EUROPEAN COMMISSION

DIGIT
Connecting Europe Facility

SML

Interface Control Document

Business Document Metadata Service Location (BDMSL)

Version [1.06]

Status [Final]

Date: 17/04/2018

Document Approver(s):

Approver Name	Role
Joao RODRIGUES	CEF eDelivery

Document Reviewers:

Reviewer Name	Role
Adrien FERIAL	Product Owner
Yves ADAM	Business Analyst

Summary of Changes:

Version	Date	Created by	Short Description of Changes
0.01	20/07/2016	Yves ADAM	Initial version
0.02	25/07/2016	Yves ADAM	First delivery for review
0.03	25/08/2016	Yves ADAM	Integrate comments of Flavio SANTOS
0.04	07/09/2016	Yves ADAM	Integrate comments of Pedro TAVARES
0.05	13/09/2016	Yves ADAM	Integrate comments of Adrien FERIAL
1.00	15/09/2016	Yves ADAM	Integrate comments of Flavio SANTOS
1.01	06/03/2017	Flavio SANTOS	Updating *MigrationKey* constraint
1.02	07/03/2017	Tiago MIGUEL	Changes related to Is Alive
1.03	25/04/2017	Flavio SANTOS	Update license to EUPL 1.1
1.04	19/07/2017	Flavio SANTOS	Update participant id error message
1.05	16/01/2018	Flavio SANTOS	Update license to EUPL 1.2
1.06	27/03/2018	CEF Support	Reuse policy notice added. E-SENS BDXL profile replaced by eDelivery BDXL profile.

Table of Contents

1. INTRODUCTION

BDMSL stands for Business Document Metadata Service Location. BDMSL is the sample implementation of the SML maintained by DG DIGIT. The version of the BDMSL refered in this document is 3.0.1. This version implements the eDelivery BDXL profile.

1.1. Purpose of the Interface Control Document

This document will univocally define the participant's interface to the BDMSL. This document describes the WSDL and the observable behaviour of the interface.

There are 3 interfaces described in this document:

Interface	Description	Version
ManageServiceMetadataService-1.0 .wsdl	Definition of the service to Manage Service Metadata.	1.0
ManageBusinessIdentifierService-1.0 .wsdl	Definition of the service to Manage Participant Identifier's.	1.0
BDMSLService-1.0.wsdl	Definition of the additional services specific to this implementation	1.0

Table 1 - Documented services

1.2. Scope of the document

This document covers the service interface of the BDMSL. It includes information regarding the description of the services available, the list of use cases, the information model and the sequence of message exchanges for the services provided. This specification is limited to the service interface of the BDMSL. All other aspects of its implementation are not covered by this document. The ICD specification provides both the provider (i.e. the implementer) of the services and their consumers with a complete specification of the following aspects:

- *Interface Functional Specification*, this specifies the set of services and the operations provided by each service and this is represented by the flows explained in the use cases.
- *Interface Behavioural Specification,* this specifies the expected sequence of steps to be respected by the participants in the implementation when calling a service or a set of services and this is represented by the sequence diagrams presented in the use cases.
- *Interface Message standards,* this specifies the syntax and semantics of the data

1.3. Audience

This document is intended to:

- The Directorate Generals and Services of the European Commission, Member States (MS) and also companies of the private sector wanting access BDMSL services. In particular:
  - **Architects** will find it useful for determining how to best exploit BDMSL services to create a fully-fledged solution integrating other components like the SMP, the DNS and the administration application;
  - **Analysts** will find it useful to understand the communication between the BDMSL and its major peer component the SMP which will enable them to have an holistic and detailed view of the operations and data involved in the use cases;
  - **Developers** will find it essential as a basis of their development concerning the interaction mainly between the BDMSL and the SMP, and also with the DNS and the administration application;
  - **Testers** can use this document in order to test the interface by following the use cases described; in particular the communications of the BDMSL with the SMP.

1.4. References

The table below provides the reader with the list of reference documents.

#	Document	Contents outline
[REF1]	Business Document Metadata Service Location Version 1.0	This specification defines service discovery methods. A method is first specified to query and retrieve an URL for metadata services. Two metadata service types are then defined. Also an auxiliary method pattern for discovering a registration service to enable access to metadata services is described. The methods defined here are instances of the generic pattern defined within IETF RFCs for Dynamic Delegation Discovery Services (DDDS). This specification then defines DDDS applications for metadata and metadata-registration services.
[REF2]	PEPPOL	The OpenPEPPOL Association is responsible for the governance and maintenance of the PEPPOL specifications that enable European businesses to easily deal electronically with any European public sector buyer in their procurement processes.

#	Document	Contents outline
[REF3]	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. The Service Metadata Locator service exposes three interfaces: Service Metadata discovery, Manage participant identifiers and Manage service metadata interfaces.
[REF4]	Service Metadata Publishing (SMP) Version 1.0	This document describes a protocol for publishing service metadata within a 4-corner network. In a 4-corner network, entities are exchanging business documents through intermediary gateway services (sometimes called Access Points). To successfully send a business document in a 4-corner network, an entity must be able to discover critical metadata about the recipient (endpoint) of the business document, such as types of documents the endpoint is capable of receiving and methods of transport supported. The recipient makes this metadata available to other entities in the network through a Service Metadata Publisher service. This specification describes the request/response exchanges between a Service Metadata Publisher and a client wishing to discover endpoint information. A client can either be an end-user business application or a gateway/access point in the 4-corner network. It also defines the request processing that must happen at the client side.
[REF5]	eDelivery BDXL profile	Specifications of eDelivery BDXLprofile.
[REF6]	Policy for use of Identifiers (PEPPOL Transport Infrastructure)	This document describes a PEPPOL policy and guidelines for use of identifiers within the PEPPOL network.

2. INTRODUCTION

2.1. Glossary

The key terms used in this document are defined in the CEF Definitions section on the CEF Digital Single Web Portal:

https://ec.europa.eu/cefdigital/wiki/display/CEFDIGITAL/CEF+Definitions

The key acronyms used in this Interface Control Document are defined in the CEF Glossary on the CEF Digital Single Web Portal:

https://ec.europa.eu/cefdigital/wiki/pages/viewpage.action?spaceKey=CEFDIGITAL&title=CEF+Glossary

3. INTERFACE FUNCTIONAL SPECIFICATION

3.1. Purpose of the BDMSL component

The BDMSL component enables Access Points to dynamically discover the IP address of the destination Access Point. Instead of looking at a static list of IP addresses, the Access Point consults a Service Metadata Publisher (SMP) where information about every participant in the document/data exchange network is kept up to date, including the IP addresses of their Access Point. As at any point in time there can be one or several active SMPs in a same network. For dynamic discovery to work, every participant must be given a unique ID in the form of a website's URL which must be known on the internet's Domain Name System (DNS) thanks to the Service Metadata Locator (SML). By knowing this URL, the Access Point is able to dynamically locate the right SMP and therefore the right Access Point.

By combining the SMP services with a Service Location solution building block like the BDMSL component, the participants of a document/data exchange network can benefit from **dynamic discovery**. In such a configuration, a participant about to send a document or data will first use the service location service (e.g. the BDMSL) to retrieve the endpoint address of the SMP. As a second step, he will query the SMP to obtain "***Metadata***"of the receiving participat; i.e. its capabilities and settings, which includes the endpoint address of the receiver's access point. The sender has then enough information and can send the message to the receiver using the adequate transport protocol, security settings, etc.

The CEF eDelivery Service Metadata Locator (SML) enables Access Points to dynamically discover the IP address of the destination Access Point. Instead of looking at a static list of IP addresses, the Access Point consults a Service Metadata Publisher (SMP) where information about every participant in the document/ data exchange network is kept up to date, including the IP addresses of their Access Point. As at any point in time there can be one or several active SMPs in a same network. For dynamic discovery to work, every participant must be given a unique ID in the form of a website's URL which must be known on the internet's Domain Name System (DNS) thanks to the Service Metadata Locator (SML). By knowing this URL, the Access Point is able to dynamically locate the right SMP and therefore the right Access Point.

The current SML software component maintained by the European Commission implements the PEPPOL Transport Infrastructure SML specifications.

3.2. Use cases overview

3.2.1. Actors

Actor	Definition
SMP	Holds the service metadata information about participants in the network

Actor	Definition
SML	Provides controlled access to the creation and updating of entries in the DNS
ADMIN	Application user with administrative privileges
Directory user	The operator of a global Directory service who is allowed to call the listParticipants service

Table 2 - Actors

3.2.2. *Use cases diagram*

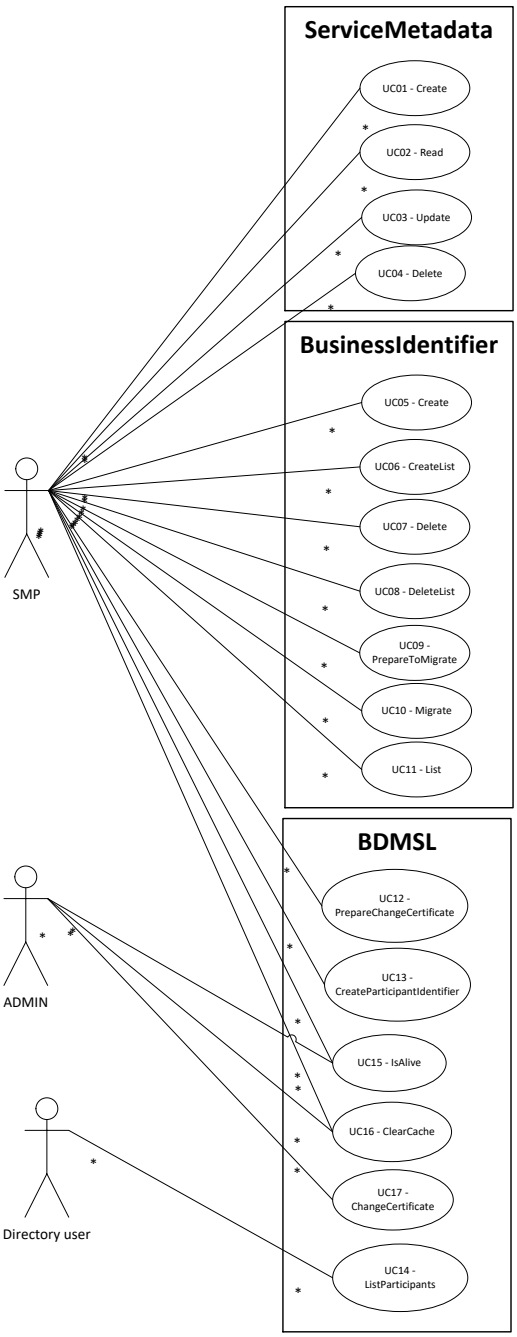


Figure 1 - Use case diagrams

ID	Operation	Actor	Short description
Interface : ManageServiceMetadataService-1.0 .wsdl			
UC1	Create	SMP	Establishes a Service Metadata Publisher metadata record, containing the metadata about the Service Metadata Publisher- information as outlined in the *ServiceMetadataPublisherService* data type.
UC2	**Read**	SMP	Retrieves the Service Metadata Publisher record for the service metadata publisher.
UC3	**Update**	SMP	Updates the Service Metadata Publisher record for the service metadata publisher.
UC4	**Delete**	SMP	Deletes the Service Metadata Publisher record for the service metadata publisher.
Interface : ManageBusinessIdentifierService-1.0.wsdl			
UC5	**Create**	SMP	Creates an entry in the Service Metadata Locator service for information relating to a specific participant identifier. Regardless of the number of services a recipient exposes, only one record corresponding to the participant identifier is created in the Service Metadata Locator service by the Service Metadata Publisher which exposes the services for that participant.
UC6	**CreateList**	SMP	Creates a set of entries in the Service Metadata Locator service for information relating to a list of participant identifiers. Regardless of the number of services a recipient exposes, only one record corresponding to each participant identifier is created in the Service Metadata Locator service by the Service Metadata Publisher which exposes the services for that participant.
UC7	**Delete**	SMP	Deletes the information that the Service Metadata Locator service holds for a specific Participant Identifier.

