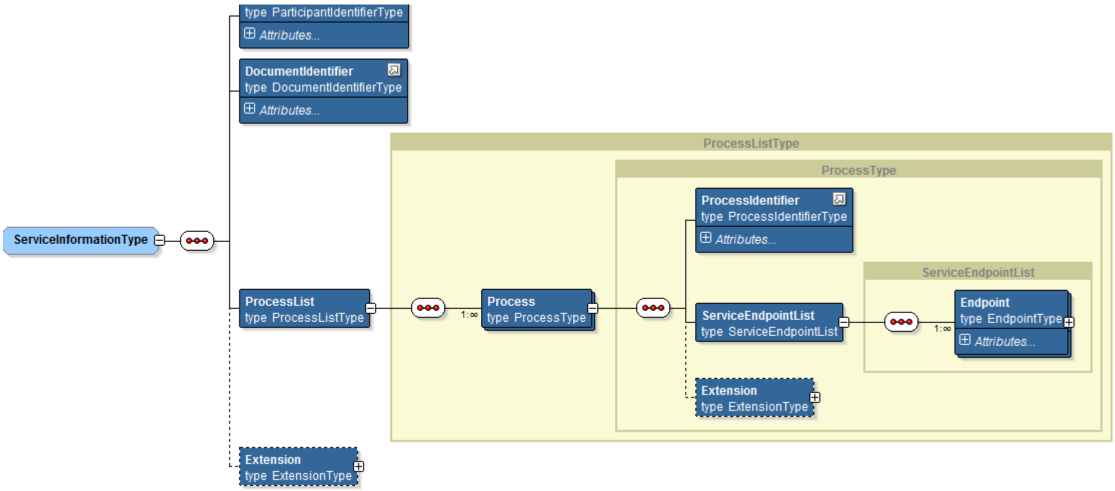
Brief description	
2a2a	**Multiple redirect**
2a2a1	The client receives redirect information for the 2nd time (and must ignore it)
2a2a2	Use case ends
Post conditions	
Successful conditions	
	The user receives ServiceMetaData information for the requested receiver participant.
Failure conditions	
	The user received no Metadata information about the requested receiver participant.

▼ *REST Service: GetSignedServiceMetadata*

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

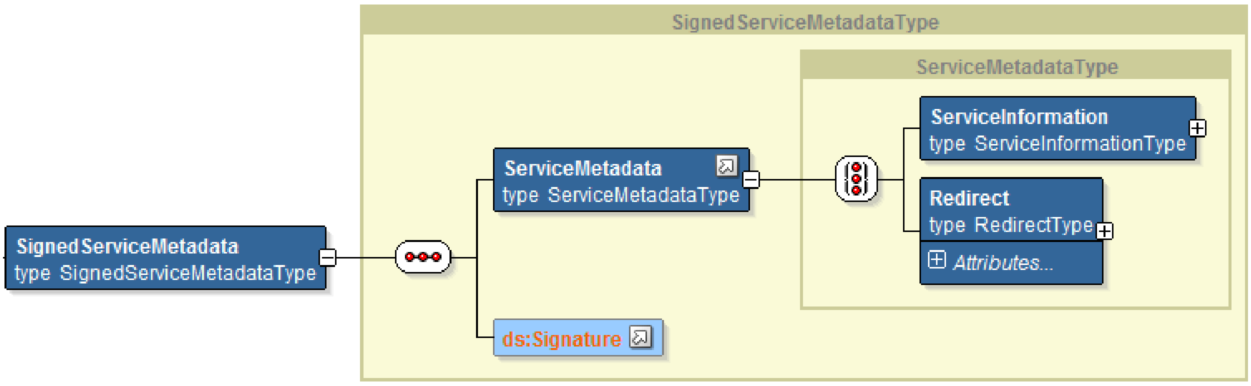


NB: there is no interaction with SML.

Output: *SignedServiceMetadata*

Cf. [REF7], §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 SML to allow the user (or any other user) verifying the authenticity of the information provided by SML by using the public key of SML before sending any document to the receiver.