ID	Operation	Actor	Short description
UC8	**DeleteList**	SMP	Deletes the information that the Service Metadata Locator service holds for a list of Participant Identifiers.
UC9	**PrepareToMigrate**	SMP	Prepares a Participant Identifier for migration to a new Service Metadata Publisher. This operation is called by the Service Metadata Publisher which currently publishes the metadata for the Participant Identifier. The Service Metadata Publisher supplies a Migration Code which is used to control the migration process. The Migration Code must be passed (out of band) to the Service Metadata Publisher which is taking over the publishing of the metadata for the Participant Identifier and which MUST be used on the invocation of the Migrate() operation. This operation can only be invoked by the Service Metadata Publisher which currently publishes the metadata for the specified Participant Identifier.
UC10	**Migrate**	SMP	Migrates a Participant Identifier already held by the Service Metadata Locator service to target a new Service Metadata Publisher. This operation is called by the Service Metadata Publisher which is taking over the publishing for the Participant Identifier. The operation requires the new Service Metadata Publisher to provide a migration code which was originally obtained from the replaced Service Metadata Publisher. The PrepareToMigrate operation MUST have been previously invoked for the supplied Participant Identifier, using the same *MigrationCode*, otherwise the Migrate() operation fails. Following the successful invocation of this operation, the lookup of the metadata for the service endpoints relating to a particular Participant Identifier will resolve (via DNS) to the new Service Metadata Publisher.

ID	Operation	Actor	Short description
UC11	**List**	SMP	Used to retrieve a list of all participant identifiers associated with a single Service Metadata Publisher for synchronization purposes. Since this list may be large, it is returned as pages of data, with each page being linked from the previous page.
Interface : BDMSLService-1.0.wsdl			
UC12	**PrepareChangeCertificate**	SMP	Allows a SMP to prepare a change of its certificate when the current one is about to expire, and the future one is available.
UC13	**CreateParticipantIdentifier**	SMP	This service has the same behaviour as the *Create*() operation in the *ManageParticipantIdentifier* (UC5) interface but it has one additional and optional() Input: the *serviceName* element.
UC14	**IsAlive**	SMP/ ADMIN	Confirms that the application is up and running (monitoring purposes)
UC15	**ClearCache**	SMP/ ADMIN	Clears all the caches managed by the application.
UC16	**ChangeCertificate**	ADMIN	This operation allows the admin team to change the SMP's certificate. It is called by the admin team in case the SMP's certificate has expired and the new one needs to be applied.
UC17	**ListParticipants**	Directory User	Lists all the participants managed by the BDMSL. This service is only meant to be called by the Directory Userapplication.

Table 3 – Use cases

3.3. Sample Requests and Responses

In addition to the detailed use cases provided here below, sample requests and responses can be found in the SoapUI project referenced here:

https://ec.europa.eu/cefdigital/code/projects/EDELIVERY/repos/bdmsl/browse/bdmsl-webapp/src/test/resources/SML-soapui-project-TEST.xml

These examples come in addition to the structure definition of the interface (cf. §4.2 - WSDL – Data model) based on their WSDL files and related XSDs.

3.4. Detailed use cases - ManageServiceMetadataService

The following paragraphs define the use cases related to the ManageServiceMetadata service.

The ManageServiceMetadata interface allows Service Metadata Publishers to manage the metadata held in the Service Metadata Locator service about their service metadata publisher services, e.g. binding, interface profile and key information.

This interface requires authentication of the user. The identity of the user derived from the authentication process identifies the Service Metadata Publisher associated with the service metadata which is managed via this interface.

3.4.1. UC1 Create

Description UC01 Create

Establishes a Service Metadata Publisher metadata record, containing the metadata about the Service Metadata Publisher (SMP), as outlined in the ServiceMetadataPublisherService data type.

Actors

SMP

Preconditions

C1	The user has a valid certificate
C2	The role associated to the user is ROLE_SMP
C3	The SMP does not already exist in the SML
C4	Input: CreateServiceMetadataPublisherService: ServiceMetadataPublisherService - contains the service metadata publisher information, which includes the logical and physical addresses for the SMP (Domain name and IP address). It is assumed that the ServiceMetadataPublisherID has been assigned to the calling user out-of-bands.

Basic Flow

Actor	Step	
SMP	1	Invokes the Create() operation
SML	2	Authenticates the user, validates the request, and adds the metadata record into its configuration database.
SML	3	Returns a positive response to the requester
SMP	4	Receives the creation confirmation
	5	Use case ends

Alternative flows

None.

Exception flows

E1	**The user is not authorized**	
2.1.1	SML	Returns an HTTP error 401 as response to the requester
2.1.2	SMP	Receives the error response
2.1.3		Use case ends
E2	**Request is not valid**	
2.2.1	SML	Returns an HTTP error 400 as response to the requester
2.2.2	SMP	Receives the error response
2.2.3		Use case ends
E3	**Any other error occurred that prevented the SML to process the request**	
2.3.1	SML	Returns an HTTP error 500 as response to the requester
2.3.2	SMP	Receives the error response
2.3.3		Use case ends

Post conditions

Successful conditions

The Metadata record has been created into the SML

Failure Conditions (HTTP errors)

unauthorizedFault (401) - returned if the caller is not authorized to invoke this operation (E1)

badRequestFault (400) - returned if the supplied CreateServiceMetadataPublisherService does not contain consistent data (E2)

internalErrorFault (500) - returned if the SML service is unable to process the request for any reason (E3)

3.4.2. UC2 Read

Description UC02 Read

Retrieves the Service Metadata Publisher record for the service metadata publisher.

Actors

SMP

Preconditions

C1	The user has a valid certificate
C2	The role associated to the user is ROLE_SMP
C3	The SMP already exists in the SML
C4	Input ReadServiceMetadataPublisherService: ServiceMetadataPublisherID - the unique ID of the Service Metadata Publisher for which the record is required

Basic Flow

Actor	Step	
SMP	1	Invokes the Read() operation
SML	2	Authenticates the user, validates the request, and reads the requested SMP metadata from its configuration database.
SML	3	Returns a positive response to the requester
SMP	4	Receives the metadata
	5	Use case ends

Alternative flows

None.

Exception flows

E1	**The user is not authorized**	
2.1.1	SML	Returns an HTTP error 401 as response to the requester
2.1.2	SMP	Receives the error response
2.1.3		Use case ends
E2	**Request is not valid**	
2.2.1	SML	Returns an HTTP error 400 as response to the requester
2.2.2	SMP	Receives the error response
2.2.3		Use case ends
E3	**Any other error occurred that prevented the SML to process the request**	
2.3.1	SML	Returns an HTTP error 500 as response to the requester
2.3.2	SMP	Receives the error response
2.3.3		Use case ends
E4	**The identifier of the SMP could not be found**	
2.4.1	SML	Returns an HTTP error 404 as response to the requester
2.4.2	SMP	Receives the error response
2.4.3		Use case ends

Post conditions

Successful conditions

The Metadata has been provided to the requester.
Output: ServiceMetadataPublisherService - the service metadata publisher record, in the form of a ServiceMetadataPublisherService data type

Failure Conditions (HTTP errors)

unauthorizedFault (401) - returned if the caller is not authorized to invoke the Create operation (E1)

badRequestFault (400) - returned if the supplied parameter does not contain consistent data (E2)

internalErrorFault (500) - returned if the SML service is unable to process the request for any reason (E3)

notFoundFault (404) - returned if the identifier of the SMP could not be found (E4)

3.4.3. UC3 Update

Description UC03 Update

Updates the Service Metadata Publisher record for the service metadata publisher.

Actors

SMP

Preconditions

C1	The user has a valid certificate
C2	The role associated to the user is ROLE_SMP
C3	The SMP already exists in the SML
C4	Input UpdateServiceMetadataPublisheServicer: ServiceMetadataPublisherService - contains the service metadata for the service metadata publisher, which includes the logical and physical addresses for the SMP (Domain name and IP address)

Basic Flow

Actor	Step	
SMP	1	Invokes the Update() operation
SML	2	Authenticates the user, validates the request, and updates the requested metadata record from its configuration database.
SML	3	Returns a positive response to the requester
SMP	4	Receives the update confirmation
	5	Use case ends

Alternative flows

None.

Exception flows

E1	**The user is not authorized**	
2.1.1	SML	Returns an HTTP error 401 as response to the requester
2.1.2	SMP	Receives the error response
2.1.3		Use case ends
E2	**Request is not valid**	
2.2.1	SML	Returns an HTTP error 400 as response to the requester
2.2.2	SMP	Receives the error response
2.2.3		Use case ends
E3	**The identifier of the SMP could not be found**	
2.3.1	SML	Returns an HTTP error 404 as response to the requester
2.3.2	SMP	Receives the error response
2.2.3		Use case ends
E4	**Any other error occurred that prevented the SML to process the request**	
2.4.1	SML	Returns an HTTP error 500 as response to the requester
2.4.2	SMP	Receives the error response

2.4.3 Use case ends

Post conditions

Successful conditions

The SMP record has been created into the SML

Failure Conditions (HTTP errors)

unauthorizedFault (401) - returned if the caller is not authorized to invoke this operation (E1)

badRequestFault (400) - returned if the supplied UpdateServiceMetadataPublisheServicer does not contain consistent data (E2)

notFoundFault (404) - returned if the identifier of the SMP could not be found (E3)

internalErrorFault (500) - returned if the SML service is unable to process the request for any reason (E4)

3.4.4. UC4 Delete

Description UC04 Delete

Deletes the Service Metadata Publisher record for the service metadata publisher.

Actors

SMP

Preconditions

C1	The user has a valid certificate
C2	The role associated to the user is ROLE_SMP
C3	The SMP already exists in the SML
C4	Input DeleteServiceMetadataPublisherService: ServiceMetadataPublisherID - the unique ID of the Service Metadata Publisher to delete

Basic Flow

Actor	Step	
SMP	1	Invokes the Delete() operation
SML	2	Authenticates the user, validates the request, and deletes the requested metadata record from its configuration database.
SML	3	Returns a positive response to the requester
SMP	4	Receives the deletion confirmation
	5	Use case ends

Alternative flows

None.

Exception flows

E1	**The user is not authorized**	
2.1.1	SML	Returns an HTTP error 401 as response to the requester
2.1.2	SMP	Receives the error response
2.1.3		Use case ends
E2	**Request is not valid**	
2.2.1	SML	Returns an HTTP error 400 as response to the requester
2.2.2	SMP	Receives the error response
2.2.3		Use case ends
E3	**The identifier of the SMP could not be found**	
2.3.1	SML	Returns an HTTP error 404 as response to the requester
2.3.2	SMP	Receives the error response
2.3.3		Use case ends
E4	**Any other error occurred that prevented the SML to process the request**	
2.4.1	SML	Returns an HTTP error 500 as response to the requester

2.4.2	SMP	Receives the error response
2.4.3		Use case ends

Post conditions

Successful conditions

The SMP record has been created into the SML

Failure Conditions (HTTP errors)

unauthorizedFault (401) - returned if the caller is not authorized to invoke this operation (E1)

badRequestFault (400) - returned if the supplied DeleteServiceMetadataPublisherService does not contain consistent data (E2)

notFoundFault (404) - returned if the identifier of the SMP could not be found (E3)

internalErrorFault (500) - returned if the SML service is unable to process the request for any reason (E4)

3.5. Detailed use cases - ManageBusinessIdentifierService

The following paragraphs define the use cases related to the ManageBusinessIdentifier service.

The ManageParticipantIdentifier interface allows Service Metadata Publishers to manage the information in the Service Metadata Locator service relating to individual participant identifiers for which they hold metadata.

This interface requires authentication of the Service Metadata Publisher. The identity of the Service Metadata Publisher derived from the authentication process identifies the Service Metadata Publisher associated with the Participant Identifier(s) which is (are) managed via this interface.

3.5.1. UC5 - Create

Description UC05 Create

Creates an entry in the Service Metadata Locator service for information relating to a specific participant identifier. Regardless of the number of services a recipient exposes, only one record corresponding to the participant identifier is created in the Service Metadata Locator Service by the Service Metadata Publisher which exposes the services for that participant.

Actors

SMP

Preconditions

C1	The user has a valid certificate
C2	The role associated to the user is ROLE_SMP
C3	The SMP does already exist in the SML
C4	The participant does not already exist
C5	Input CreateParticipantIdentifier: ServiceMetadataPublisherServiceForParticipantType - contains the Participant Identifier for a given participant and the identifier of the SMP which holds its data

Basic Flow

Actor	Step	
SMP	1	Invokes the Create() operation
SML	2	Authenticates the user, validates the request, and adds the SMP record into its configuration database.
SML	3	Returns a positive response to the requester
SMP	4	Receives the creation confirmation
	5	Use case ends

Alternative flows

None.

Exception flows

E1	**The user is not authorized**	
2.1.1	SML	Returns an HTTP error 401 as response to the requester
2.1.2	SMP	Receives the error response
2.1.3		Use case ends
E2	**Request is not valid**	
2.2.1	SML	Returns an HTTP error 400 as response to the requester
2.2.2	SMP	Receives the error response
2.2.3		Use case ends
E3	**The SMP could not be found**	
2.3.1	SML	Returns an HTTP error 404 as response to the requester

2.3.2	SMP	Receives the error response
2.3.3		Use case ends
E4	**Any other error occurred that prevented the SML to process the request**	
2.4.1	SML	Returns an HTTP error 500 as response to the requester
2.4.2	SMP	Receives the error response
2.4.3		Use case ends

Post conditions

Successful conditions

The SMP record has been created into the SML

Failure Conditions (HTTP errors)

unauthorizedFault (401) - returned if the caller is not authorized to invoke this operation (E1)

badRequestFault (400) - returned if the supplied CreateParticipantIdentifier does not contain consistent data (E2)

notFoundFault (404) - returned if the identifier of the SMP could not be found (E3)

internalErrorFault (500) - returned if the SML service is unable to process the request for any reason (E4)

3.5.2. UC6 - CreateList

Description UC06 CreateList

Creates a set of entries in the Service Metadata Locator service for information relating to a list of participant identifiers. Regardless of the number of services a recipient exposes, only one record corresponding to each participant identifier is created in the Service Metadata Locator service by the Service Metadata Publisher which exposes the services for that participant.

Actors

SMP

Preconditions

C1	The user has a valid certificate
C2	The role associated to the user is ROLE_SMP
C3	The SMP does already exist in the SML
C4	The participants don't already exist
C5	Input CreateList: ParticipantIdentifierPage - contains the list of Participant Identifiers for the participants which are added to the Service Metadata Locator service. The NextPageIdentifier is absent

Basic Flow

Actor	Step	
SMP	1	Invokes the CreateList() operation
SML	2	Authenticates the user, validates the request, and adds the SMP records into its configuration database.
SML	3	Returns a positive response to the requester
SMP	4	Receives the creation confirmation
	5	Use case ends

Alternative flows

None.

Exception flows

E1	**The user is not authorized**	
2.1.1	SML	Returns an HTTP error 401 as response to the requester
2.1.2	SMP	Receives the error response
2.1.3		Use case ends
E2	**Request is not valid**	
2.2.1	SML	Returns an HTTP error 400 as response to the requester
2.2.2	SMP	Receives the error response
2.2.3		Use case ends
E3	**One or several SMP or participants could not be found**	
2.3.1	SML	Returns an HTTP error 404 as response to the requester

2.3.2	SMP	Receives the error response
2.3.3		Use case ends
E4	**Any other error occurred that prevented the SML to process the request**	
2.4.1	SML	Returns an HTTP error 500 as response to the requester
2.4.2	SMP	Receives the error response
2.4.3		Use case ends

Post conditions

Successful conditions

The SMP records have been created into the SML

Failure Conditions (HTTP errors)

unauthorizedFault (401) - returned if the caller is not authorized to invoke this operation (E1)

badRequestFault (400) - returned if:
o The supplied CreateList does not contain consistent data
o The number of participants in the list is greater than 100
(E2)

notFoundFault (404) - returned if one of the identifiers of the SMPs or of the participants could not be found (E3)

internalErrorFault (500) - returned if the SML service is unable to process the request for any reason (E4)

3.5.3. UC7- Delete

Description UC07 Delete

Deletes the information that the SML Service holds for a specific Participant Identifier.

Actors

SMP

Preconditions

C1	The user has a valid certificate
C2	The role associated to the user is ROLE_SMP
C3	The SMP does already exist in the SML
C4	The participant already exists
C5	Input DeleteParticipantIdentifier : ServiceMetadataPublisherServiceForParticipantType - contains the Participant Identifier for a given participant and the identifier of the SMP that publishes its metadata

Basic Flow

Actor	Step	
SMP	1	Invokes the Delete() operation
SML	2	Authenticates the user, validates the request, and removes the SMP record from its configuration database.
SML	3	Returns a positive response to the requester
SMP	4	Receives the deletion confirmation
	5	Use case ends

Alternative flows

None.

Exception flows

E1	**The user is not authorized**	
2.1.1	SML	Returns an HTTP error 401 as response to the requester
2.1.2	SMP	Receives the error response
2.1.3		Use case ends
E2	**Request is not valid**	
2.2.1	SML	Returns an HTTP error 400 as response to the requester
2.2.2	SMP	Receives the error response
2.2.3		Use case ends
E3	**The SMP or participant could not be found**	
2.3.1	SML	Returns an HTTP error 404 as response to the requester
2.3.2	SMP	Receives the error response
2.3.3		Use case ends
E4	**Any other error occurred that prevented the SML to process the request**	
2.4.1	SML	Returns an HTTP error 500 as response to the requester

2.4.2	SMP	Receives the error response
2.4.3		Use case ends

Post conditions

Successful conditions

The SMP record has been deleted from the SML

Failure Conditions (HTTP errors)

unauthorizedFault (401) - returned if the caller is not authorized to invoke this operation (E1)

badRequestFault (400) - returned if the supplied DeleteParticipantIdentifier does not contain consistent data (E2)

notFoundFault (404) - returned if the identifier of the SMP or of the participant could not be found (E3)

internalErrorFault (500) - returned if the SML service is unable to process the request for any reason (E4)

3.5.4. UC8-DeleteList

Description UC08 DeleteList

Deletes the information that the SML Service holds for a list of Participant Identifiers.

Actors

SMP

Preconditions

C1	The user has a valid certificate
C2	The role associated to the user is ROLE_SMP
C3	The SMP does already exist in the SML
C4	The participant already exists
C5	Input DeleteList: ParticipantIdentifier - contains the list of Participant Identifiers for the participants which are removed from the Service Metadata Locator service. The NextPageIdentifier element is absent.
C6	The number of participants in the list is less than 100

Basic Flow

Actor	Step	
SMP	1	Invokes the DeleteList() operation
SML	2	Authenticates the user, validates the request, and adds the SMP records into its configuration database.
SML	3	Returns a positive response to the requester
SMP	4	Receives the deletion confirmation
	5	Use case ends

Alternative flows

None.

Exception flows

E1	**The user is not authorized**	
2.1.1	SML	Returns an HTTP error 401 as response to the requester
2.1.2	SMP	Recevies the error response
2.1.3		Use case ends
E2	**Request is not valid**	
2.2.1	SML	Returns an HTTP error 400 as response to the requester
2.2.2	SMP	Recevies the error response
2.2.3		Use case ends
E3	**The SMP or participant could not be found**	
2.3.1	SML	Returns an HTTP error 404 as response to the requester
2.3.2	SMP	Recevies the error response
2.3.3		Use case ends
E4	**Any other error occured that prevented the SML to process the request**	

2.4.1	SML	Returns an HTTP error 500 as response to the requester
2.4.2	SMP	Recevies the error response
2.4.3		Use case ends

Post conditions

Successful conditions

The SMP records have been deleted from the SML

Failure Conditions (HTTP errors)

unauthorizedFault (401) - returned if the caller is not authorized to invoke this operation (E1)

badRequestFault (400) - returned if:
o The supplied DeleteList does not contain consistent data
o The number of participants in the list is greater than 100
(E2)

notFoundFault (404) - returned if one of the identifiers of the SMPs or of the participants could not be found (E3)

internalErrorFault (500) - returned if the SML service is unable to process the request for any reason (E4)

3.5.5. UC9 PrepareToMigrate

Description UC09 PrepareToMigrate

Prepares a Participant Identifier for migration to a new Service Metadata Publisher. This operation is called by the Service Metadata Publisher which currently publishes the metadata for the Participant Identifier. The Service Metadata Publisher supplies a Migration Code which is used to control the migration process. The Migration Code must be passed (out of band) to the Service Metadata Publisher which is taking over the publishing of the metadata for the Participant Identifier and which MUST be used on the invocation of the Migrate() operation. This operation can only be invoked by the Service Metadata Publisher which currently publishes the metadata for the specified Participant Identifier.