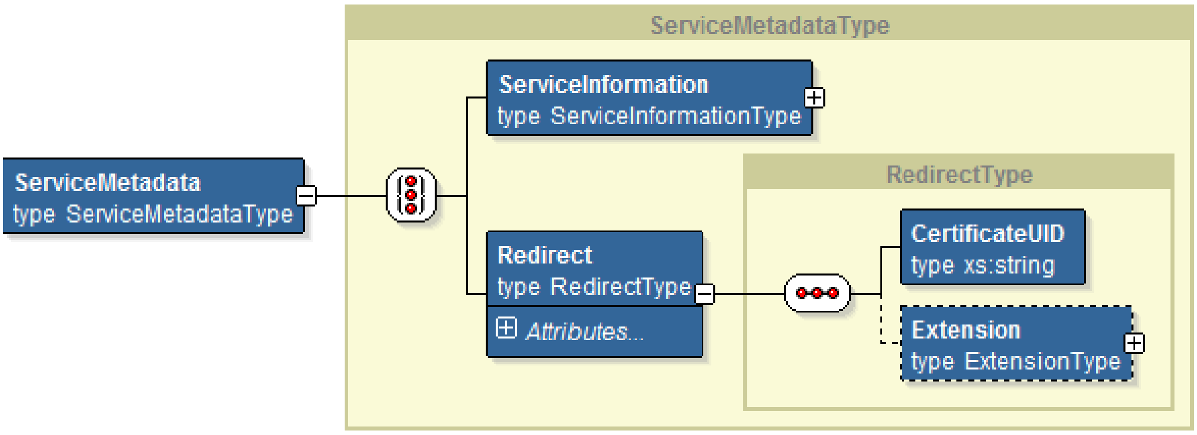
Output (alternative): Redirection (supports the alternative flow ‘a’ in the use case)

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

Redirection is exhaustively explained in [REF7] §4.3 ServiceMetadata and in [REF4] §2.1.3 Service Metadata Publisher Redirection.

In such a case, the information returned is the reference to SML 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/2016/05 ">

<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">

&lt;EndpointURI&gt;<a href="http://busdox.org/sampleService/&lt;/EndpointURI&gt" class="bare">http://busdox.org/sampleService/&lt;/EndpointURI&gt</a>;

<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">
&lt;EndpointURI&gt;<a href="http://busdox.org/sampleService/&lt;/EndpointURI&gt" class="bare">http://busdox.org/sampleService/&lt;/EndpointURI&gt</a>;
<RequireBusinessLevelSignature>true
</RequireBusinessLevelSignature>
<MinimumAuthenticationLevel>1</MinimumAuthenticationLevel>
<ServiceActivationDate>2009-05-01T09:00:00
</ServiceActivationDate>
<ServiceExpirationDate>2016-05-01T09:00:00
</ServiceExpirationDate>
<Certificate>AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA</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>
```

```xml
</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+mDbUsV8CrWzRVJvEJMnnyuI2bPMe6k4MJwp9A
4bTkzjvkMPARYAhyVNm6MNNlJRAFL4qddsRrWa4Jgf/QF0zQgpJ7ZUPdVQ8L8A54FiPZWltOIgZCf
O7sDbEcB00V4gKmzVPBsVu6BIBOws/UY=</SignatureValue>