Actors

SMP

Preconditions

C1	The user has a valid certificate
C2	The role associated to the user is ROLE_SMP
C3	The SMP already exists in the SML
C4	The participant already exists
C5	Input PrepareMigrationRecord: MigrationRecordType - contains the Migration Key and the Participant Identifier which is about to be migrated from one Service Metadata Publisher to another.

Basic Flow

Actor	Step	
SMP	1	Invokes the PrepareToMigrate() operation
SML	2	Authenticates the user, validates the request, and adds the SMP record into its configuration database.
SML	3	Returns a positive response to the requester
SMP	4	Receives the prepared to migrate confirmation
	5	Use case ends

Alternative flows

None.

Exception flows

E1	**The user is not authorized**	
2.1.1	SML	Returns an HTTP error 401 as response to the requester
2.1.2	SMP	Receives the error response
2.1.3		Use case ends
E2	**Request is not valid**	

2.2.1	SML	Returns an HTTP error 400 as response to the requester
2.2.2	SMP	Receives the error response
2.2.3		Use case ends
E3	**The SMP or participant could not be found**	
2.3.1	SML	Returns an HTTP error 404 as response to the requester
2.3.2	SMP	Receives the error response
2.3.3		Use case ends
E4	**Any other error occurred that prevented the SML to process the request**	
2.4.1	SML	Returns an HTTP error 500 as response to the requester
2.4.2	SMP	Receives the error response
2.4.3		Use case ends

Post conditions

Successful conditions

The SMP record is ready to be migrated into the SML

Failure Conditions (HTTP errors)

unauthorizedFault (401) - returned if the caller is not authorized to invoke this operation (E1)

badRequestFault (400) - returned if the supplied PrepateMigrationRecord does not contain consistent data (E2)

notFoundFault (404) - returned if the participant identifier or the identifier of the SMP could not be found (E3)

internalErrorFault (500) - returned if the SML service is unable to process the request for any reason (E4)

3.5.6. UC10 Migrate

Description UC10 Migrate

Migrates a Participant Identifier already held by the Service Metadata Locator service to target a new Service Metadata Publisher. This operation is called by the Service Metadata Publisher which is taking over the publishing for the Participant Identifier. The operation requires the new Service Metadata Publisher to provide a migration code which was originally obtained from the old Service Metadata Publisher. The PrepareToMigrate operation MUST have been previously invoked for the supplied Participant Identifier, using the same MigrationCode, otherwise the Migrate() operation fails. Following the successful invocation of this operation, the lookup of the metadata for the service endpoints relating to a particular Participant Identifier will resolve (via DNS) to the new Service Metadata Publisher.

Actors

SMP

Preconditions

C1	The user has a valid certificate
C2	The role associated to the user is ROLE_SMP
C3	The SMP does already exist in the SML
C4	The participant already exists
C5	The PrepareToMigrate operation MUST have been previously invoked for the supplied Participant Identifier, using the same MigrationCode
C6	Input CompleteMigrationRecord: MigrationRecordType - contains the Migration Key and the Participant Identifier which is to be migrated from one Service Metadata Publisher to another.

Basic Flow

Actor	Step	
SMP	1	Invokes the Migrate() operation
SML	2	Authenticates the user, validates the request, and migrates the SMP record into its configuration database.
SML	3	Returns a positive response to the requester
SMP	4	Receives the migration confirmation
	5	Use case ends

Alternative flows

None.

Exception flows

E1	The user is not authorized	
2.1.1	SML	Returns an HTTP error 401 as response to the requester
2.1.2	SMP	Receives the error response
2.1.3		Use case ends

E2	**Request is not valid**	
2.2.1	SML	Returns an HTTP error 400 as response to the requester
2.2.2	SMP	Receives the error response
2.2.3		Use case ends
E3	**The SMP or migration key could not be found**	
2.3.1	SML	Returns an HTTP error 404 as response to the requester
2.3.2	SMP	Receives the error response
2.3.3		Use case ends
E4	**Any other error occurred that prevented the SML to process the request**	
2.4.1	SML	Returns an HTTP error 500 as response to the requester
2.4.2	SMP	Receives the error response
2.4.3		Use case ends

Post conditions

Successful conditions

The participant identifier has been migrated into the SML

Failure Conditions (HTTP errors)

unauthorizedFault (401) - returned if the caller is not authorized to invoke this operation (E1)

badRequestFault (400) - returned if the supplied CompleteMigrationRecord does not contain consistent data (E2)

notFoundFault (404) -returned if the migration key or the identifier of the SMP could not be found (E3)

internalErrorFault (500) - returned if the SML service is unable to process the request for any reason (E4)

3.5.7. UC11 List

Description UC11 List

List() is used to retrieve a list of all participant identifiers associated with a single Service Metadata Publisher, for synchronization purposes. Since this list may be large, it is returned as pages of data, with each page being linked from the previous page.

Actors

SMP

Preconditions

C1	The user has a valid certificate
C2	The role associated to the user is ROLE_SMP
C3	The SMP does already exist in the SML
C5	Input Page: PageRequest - contains a PageRequest containing the ServiceMetadataPublisherID of the SMP and (if required) an identifier representing the next page of data to retrieve. If the NextPageIdentifier is absent, the first page is returned.

Basic Flow

Actor	Step	
SMP	1	Invokes the List() operation
SML	2	Authenticates the user, validates the request, and builds the list of SMP from its configuration database.
SML	3	Returns the requested list to the requester
SMP	4	Receives the requested list
	5	Use case ends

Alternative flows

None.

Exception flows

E1	**The user is not authorized**	
2.1.1	SML	Returns an HTTP error 401 as response to the requester
2.1.2	SMP	Receives the error response
2.1.3		Use case ends
E2	**Request is not valid**	
2.2.1	SML	Returns an HTTP error 400 as response to the requester
2.2.2	SMP	Receives the error response
2.2.3		Use case ends
E3	**The SMP or migration key could not be found**	
2.3.1	SML	Returns an HTTP error 404 as response to the requester
2.3.2	SMP	Receives the error response
2.3.3		Use case ends
E4	**Any other error occurred that prevented the SML to process the request**	

2.4.1	SML	Returns an HTTP error 500 as response to the requester
2.4.2	SMP	Receives the error response
2.4.3		Use case ends

Post conditions

Successful conditions

Output: ParticipantIdentifierPage - a page of Participant Identifier entries associated with the Service Metadata Publisher, also containing a <Page/> element containing the identifier that represents the next page, if any.

Note that the underlying data may be updated between one invocation of List() and a subsequent invocation of List(), so that a set of retrieved pages of participant identifiers may not represent a consistent set of data.

Failure Conditions (HTTP errors)

unauthorizedFault (401) - returned if the caller is not authorized to invoke this operation (E1)

badRequestFault (400) - returned if the supplied NextPage does not contain consistent data (E2)

notFoundFault (404) - returned if the next page or the identifier of the SMP could not be found (E3)

internalErrorFault (500) - returned if the SML service is unable to process the request for any reason (E4)

3.6. Detailed use cases – BDMSLService

This interface describes non-core services that are not defined in the SML or BDX specifications.

The following paragraphs define the use cases related to the BDMSLService service.

3.6.1. UC12- PrepareChangeCertificate

Description UC12 PrepareChangeCertificate
This operation allows an SMP to prepare a change of its certificate. It is typically called when an SMP has a certificate that is about to expire and already has the new one. This operation MUST be called while the certificate that is already registered in the BDMSL is still valid. If the migrationDate is not empty, then the new certificate MUST be valid at the date provided in the migrationDate element. If the migrationDate element is empty, then the "Valid From" date is extracted from the certificate and is used as the migrationDate. In this case, the "Not Before" date of the certificate must be in the future.

Actors
SMP

Preconditions	
C1	The user has a valid certificate
C2	The role associated to the user is ROLE_SMP
C3	The user has the new certificate for the SMP(s)
C4	Input: PrepareChangeCertificate containing the new certificate and the validity start date

Basic Flow		
Actor	**Step**	
SMP	1	Invokes the PrepareChangeCertificate() operation
SML	2	Authenticates the user, validates the request, and stores the future certificate into its configuration database.
SML	3	Returns a positive response to the requester
SMP	4	Receives the creation confirmation
	5	Use case ends

Alternative flows
None.

Exception flows		
E1	**The user is not authorized**	
2.1.1	SML	Returns an HTTP error 401 as response to the requester
2.1.2	SMP	Receives the error response
2.1.3		Use case ends
E2	**Request is not valid**	
2.2.1	SML	Returns an HTTP error 400 as response to the requester
2.2.2	SMP	Receives the error response
2.2.3		Use case ends

E3	Any other error occurred that prevented the SML to process the request	
2.3.1	SML	Returns an HTTP error 500 as response to the requester
2.3.2	SMP	Receives the error response
2.3.3		Use case ends

Post conditions

Successful conditions

The Metadata record has been created into the SML

Failure Conditions (HTTP errors)

unauthorizedFault (401) - returned if the caller is not authorized to invoke this operation (E1)

badRequestFault (400) - returned if :
o The supplied request does not contain consistent data
o The new certificate is not valid at the date provided in the migrationDate element
o The migrationDate is not in the future.
o The migrationDate is not provided and the "Not Before" date of the new certificate is not in the future
o The migrationDate is not provided and the "Valid From" is in the past
(E2)

internalErrorFault (500) - returned if the SML service is unable to process the request for any reason (E3)

3.6.2. UC13- CreateParticipantIdentifier

Description UC13 CreateParticipantIdentifier

This service has the same behaviour as the Create() operation in the ManageParticipantIdentifier interface but it has one additional and optional() Input: the serviceName element. In the Create() operation, the service name is "Meta:SMP" by default. In the CreateParticipantIdentifier() operation, this service name can be customized.
- serviceName: the name of the service for the NAPTR record

Actors

SMP

Preconditions

C1	The user has a valid certificate
C2	The role associated to the user is ROLE_SMP
C3	The SMP does already exist in the SML
C4	The participant does not already exist
C5	Input CreateParticipantIdentifier: ServiceMetadataPublisherServiceForParticipantType - contains the Participant Identifier for a given participant and the identifier of the SMP which holds its data Additional parameter: serviceName

Basic Flow

Actor	Step	
SMP	1	Invokes the CreateParticipantIdentifier() operation
SML	2	Authenticates the user, validates the request, and adds the SMP record into its configuration database.
SML	3	Returns a positive response to the requester
SMP	4	Receives the creation confirmation
	5	Use case ends

Alternative flows

None.