<KeyInfo>

<X509Data>

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

<X509Certificate>MIICpTCCAg6gAwIBAgIBATANBgkqhkiG9w0BAQUFADB4MQswCQYDVQQGEwJ
CRTELMAkGA1UECAwCQkUxETAPBgNVBAcMCEJydXNzZWxzMQ4wDAYDVQQKDAVESUdJVDEL
MAkGA1UECwwCQjQxDzANBgNVBAMMBnJvb3RDTjEbMBkGCSqGSIb3DQEJARYMcm9vdEB0ZXN
0LmJlMB4XDTE1MDMxNzE2MTkwN1oXDTI1MDMxNDE2MTkwN1owfDELMAkGA1UEBhMCQkU
```

xCzAJBgNVBAgMAkJFMREwDwYDVQQHDAhCcnVzc2VsczEOMAwGA1UECgwFRElHSVQxCzAJBgN
VBAsMAkI0MREwDwYDVQQDDAhzZW5kZXJDTjEdMBsGCSqGSIb3DQEJARYOc2VuZGVyQHRlc3Q
uYmUwgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBANxLUPjIn7R0CsHf86kIwNzCu+6AdmW
M8fBLUHL+VXT6ayr1kwgGbFMb/vUUX6a46jRCiZBM+9IK1Hpjg9QX/QIQiWtvD+yDr6jUxahZ/w13
kqFG/K81IVu9DwLBoiNwDvQ6l6UbvMvV+1nWy3gjRcKlFs/C+E2uybgJxSM/sMkbAgMBAAGjOzA5
MB8GA1UdIwQYMBaAFHCVSh4WnWR8MGBGedr+bJH96tc4MAkGA1UdEwQCMAAwCwYDVR0PB
AQDAgTwMA0GCSqGSIb3DQEBBQUAA4GBAK6idNRxyeBmqPoSKxq7Ck3ej6R2QPyWbwZ+6/S7iC
Rt8PfgOu++Yu5YEjlUX1hlkbQKF/JuKTLqxNnKIE6Ef65+JP2ZaI9O2wdzpRclAhAd00XbNKpyipr4jM
dWmu2U8vyBBwn/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/2016/05 ">

<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+mDbUsV8CrWzRVJvEJMnnyuI2bPMe6k4MJwp9A
4bTkzjvkMPARYAhyVNm6MNNlJRAFL4qddsRrWa4Jgf/QF0zQgpJ7ZUPdVQ8L8A54FiPZWltOIgZCf
O7sDbEcB00V4gKmzVPBsVu6BIBOws/UY=</SignatureValue>

<KeyInfo>

<X509Data>

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

<X509Certificate>MIICpTCCAg6gAwIBAgIBATANBgkqhkiG9w0BAQUFADB4MQswCQYDVQQGEwJ
CRTELMAkGA1UECAwCQkUxETAPBgNVBAcMCEJydXNzZWxzMQ4wDAYDVQQKDAVESUdJVDEL
MAkGA1UECwwCQjQxDzANBgNVBAMMBnJvb3RDTjEbMBkGCSqGSIb3DQEJARYMcm9vdEB0ZXN
0LmJlMB4XDTE1MDMxNzE2MTkwN1oXDTI1MDMxNDE2MTkwN1owfDELMAkGA1UEBhMCQkU
xCzAJBgNVBAgMAkJFMREwDwYDVQQHDAhCcnVzc2VsczEOMAwGA1UECgwFRElHSVQxCzAJBg
NVBAsMAkI0MREwDwYDVQQDDAhzZW5kZXJDTjEdMBsGCSqGSIb3DQEJARYOc2VuZGVyQHRlc3
QuYmUwgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBANxLUPjIn7R0CsHf86kIwNzCu+6Adm
WM8fBLUHL+VXT6ayr1kwgGbFMb/vUUX6a46jRCiZBM+9IK1Hpjg9QX/QIQiWtvD+yDr6jUxahZ/w
```

13kqFG/K81IVu9DwLBoiNwDvQ6l6UbvMvV+1nWy3gjRcKlFs/C+E2uybgJxSM/sMkbAgMBAAGjOzA
5MB8GA1UdIwQYMBaAFHCVSh4WnWR8MGBGedr+bJH96tc4MAkGA1UdEwQCMAAwCwYDVR0P
BAQDAgTwMA0GCSqGSIb3DQEBBQUAA4GBAK6idNRxyeBmqPoSKxq7Ck3ej6R2QPyWbwZ+6/S7i
CRt8PfgOu++Yu5YEjlUX1hlkbQKF/JuKTLqxNnKIE6Ef65+JP2ZaI9O2wdzpRclAhAd00XbNKpyipr4j
MdWmu2U8vyBBwn/utG1ZrLhAUiqnPvmaQrResiGHM2xzCmVwtse</X509Certificate>

</X509Data>

</KeyInfo>

</Signature>

</SignedServiceMetadata>

Error codes

Table 7. ##Figure 13- Local administration model

HTTP code	HTTP Message	Business code	Meaning
200	OK	n/a	The request was completed successfully
400	Bad Request	OTHER_ERROR	Some other specific error was encountered processing the request (more information in the ErrorDescription field)
404	Resource not found	NOT_FOUND	The requested information was not found
500	Internal Server 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

3.6. Security

3.6.1. User management

Administration process

As described in §2.3.1 – “Actors”, there are 3 types of users accessing SML. Among them, only “Resource Admin” and “Group Admin” types of users are registered into the configuration of SML.