Exception flows

E1	**The user is not authorized**	
2.1.1	SML	Returns an HTTP error 401 as response to the requester
2.1.2	SMP	Receives the error response
2.1.3		Use case ends
E2	**Request is not valid**	
2.2.1	SML	Returns an HTTP error 400 as response to the requester
2.2.2	SMP	Receives the error response
2.2.3		Use case ends
E3	**The SMP could not be found**	

2.3.1	SML	Returns an HTTP error 404 as response to the requester
2.3.2	SMP	Receives the error response
2.3.3		Use case ends

E4	**Any other error occurred that prevented the SML to process the request**
2.4.1	SML	Returns an HTTP error 500 as response to the requester
2.4.2	SMP	Receives the error response
2.4.3		Use case ends

Post conditions

Successful conditions

The SMP record has been created into the SML

Failure Conditions (HTTP errors)

unauthorizedFault (401) - returned if the caller is not authorized to invoke this operation (E1)

badRequestFault (400) - returned if the supplied CreateParticipantIdentifier does not contain consistent data (E2)

notFoundFault (404) - returned if the identifier of the SMP could not be found (E3)

internalErrorFault (500) - returned if the SML service is unable to process the request for any reason (E4)

3.6.3. UC14 IsAlive

Description UC14 IsAlive

This service has only a monitoring purpose. It can be called to check if the application is up and running. This service checks if the database and the DNS are accessible by trying to read from the database and to write to and read from DNS.

Actors

SMP

Preconditions

C1	The user has a valid certificate
C2	The user has the role ROLE_SMP or ROLE_ADMIN
C4	Input: None

Basic Flow

Actor	Step	
SMP	1	Invokes the IsAlive() operation
SML	2	Authenticates the user
SML	3	Returns a positive response to the requester
SMP	4	Receives the alive confirmation
	5	Use case ends

Alternative flows

None.

Exception flows

E1	**The user is not authorized**	
2.1.1	SML	Returns no response to the requester
2.1.3		Use case ends
E2	**Any other error occurred that prevented the SML to process the request**	
2.2.1	SML	Returns no response to the requester
2.2.2		Use case ends

Post conditions

Successful conditions

HTTP 200 OK response sent to the requester.

Failure Conditions (HTTP errors)

unauthorizedFault (401) - returned if the caller is not authorized to invoke this operation (E1)

internalErrorFault (500) - returned if the SML service is unable to process the request for any reason (E2)

3.6.4. UC15 ClearCache

Description UC15 ClearCache

The in-memory caches are used for:
• The list of trusted aliases and their corresponding domains, because these data are not supposed to be changed frequently.
• The content of the Certificate Revocation List, in order to avoid the cost of downloading each time the CRLM for each certificate.

Actors

SMP

Preconditions

C1	The user has a valid certificate
C2	The user has the role ROLE_SMP or ROLE_ADMIN
C3	Input: none.

Basic Flow

Actor	Step	
SMP	1	Invokes the ClearCache() operation
SML	2	Authenticates the user, validates the request, and clears the in-memory cache.
SML	3	Returns a positive response to the requester
SMP	4	Receives the reset confirmation
	5	Use case ends

Alternative flows

None.

Exception flows

E1	**The user is not authorized**	
2.1.1	SML	Returns an HTTP error 401 as response to the requester
2.1.2	SMP	Receives the error response
2.1.3		Use case ends
E2	**Any other error occurred that prevented the SML to process the request**	
2.2.1	SML	Returns an HTTP error 500 as response to the requester
2.2.2	SMP	Receives the error response
2.2.3		Use case ends

Post conditions

Successful conditions

The cache is reset. HTTP 200 OK response sent to the requester.

Failure Conditions (HTTP errors)

No or unspecified type of response (E1, E2)

3.6.5. UC16 ChangeCertificate

Description UC16 ChangeCertificate

This operation allows the admin team to change the SMP's certificate. It is called by the admin team in case the SMP's certificate has expired and the new one needs to be applied. The new certificate MUST be valid at the date time the request is sent.

Actors

ADMIN

Preconditions

C1	The user has a valid certificate
C2	The role associated to the user is ROLE_ADMIN
C3	The user has the new certificate for the SMP(s)
C4	Input : SMP id, New certificate public key

Basic Flow

Actor	Step	
SMP	1	Invokes the ChangeCertificate() operation
SML	2	Authenticates the user, validates the request, and stores the new certificate into its configuration database.
SML	3	Returns a positive response to the requester
SMP	4	Receives the creation confirmation
	5	Use case ends

Alternative flows

None.

Exception flows

E1	**The user is not authorized**	
2.1.1	SML	Returns an HTTP error 401 as response to the requester
2.1.2	SMP	Receives the error response
2.1.3		Use case ends
E2	**Request is not valid**	
2.2.1	SML	Returns an HTTP error 400 as response to the requester
2.2.2	SMP	Receives the error response
2.2.3		Use case ends
E3	**Any other error occurred that prevented the SML to process the request**	
2.3.1	SML	Returns an HTTP error 500 as response to the requester
2.3.2	SMP	Receives the error response
2.3.3		Use case ends

Post conditions

Successful conditions

New Certificate is stored

Output : none.

HTTP 200 OK expected

Failure Conditions (HTTP errors)

unauthorizedFault (401) - returned if the caller is not authorized to invoke this operation (E1)

badRequestFault (400) - returned if :
o The supplied request does not contain consistent data
o The new certificate is not valid at the date provided in the migrationDate element
o The migrationDate is not in the future.
o The migrationDate is not provided and the "Not Before" date of the new certificate is not in the future
o The migrationDate is not provided and the "Valid From" is in the past
(E2)

internalErrorFault (500) - returned if the SML service is unable to process the request for any reason (E3)

3.6.6. UC17 ListParticipants

Description UC17 ListParticipants

Lists all the participants managed by the BDMSL. This service is only meant to be called by the Directory user application.

Actors

SMP

Preconditions

C1	The user has a valid certificate
C2	The user has the role ROLE_PYP
C3	Input: none

Actor	Step	
SMP	1	Invokes the ListParticipants() operation
SML	2	Authenticates the user, validates the request, and retrieves the list of participants from its configuration database.
SML	3	Returns the list to the requester
SMP	4	Receives the list
	5	Use case ends

Alternative flows

None.

Exception flows

E1	**The user is not authorized**	
2.1.1	SML	Returns an HTTP error 401 as response to the requester
2.1.2	SMP	Receives the error response
2.1.3		Use case ends
E2	**Any other error occurred that prevented the SML to process the request**	
2.2.1	SML	Returns an HTTP error 500 as response to the requester
2.2.2	SMP	Receives the error response
2.2.3		Use case ends

Post conditions

Successful conditions

Output : ListParticipantsType : the complete list of the participants managed by the BDMSL

Failure Conditions (HTTP errors)

unauthorizedFault (401) - returned if the caller is not authorized to invoke this operation (E1)
internalErrorFault (500) - Returned if the BDMSL service is unable to process the request for any reason (E2)

4. INTERFACE BEHAVIOURAL SPECIFICATION

4.1. Sequence diagrams

4.1.1. ManageServiceMetadataService

(cf. [REF3], p.18)

The ManageServiceMetadata interface allows Service Metadata Publishers to manage the metadata held in the Service Metadata Locator service about their service metadata publisher services, e.g. binding, interface profile and key information. This interface requires authentication of the user. The identity of the user derived from the authentication process identifies the Service Metadata Publisher associated with the service metadata which is managed via this interface. The ManageServiceMetadata interface has the following operations:

- Create
- Read
- Update
- Delete

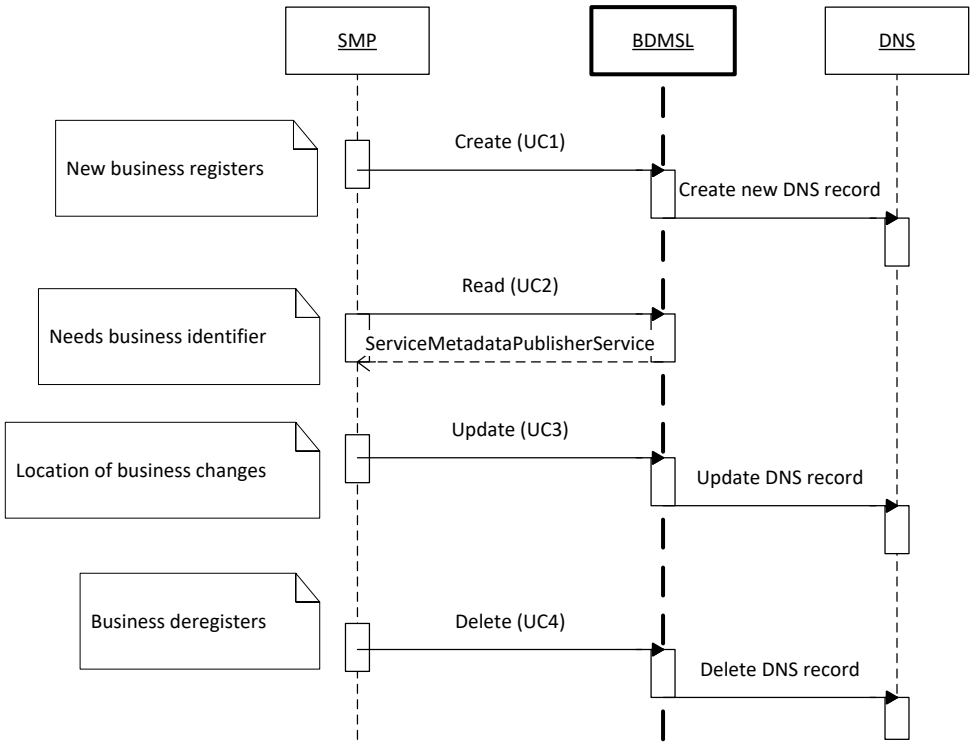


Figure 2 – Sequence diagram: *ManageServiceMetadataService*

4.1.2. *ManageBusinessIdentifierService*

The ManageParticipantIdentifier interface has the following operations:

- Create
- CreateList
- Delete
- DeleteList
- PrepareToMigrate
- Migrate
- List

These services are listed in the sequence diagram below:

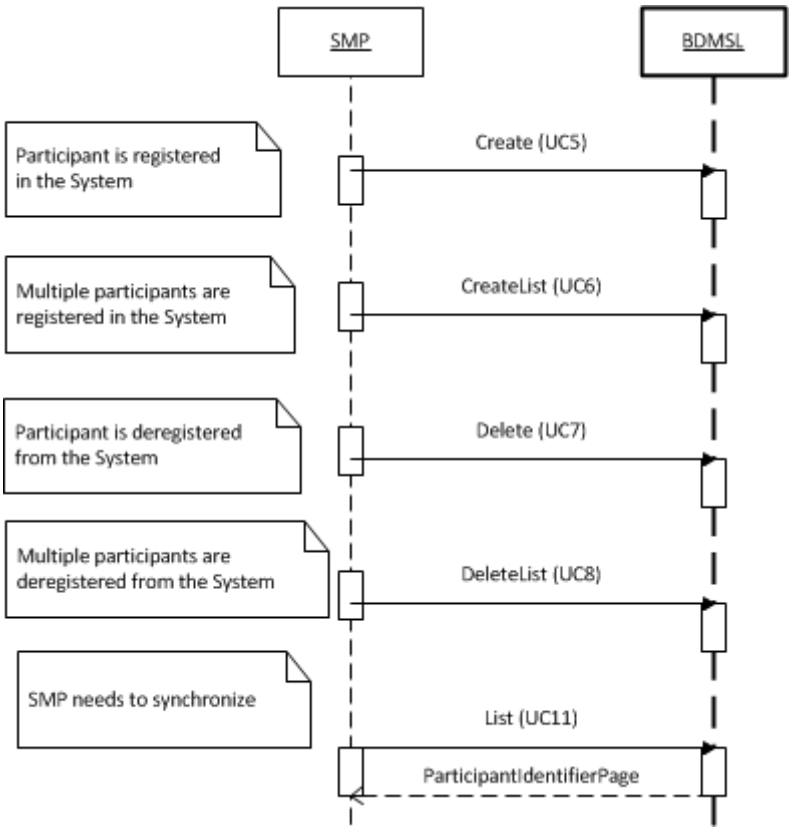


Figure 3 – Sequence diagram: *ManageBusinessIdentifierService*

The usage of the services related to the SMP migration process – involving more than a single step like the others above – are shown in the sequence diagram below:

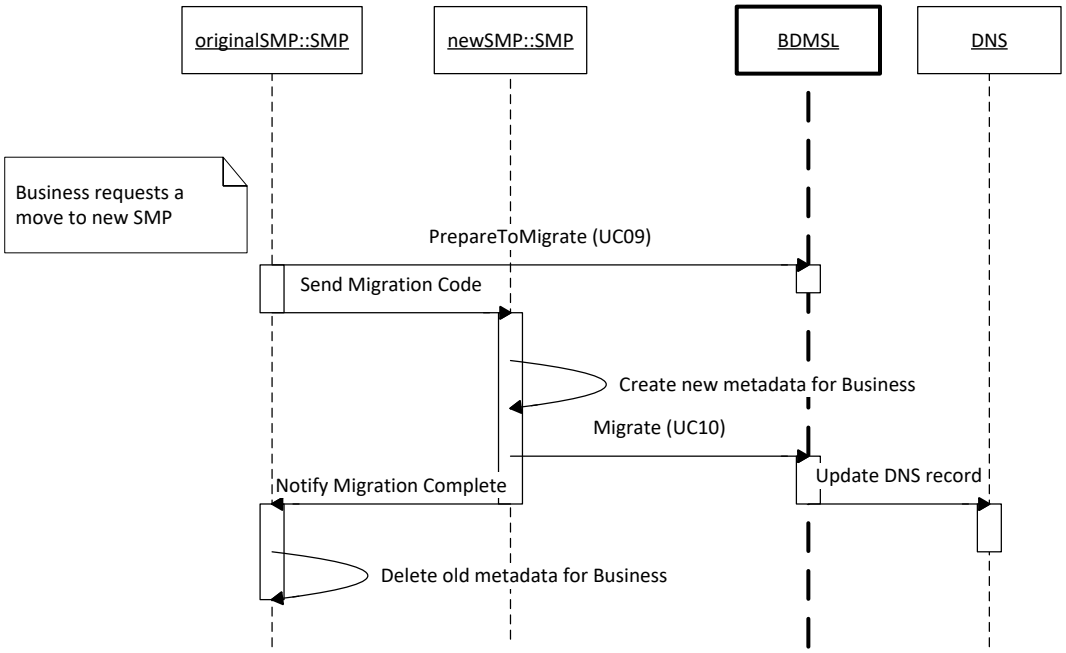


Figure 4 – Sequence diagram : SMP Migraion

4.1.3. BDMSLService

This interface describes non-core services that are not defined in the SML or BDX specifications.

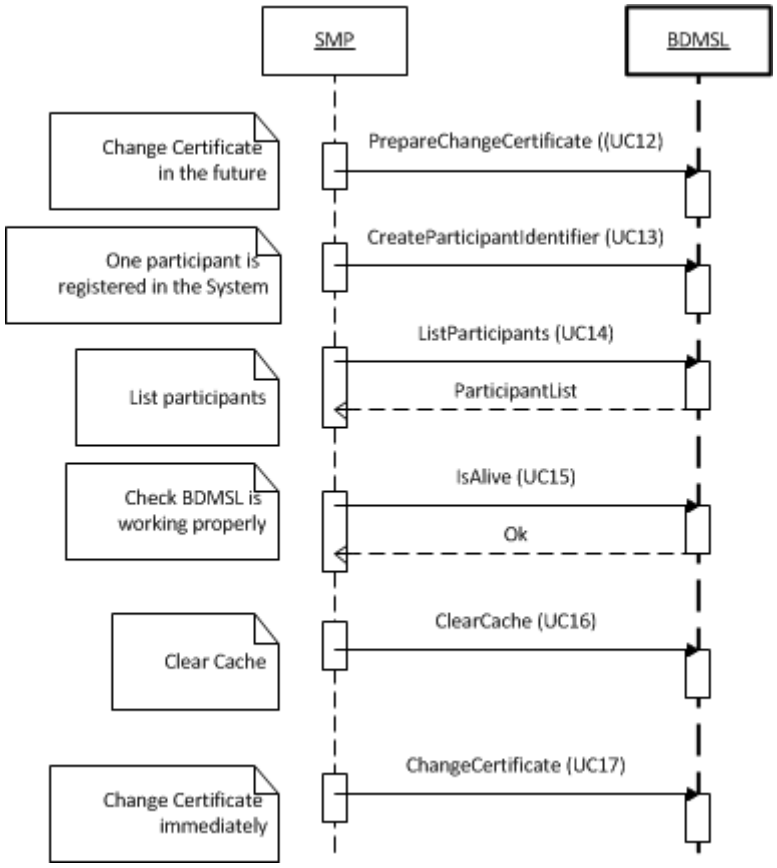
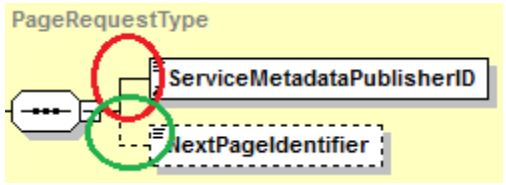


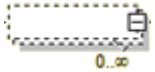
Figure 5 – Sequence diagram : BDMSLService

4.2. WSDL – Data model

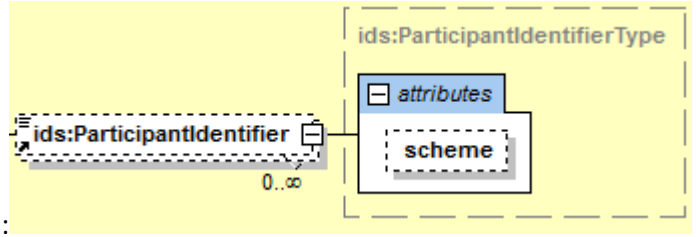
The interface data model of the BDMSL is described as follows:

- One paragraph for each of the 3 web services will introduce all their operations
- One paragraph for each operation will specify their Input and Output structures and the fault that these operations may return. In some cases there is no argument, in which case there is no related structure (the text below will mention "none" in those cases).
- For each input or output structure, the related XSD structure is detailed in a graphical way specifying:
    - The arborescence structure;
    - The mandatory (solid lines) and optional (dashed line) fields.

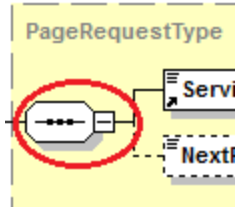
Example: 

    - The repeated fields and their cardinality (icon 

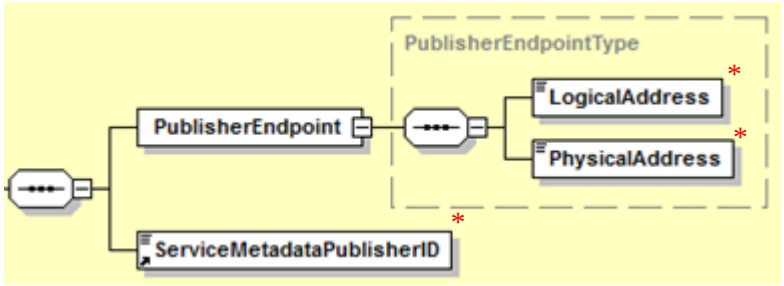
with min/max indication on the bottom right).

Example: 

- The sequences (icon ).

Example: 

- The leave attributes as defined in the first paragraph.

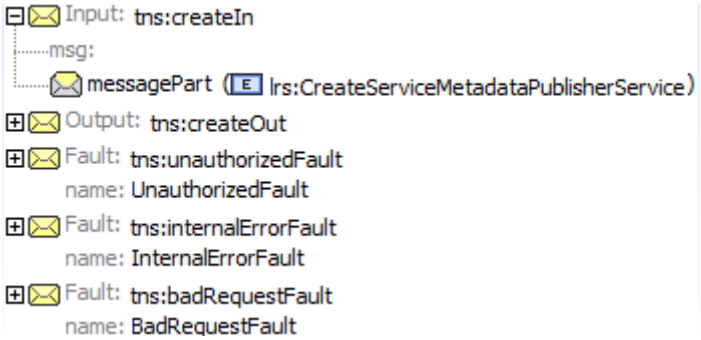
Examples (*): 

4.2.1. WSDL model for ManageServiceMetadataService

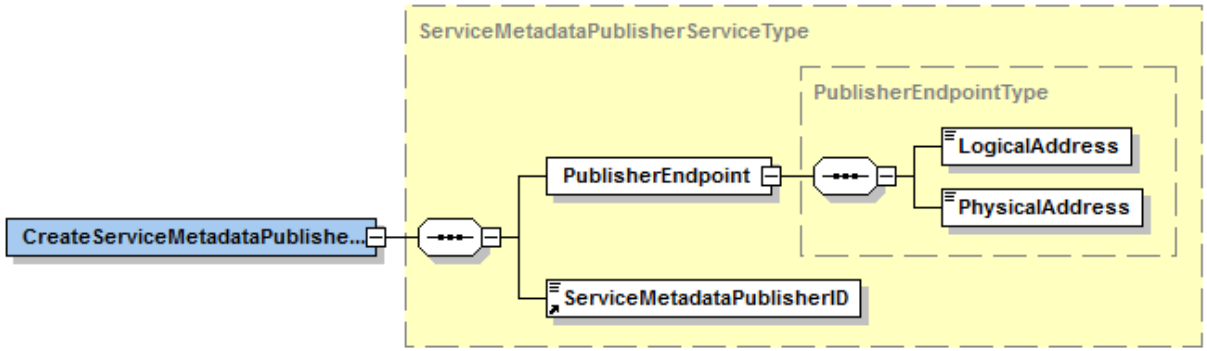
Figure 6 – WSDL ManageServiceMetadataService