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

1. **Creation of a Group Admin**

   ▾ *Details*

   The "Domain Admin" sets existing users as an "Group Admin".

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

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

   ▾ *Details*

   This step is necessary for remote administration of ServiceGroups (if administration is local it is done by the "Group Admin" himself).

   The "System Admin”:

   - deploys the certificates that will be used to access SML for a new participant’s administration (if certificates are used);
   - creates manually the " Resource Admin" entry in the "Administrator" table

3. **Creation of the ServiceGroup (for one Participant)**

   ▾ *Details*

   The "Group Admin" accesses SML 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 "Group Admin" provides either:

   - A "*ServiceGroup-Owner*" in the HTTP header, i.e. some pieces of the Participant’s certificates that will be used to identify the " Resource Admin" user accessing SML for configuration purposes (mostly for distributed SMP model)
   - Nothing: in that case, the basic authentication information of the "Group Admin" (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 can to remove that Service Group via the same access method using "UC03 - Erase Service Group" (cf. §2.4.3).

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

4. **Creation of ServiceMetadata**

▼ *Details*

The "Resource Admin" accesses SML using its certificate.

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

Later he can remove deprecated services similarly with "UC05 - Erase Service Metadata" (cf. §2.4.5).

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

5. **Discovering a participant’s services capabilities**

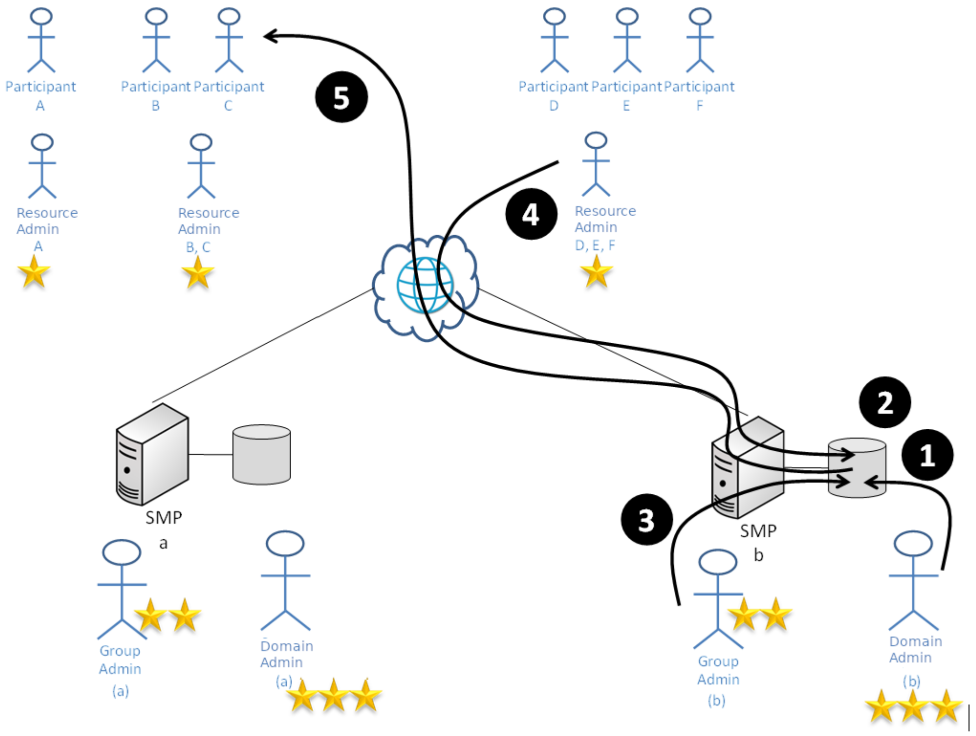
▼ *Details*

The Participant access SML 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.

The following diagram illustrates distributed (remote) "Resource Admin"'s:



The following diagram illustrates centralised ServiceGroup management (by the "Group Admin"):

[image43] | *image43.emf*

The specifications allow the coexistence of both models: some domain may decide to manage some ServiceGroups centrally (by the "Group Admin"), others in a distributed manner (by multiple remote "Resource Admin" 's).

▼ *Simple User*

The regular users (Actor “User”) are any user accessing the system public services. As these users do not need to be authenticated, they do not have to be known in advance by the System and are therefore not preregistered in any way on SML.

▼ *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 SML, this “system user” is able to modify the content of SML configuration database, i.e. he must have full read/write data access on this configuration database, in particular table "Administrator" described in §2.6.1.4 "Security tables".

He is responsible for creating and maintaining the definition of all “Group Admin” and "Resource Admin" administrators (as described by use case UC01).

▼ *Security tables*

Administrator

This table identifies the administrators of SML; this means `Group Admin` and Resource `Admin` actors introduced above.

Administrator
Identifier
Password

There are two possible means to obtain access to SML 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 administrator at logon
  - **Password** column contains then the hash of the password used to authenticate the user at logon
- through **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 SML to hash the password (and apply the same algorithm for authentication). The participant will send the password in 'clear' in the HTTP header.

Service Group Ownership

1-N relationship materialization between the service groups and the “Resource Admin” type of users of SML. More details are available under §2.6.1.6 – “Resource Resource Admin”.

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`.

▼ *Group Admin*

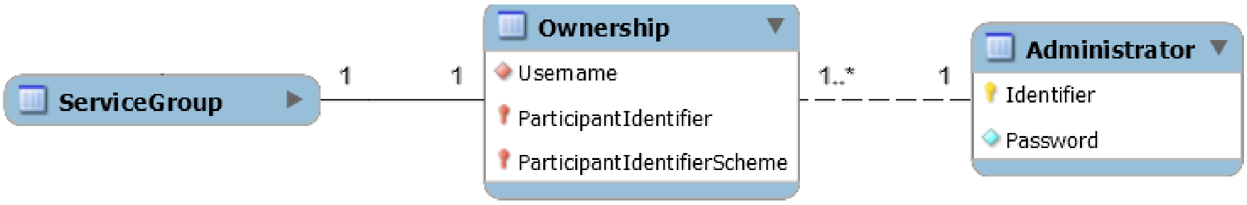
The “Group Admin” user is created by the system administrator (cf. §2.3.1 – “Actors” and §2.4.1 – "UC01 - Manage Administrators").

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

▼ *Resource Admin*

The “Resource Admin” 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 SignedServiceMetadata belonging to/referenced by a ServiceGroup.

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



The ServiceGroup can be managed by:

- The related "Resource Admin" (if any); and,
- The Group Admin (who may administer all service groups).

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

3.6.2. Access rights

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

		System Admin	Group Admin	Resource Admin	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

CAUTION "Group Admin" user may act on behalf of all the "Resource Admin" defined in

SML.

3.6.3. HTTP Authentication

SSL will be used at all times (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 SML 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 “Group Admin” users and optionally for “Resource Admin” users (cf. "Test at EC" above).
2. HTTP 2-way SSL for remote “Resource Admin” users (only) when and if this method is preferred for those to basic authentication (see "Authentication method" tables in §2.6.2 above: this authentication method might be used at EC in production environment).

If HTTP basic authentication is available for both types of users, 2-way SSL will also be usable for authenticating “Resource Admin” 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 (“Resource Admin” 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 *Administrator* table will be used as follows differently in the different possible cases (by user roles and authentication methods):

Table 8. ##Table 10 – Access rights summary

User role:		**Group Admin**	**Resource Admin**	
Authentication type:		**Basic Authentication**	**2 way-ssl**	**Basic Authentication**
Identifier:		Basic username	HTTP client cert	Basic username
Password:		password hash	n/a	password hash

<u>NB</u>: Only basic authentication is allowed for “Group Admin” 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.

3.6.4. Reverse proxy

This paragraph discusses the specific deployment in production and at the European Commission <u>for information only</u>.

An existing BDMSL server is already hosted <u>at the European Commission</u> behind a “Reverse Proxy” as explained in [REF10] §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.