4.2.1.1. Operations Signatures

4.2.1.1.1. Create() Signature



4.2.1.1.2. Create() Input



argument	Description	Format/XSD/Xpath	Constraint
PublisherEndpoint.LogicalAddress	The logical address of the endpoint (Domain name)	xs:anyURI ServiceMetadataLocatorTypes-1.0.xsd /*[local-name()='schema']/*[local-name()='complexType' and @name='PublisherEndpointType']//*[local-name()='sequence']/*[local-name()='element' and @name='LogicalAddress']	Must not be null or empty. Must be valid and well formatted.
PublisherEndpoint.PhysicalAddress	IP Address of the endpoint	xs:string ServiceMetadataLocatorTypes-1.0.xsd /*[local-name()='schema']/*[local-name()='complexType' and @name='PublisherEndpointType']//*[local-name()='sequence']/*[local-name()='element' and @name='PhysicalAddress']	This physical address is used as the ALIAS on the CNAME DNS record by default. However a NAPTR DNS record is also provided in order to give the possibility to process regular expressions for accessing the domain if necessary. NAPTR records are based

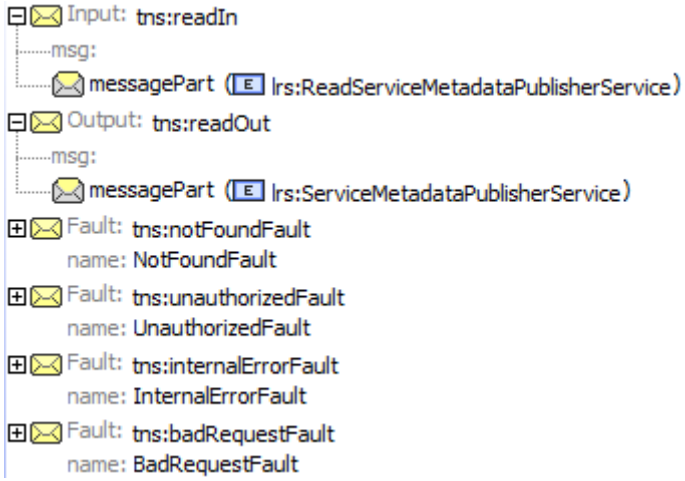
argument	Description	Format/XSD/Xpath	Constraint
			on a different type of DNS resource records called "URI-enabled Naming Authority Pointer records" (U-NAPTR), which are defined to support Dynamic Delegation Discovery Service (DDDS). The result of a query is a full URI, which can use HTTPS and supports server and optionally client authentication. Must not be null or empty. Must be valid according to IPv4 and well formatted.
ServiceMetadataPublisherID	Unique identifier of the SMP	xs:string ServiceMetadataLocatorTypes-1.0.xsd /*[local-name()='schema']/*[local-name()='complexType' and @name='ServiceMetadataPublisherServiceForParticipantType']//*[local-name()='sequence']/*[local-name()='element' and @ref='ServiceMetadataPublisherID']	In ManageServiceMetadata service, this establishes the link with the managing SMP of the participant as defined by in the ManageBusinessIdentifier service.

Table 4 – Create() input

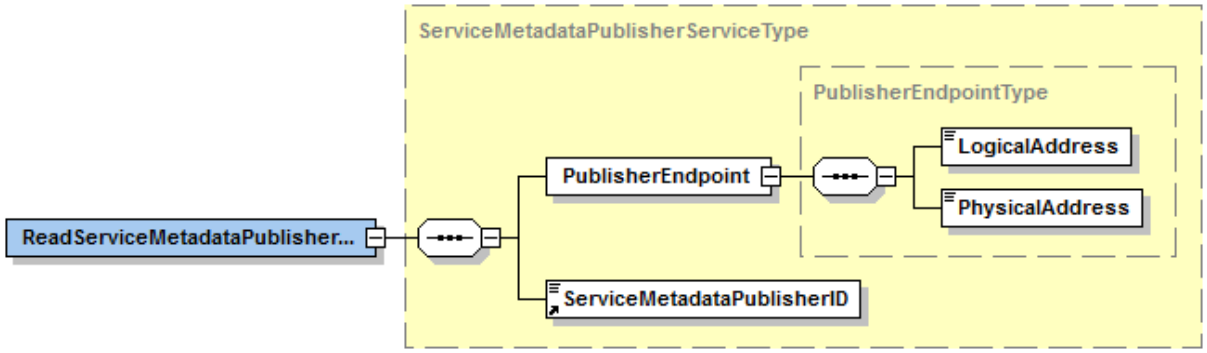
4.2.1.1.3. Create() Output

None

4.2.1.1.4. Read() Signature



4.2.1.1.5. Read() Input

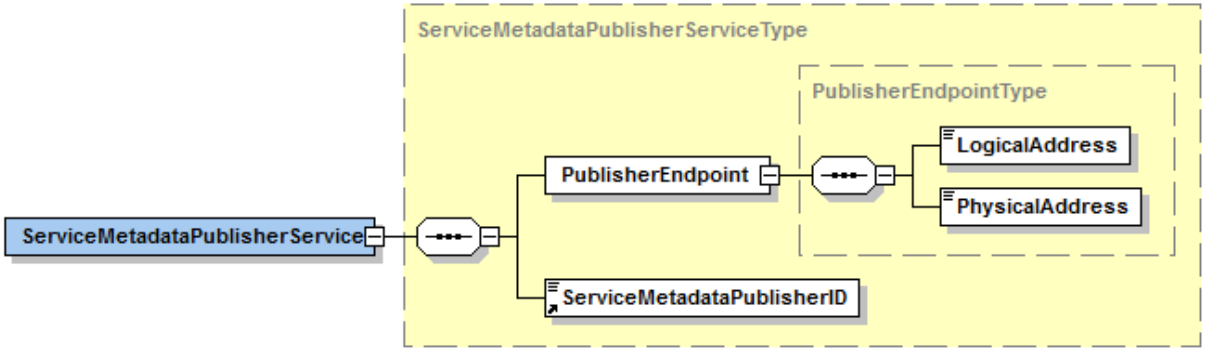


argument	Description	Format/XSD/Xpath	Constraint
PublisherEndpoint.LogicalAddress	The logical address of the endpoint (Domain name)	xs:anyURI ServiceMetadataLocatorTypes-1.0.xsd /*[local-name()='schema']/*[local-name()='complexType' and @name='PublisherEndpointType']//*[local-name()='sequence']/*[local-name()='element' and @name='LogicalAddress']	Must not be null or empty. Must be valid and well formatted.
PublisherEndpoint.PhysicalAddress	IP Address of the endpoint	xs:string ServiceMetadataLocatorTypes-1.0.xsd /*[local-name()='schema']/*[local-name()='complexType' and @name='PublisherEndpointType']//*[local-name()='sequence']/*[local-name()='element' and @name='PhysicalAddress']	This physical address is used as the ALIAS on the CNAME DNS record by default. However a NAPTR DNS record is also provided in order to give the possibility to process regular expressions for accessing the domain if necessary. NAPTR records are based on a different type of DNS resource records called "URI-enabled Naming Authority Pointer records" (U-NAPTR), which are defined to support Dynamic Delegation Discovery Service (DDDS). The result of a query is a full URI, which can use HTTPS and supports server and optionally client

argument	Description	Format/XSD/Xpath	Constraint
			authentication. Must not be null or empty. Must be valid according to IPv4 and well formatted.
ServiceMetadataPublisherID	Unique identifier of the SMP	xs:string ServiceMetadataLocatorTypes-1.0.xsd /*[local-name()='schema']/*[local-name()='complexType' and @name='ServiceMetadataPublisherServiceForParticipantType']//*[local-name()='sequence']/*[local-name()='element' and @ref='ServiceMetadataPublisherID']	In ManageServiceMetadata service, this establishes the link with the managing SMP of the participant as defined by in the ManageBusinessIdentifier service.

Table 5 – Read input()

4.2.1.1.6. Read() Output

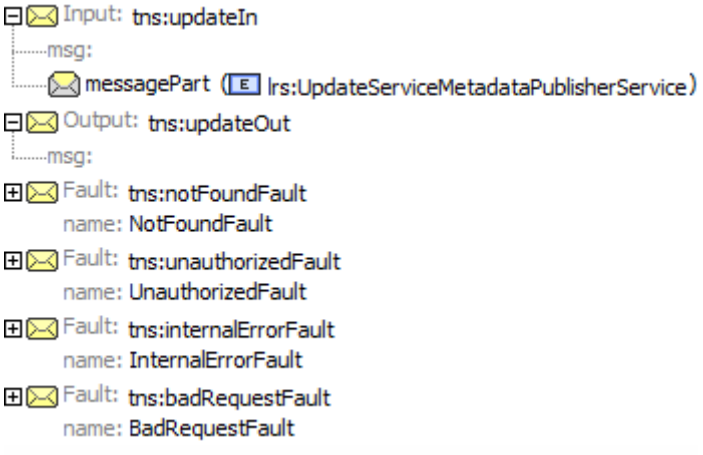


argument	Description	Format/XSD/Xpath	Constraint
PublisherEndpoint.LogicalAddress	The logical address of the endpoint (Domain name)	xs:anyURI ServiceMetadataLocatorTypes-1.0.xsd /*[local-name()='schema']/*[local-name()='complexType' and @name='PublisherEndpointType']//*[local-name()='sequence']/*[local-name()='element' and @name='LogicalAddress']	Must not be null or empty. Must be valid and well formatted.
PublisherEndpoint.PhysicalAddress	IP Address of the endpoint	xs:string ServiceMetadataLocatorTypes-1.0.xsd /*[local-name()='schema']/*[local-name()='complexType' and @name='PublisherEndpointType']//*[local-name()='sequence']/*[local-name()='element' and @name='PhysicalAddress']	This physical address is used as the ALIAS on the CNAME DNS record by default. However a NAPTR DNS record is also provided in order to give the possibility to process regular expressions for accessing the domain if necessary. NAPTR records are based

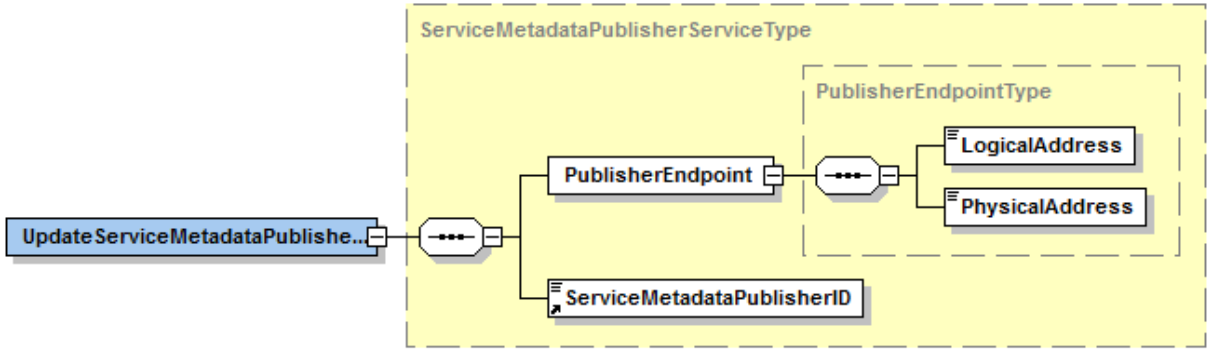
argument	Description	Format/XSD/Xpath	Constraint
			on a different type of DNS resource records called "URI-enabled Naming Authority Pointer records" (U-NAPTR), which are defined to support Dynamic Delegation Discovery Service (DDDS). The result of a query is a full URI, which can use HTTPS and supports server and optionally client authentication. Must not be null or empty. Must be valid according to IPv4 and well formatted.
ServiceMetadataPublisherID	Unique identifier of the SMP	xs:string ServiceMetadataLocatorTypes-1.0.xsd /*[local-name()='schema']/*[local-name()='complexType' and @name='ServiceMetadataPublisherServiceForParticipantType']//*[local-name()='sequence']/*[local-name()='element' and @ref='ServiceMetadataPublisherID']	In ManageServiceMetadata service, this establishes the link with the managing SMP of the participant as defined by in the ManageBusinessIdentifier service.