[image46] | *image46.emf*

As stated above, this type of access will be provided for remote “Resource Admin” type of users only and is optional. Basic authentication will be used instead when there is no remote " Resource Admin"; i.e., when the "Group Admin" administers the ServiceGroup himself.

[image47] | *image47.emf*

Therefore, 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 “Administrator”.
  The “Client Certificate” values will be inserted in the HTTP header to SML 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 “Administrator” table column “username” should contain necessary information to validate that the provided value match.

- If no “Client certificate” information is available (meaning there is no reverse proxy between the client and SML), use Basic HTTP authentication: check provided username and password (clear value) to identify and authenticate the requesting user and authorize access.

To summarize, SML deployed at the European Commission has the following accesses:

1. Direct System & database logins are used by the System Admin.
2. Basic authentication over HTTP is used for the Group Admin and Resource’s that are on the same local network than SML itself.
   SMP authenticates local Group Admin’s based on the hash of the password that 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 SML for authorization (*Client-Cert* attribute) – password is blank
5. Parties do not have to authenticate themselves but may use SML’s certificate to authenticate it.

[image48] | *image48.emf*

3.6.5. Auditing

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

Table 9. ##Table 11 – Authentication types usage

		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**	**UC 01**	**UC 02**	**UC 03**	**UC 04**	**UC 05**	**UC 06**	**UC 07**
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 40x	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.

3.7. 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/DELETE services for 90% of the requests.

3.8. Annex

3.9. XSD files

3.9.1. OASIS SMP XSD

▼ *bdx-smp-201605.xsd*

```
<?xml version="1.0" encoding="UTF-8"?>
<!--
     Service Metadata Publishing (SMP) Version 1.0
     Committee Specification 03
     30 June 2016
     Copyright (c) OASIS Open 2016. All Rights Reserved.
     Source: http://docs.oasis-open.org/bdxr/bdx-smp/v1.0/cs03/schemas/
     Latest version of the specification: http://docs.oasis-open.org/bdxr/bdx-
smp/v1.0/bdx-smp-v1.0.html
     TC IPR Statement: https://www.oasis-open.org/committees/bdxr/ipr.php
     -->
<xs:schema
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    xmlns="http://docs.oasis-open.org/bdxr/ns/SMP/2016/05"
    xmlns:ds="http://www.w3.org/2000/09/xmldsig#" elementFormDefault="qualified"
targetNamespace="http://docs.oasis-open.org/bdxr/ns/SMP/2016/05"
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: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"
maxOccurs="unbounded"/>
        </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"
maxOccurs="unbounded"/>
        </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"
minOccurs="0" default="false"/>
            <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"
maxOccurs="unbounded"/>
        </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"
maxOccurs="unbounded"/>
        </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"
maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:attribute name="href" type="xs:anyURI" use="required"/>
    </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:complexType name="ExtensionType">
        <xs:annotation>
            <xs:documentation>
                A single extension for private use.
            </xs:documentation>
        </xs:annotation>
        <xs:sequence>
            <xs:element maxOccurs="1" minOccurs="0" name="ExtensionID"
type="xs:token">
                <xs:annotation>
                    <xs:documentation>
                        An identifier for the Extension assigned by the creator of
the extension.
                    </xs:documentation>
                </xs:annotation>
            </xs:element>
            <xs:element maxOccurs="1" minOccurs="0" name="ExtensionName"
type="xs:string">
                <xs:annotation>
                    <xs:documentation>
                        A name for the Extension assigned by the creator of the
extension.
                    </xs:documentation>
                </xs:annotation>
            </xs:element>
            <xs:element maxOccurs="1" minOccurs="0" name="ExtensionAgencyID"
type="xs:string">
                <xs:annotation>
                    <xs:documentation>
                        An agency that maintains one or more Extensions.
                    </xs:documentation>
                </xs:annotation>
            </xs:element>
            <xs:element maxOccurs="1" minOccurs="0" name="ExtensionAgencyName"
type="xs:string">
                <xs:annotation>
                    <xs:documentation>
                        The name of the agency that maintains the Extension.
                    </xs:documentation>
                </xs:annotation>
            </xs:element>
            <xs:element maxOccurs="1" minOccurs="0" name="ExtensionAgencyURI"
type="xs:anyURI">
                <xs:annotation>
                    <xs:documentation>
```

```
                        A URI for the Agency that maintains the Extension.
                    </xs:documentation>
                </xs:annotation>
            </xs:element>
            <xs:element maxOccurs="1" minOccurs="0" name="ExtensionVersionID"
type="xs:normalizedString">
                <xs:annotation>
                    <xs:documentation>
                        The version of the Extension.
                    </xs:documentation>
                </xs:annotation>
            </xs:element>
            <xs:element maxOccurs="1" minOccurs="0" name="ExtensionURI"
type="xs:anyURI">
                <xs:annotation>
                    <xs:documentation>
                        A URI for the Extension.
                    </xs:documentation>
                </xs:annotation>
            </xs:element>
            <xs:element maxOccurs="1" minOccurs="0" name="ExtensionReasonCode"
type="xs:token">
                <xs:annotation>
                    <xs:documentation>
                        A code for reason the Extension is being included.
                     </xs:documentation>
                </xs:annotation>
            </xs:element>
            <xs:element maxOccurs="1" minOccurs="0" name="ExtensionReason"
type="xs:string">
                <xs:annotation>
                    <xs:documentation>
                        A description of the reason for the Extension.
                    </xs:documentation>
                </xs:annotation>
            </xs:element>
            <xs:any namespace="##other" processContents="lax"/>
        </xs:sequence>
    </xs:complexType>
</xs:schema>
```

- source: http://docs.oasis-open.org/bdxr/bdx-smp/v1.0/cs03/schemas/bdx-smp-201605.xsd

3.9.2. Extended SMP XSD

ErrorResponse was defined as a response to return available detailed information on occurring error(s). You can find additional information for the values of elements BusinessCode and ErrorDescription in the Error Codes Table.

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

```
<xs:schema
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    xmlns="ec:services:SMP:1.0" targetNamespace="ec:services:SMP:1.0"
elementFormDefault="qualified" id="ServiceMetadataPublishing">
    <xs:element name="ErrorResponse" type="ErrorResponseType"/>
    <xs:complexType name="ErrorResponseType">
        <xs:sequence>
            <xs:element name="BusinessCode" type="xs:string"/>
            <xs:element name="ErrorDescription" type="xs:string" minOccurs="0"/>
            <xs:element name="ErrorUniqueId" type="xs:string"/>
        </xs:sequence>
    </xs:complexType>
</xs:schema>
```

3.9.3. Errors

Error Codes Table

The following table summarizes all possible errors returned by SML services:

				Applicable UC						
HTTP code	**HTTP Message**	**Business code**	**Meaning**	**UC 01**	**UC 02**	**UC 03**	**UC 04**	**UC 05**	**UC 06**	**UC 07**
				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	XSD_INVALID	The XML included in the request is not validate against the XSD defining the input structure.	-	X		X		-	-
400	Bad Request	MISSING_FIELD	Some field that is optional in the XSD but mandatory for this invocation is missing (missing field name in description).	-	X		X		-	-

				Applicable UC						
400	Bad Request	WRONG_FIELD	Some field is valid against XSD definition, but the more specific content is invalid (erroneous field name in description) Or Some header field is either missing or invalid.	-	X		X		-	-
400	Bad Request	OUT_OF_RANGE	Some numeric (or date field) is out of the valid range (erroneous field name in description).	-			X		-	-
400	Bad Request	UNAUTHOR_FIELD	Some field that is optional in the XSD but forbidden for this invocation is present (unauthorized field name in description).	-			X		-	-
400	Bad Request	FORMAT_ERROR	Some field is expected to have a specific format is not valid (erroneous field name in description).		X	X	X		-	-
400	Bad Request	USER_NOT_FOUND	The referenced " Resource Admin" was not found as Administrator.		X				-	-
400	Bad Request	OTHER_ERROR	Some other specific error was encountered processing the request (more information in the *ErrorDescription* field).				(x)	(x)	(x)	(x)
401	Unauthorized	UNAUTHORIZED	The user is not granted the right to issue this request.	-	X	X	X	X	-	-
404	Resource not found	NOT_FOUND	The requested information was not found.	-		X		X	X	X

				Applicable UC						
500	Internal Server Error	TECHNICAL	Some unexpected technical error occurred (detailed information is available in the response).	-	X	X	X	X	X	X

Legend

- X = This service returns this kind of errors
- (x) = This service might return this kind of errors, and in the event might provide more unstructured information in the *errorDescription* field of the *ErrorResponse* structure.

Detailed Errors structure

In case of error, a response text will be provided, in an “ErrorResponse” type of element (cf. definition in 3.1.2 – “Extended SMP XSD ”).

The *ErrorResponse* holds the following elements:

- *BusinessCode*

> This code allows the client application to behave appropriately according to the encountered error. The expected values are summarized in §§3.2 –"Errors codes table" and their applicability explicitly specified for each service in the corresponding paragraph.

- *ErrorDescription*

> This description provides some detailed information on the encountered error. Its content is not predefined and should be intended to help the client developer or administrator to investigate the encountered error.

- *ErrorUniqueId*

> This identifier uniquely identifies the occurrence of the error. This value is intended to facilitate further investigations on a specific error in particular to search into log files.

Example:

```xml
<ErrorResponse xmlns="ec:services:SMP:1.0">
    <BusinessCode>TECHNICAL</BusinessCode>
    <ErrorDescription>Some unexpected technical error occurred. (detailed
    information available here)</ErrorDescription>
    <ErrorUniqueId>5378C627DA4275F698458AB6845C68456845</ErrorUniqueId>
</ErrorResponse>
```

Chapter 4. Reference Guides

4.1. SMP Properties Reference

▼ **SMP Configuration Properties**

Configuration Property	Description and Usage	Default
smp.configuration.file	*Configuration property file path.*	smp.config.properties
smp.init.configuration.file	*Init configuration property file path.*	smp.init.properties
smp.security.folder	*Security folder for storing the keystore and the truststore.*	smp
smp.jdbc.driver	*Database Configuration - Driver* • MySQL: com.mysql.jdbc.Driver • Oracle: oracle.jdbc.OracleDriver	com.mysql.jdbc.Driver
smp.jdbc.url	*Database Configuration - URL* • MySQL: jdbc:mysql://dbhost:dbport/smp_database • Oracle: ◦ jdbc:oracle:thin:@dbhost:dbport:smp_database ◦ jdbc:oracle:thin:@dbhost:dbport/smp_service	jdbc:mysql://localhost:3306/smp
smp.jdbc.user	*Database User/Password Configuration - User*	smp
smp.jdbc.password	*Database User/password Configuration - Password*	The_password
smp.datasource.jndi	*If the data source is configured on the application server (*recommended), the property defines the JNDI name of the database connection.*	jdbc/eDeliverySmpDs
smp.database.show-sql	*Print generated sql queries to logs. The property is effective only when* smp.mode.development=true.	false

Configuration Property	Description and Usage	Default
`smp.database.create-ddl`	*Auto create/update database objects. The property is effective only when* `smp.mode.development=true`.	`false`
`smp.log.folder`	**IMPORTANT** Do NOT this feature in production, it is only intended for tests, demonstrations and development purposes. *The provided logback.xml configuration defines logging file as* `<file>${log.folder:-logs}/edelivery-smp.log</file>` *With the property we can define the folder for the logging files.*	`/var/logs/smp`
`smp.log.configuration.file`	*Custom logback configuration file (filepath can be absolute or relative to smp configuration.dir).*	`/opt/logging/smp-logback.xml`
`smp.libraries.folder`	*Path where SMP extensions are located. The folder is loaded by the SMP classloader at startup.*	`/opt/smp/extension-libs`
`smp.smp.mode.development`	*The development mode uses semi-random generators for password and key generation. Setting the property value to ‘true‘ makes the first startup and access token generation faster. To ensure high security, this option MUST NOT be enabled in production.*	`false`

Chapter 5. Support

eDelivery Support Team maintains and supports the DomiSMP Documentation. For any questions, comments or requests for change, please contact:

- **Email**: ec-edelivery-support@ec.europa.eu
- **Hours**: 8AM to 6PM (Normal EC working days)