Table 6 – Read() ouput

4.2.1.1.7. Update() Signature



4.2.1.1.8. Update() Input



argument	Description	Format/XSD/Xpath	Constraint
PublisherEndpoint.LogicalAddress	The logical address of the endpoint (Domain name)	xs:anyURI ServiceMetadataLocatorTypes-1.0.xsd /*[local-name()='schema']/*[local-name()='complexType' and @name='PublisherEndpointType']//*[local-name()='sequence']/*[local-name()='element' and @name='LogicalAddress']	Must not be null or empty. Must be valid and well formatted.
PublisherEndpoint.PhysicalAddress	IP Address of the endpoint	xs:string ServiceMetadataLocatorTypes-1.0.xsd /*[local-name()='schema']/*[local-name()='complexType' and @name='PublisherEndpointType']//*[local-name()='sequence']/*[local-name()='element' and @name='PhysicalAddress']	This physical address is used as the ALIAS on the CNAME DNS record by default. However a NAPTR DNS record is also provided in order to give the possibility to process regular expressions for accessing the domain if necessary. NAPTR records are based on a different type of DNS resource records called "URI-enabled Naming Authority Pointer records" (U-NAPTR), which are defined to support Dynamic Delegation Discovery Service (DDDS). The result of a query is a full URI, which can use HTTPS and supports server and optionally client authentication.

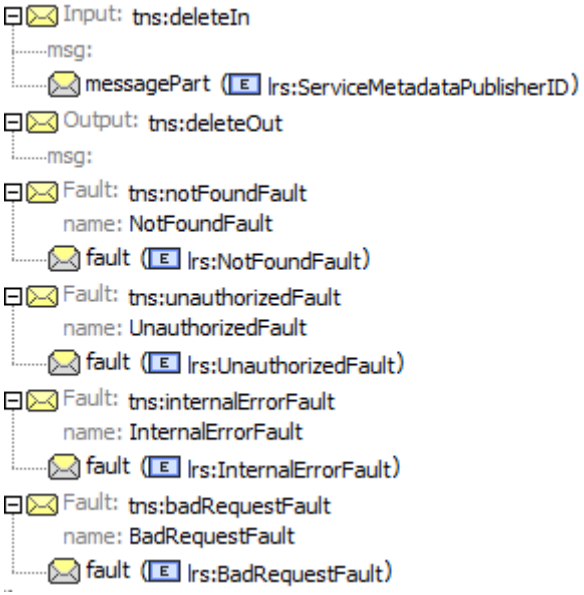
argument	Description	Format/XSD/Xpath	Constraint
			Must not be null or empty. Must be valid according to IPv4 and well formatted.
ServiceMetadataPublisherID	Unique identifier of the SMP	xs:string ServiceMetadataLocatorTypes-1.0.xsd /*[local-name()='schema']/*[local-name()='complexType' and @name='ServiceMetadataPublisherServiceForParticipantType']//*[local-name()='sequence']/*[local-name()='element' and @ref='ServiceMetadataPublisherID']	In ManageServiceMetadata service, this establishes the link with the managing SMP of the participant as defined by in the ManageBusinessIdentifier service.

Table 7 – Update() input

4.2.1.1.9. Update() Output

None

4.2.1.1.10. Delete() Signature



4.2.1.1.11. Delete() Input

argument	Description	Format/XSD/Xpath	Constraint
ServiceMetadataPublisherID	Unique identifier of the SMP	xs:string ServiceMetadataLocatorTypes-1.0.xsd /*[local-name()='schema']/*[local-name()='complexType' and @name='ServiceMetadataPublisherServiceForParticipantType']//*[local-name()='sequence']/*[local-name()='element' and @ref='ServiceMetadataPublisherID']	In ManageServiceMetadata service, this establishes the link with the managing SMP of the participant as defined by in the ManageBusinessIdentifier service.

Table 8 – Delete() input

4.2.1.1.12. Delete() Output

None

4.2.2. WSDL model for ManageBusinessIdentifierService

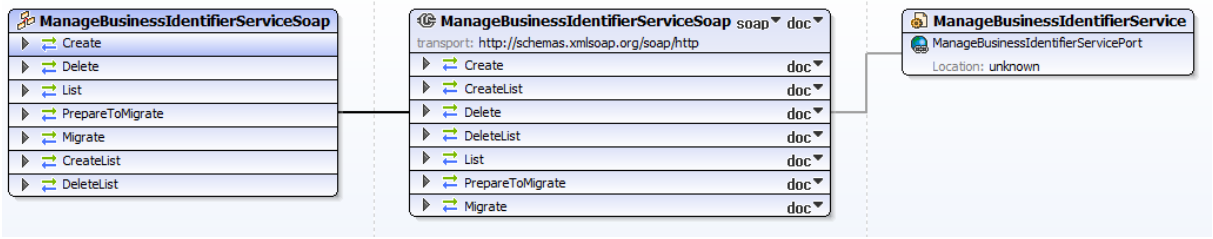
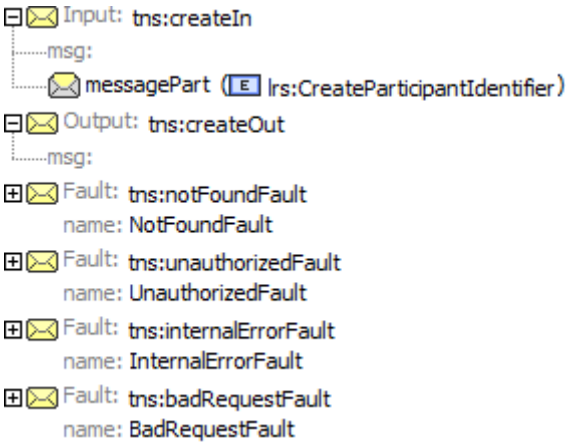


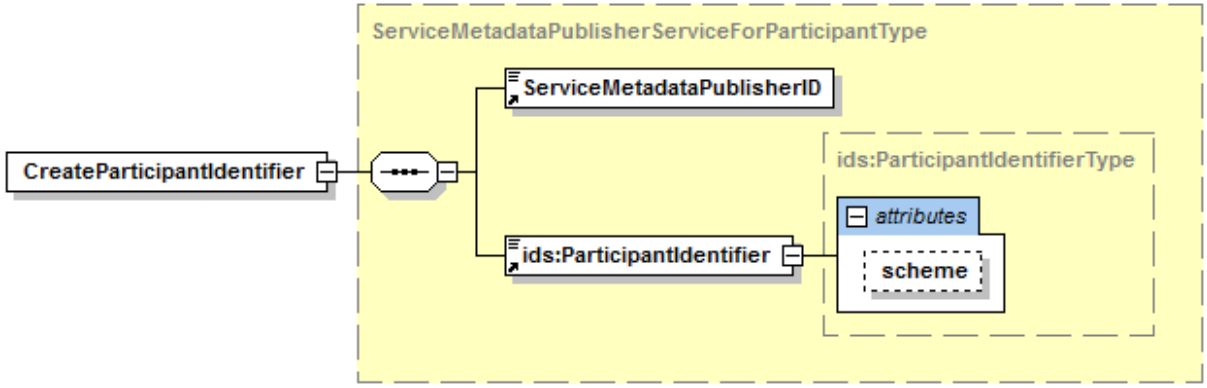
Figure 7 – WSDL ManageBusinessIdentifierService

4.2.2.1. Operations Signatures

4.2.2.1.1. Create() Signature



4.2.2.1.2. Create() Input



argument	Description	Format/XSD/Xpath	Constraint
ServiceMetadataPublisherID	Unique identifier of the SMP	xs:string ServiceMetadataLocatorTypes-1.0.xsd /*[local-name()='schema']/*[local-name()='complexType' and @name='ServiceMetadataPublisherServiceForParticipantType']//*[local-name()='sequence']/*[local-name()='element' and @ref='ServiceMetadataPublisherID']	In ManageServiceMetadata service, this establishes the link with the managing SMP of the participant as defined by in the ManageBusinessIdentifier service.

argument	Description	Format/XSD/Xpath	Constraint
ParticipantIdentifier	Business unique identifier of the Participant. Represents a business level endpoint key that uniquely identifies an end-user entity in the network. Examples of identifiers are company registration and VAT numbers, DUNS numbers, GLN numbers, email addresses, etc…	The format must comply with ISO 15459 constraints as defined in [REF6]. Example: 0088:4035811991014 Identifiers-1.0.xsd /*[local-name()='schema']/*[local-name()='complexType' and @name='ParticipantIdentifierType']	Must be unique. May not exceed 50 characters and must be at least 1 character long. May only contain ASCII characters. Participant identifier must be trimmed. If the scheme refers to the (default) PEPPOL participant identifier scheme (iso6523-actorid-upis) and the participant identifier is not a wildcard ('*'), then the issuing agency specified in the ParticipantIdentifier (as its leading part before ':') must be included in the official list (cf. §0 - "List of valid issuing agencies for PEPPOL") Remark: the participant identifier is case insensitive.

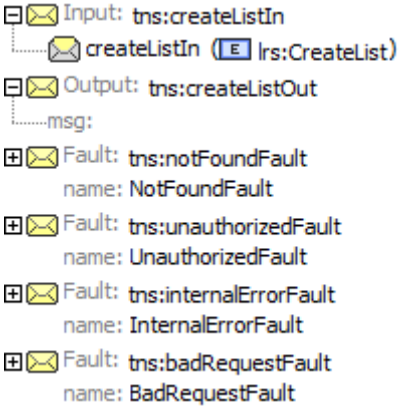
argument	Description	Format/XSD/Xpath	Constraint
ParticipantIdentifier.scheme	The scheme of the participant identifier	Identifier schemes for all schemed identifier types (participants, documents, profiles, transports) may be defined outside of this specification. Any instance of a 4-cornered infrastructure may choose to define identifier schemes that match the type of documents, participants or profiles that are relevant to support in that instance. Examples: iso6523-actorid-upis, busdox-actorid-upis Identifiers-1.0.xsd /*[local-name()='schema']/*[local-name()='complexType' and @name='ProcessIdentifierType']/xs:simpleContent/xs:extension/xs:attribute	May not exceed 25 characters. Must match the two following pattern: [a-zA-Z0-9]+-[a-zA-Z0-9]+-[a-zA-Z0-9] Which defines the following parts: <domain>-<identifier Area>-<identifier type>

Table 9 – Create() input

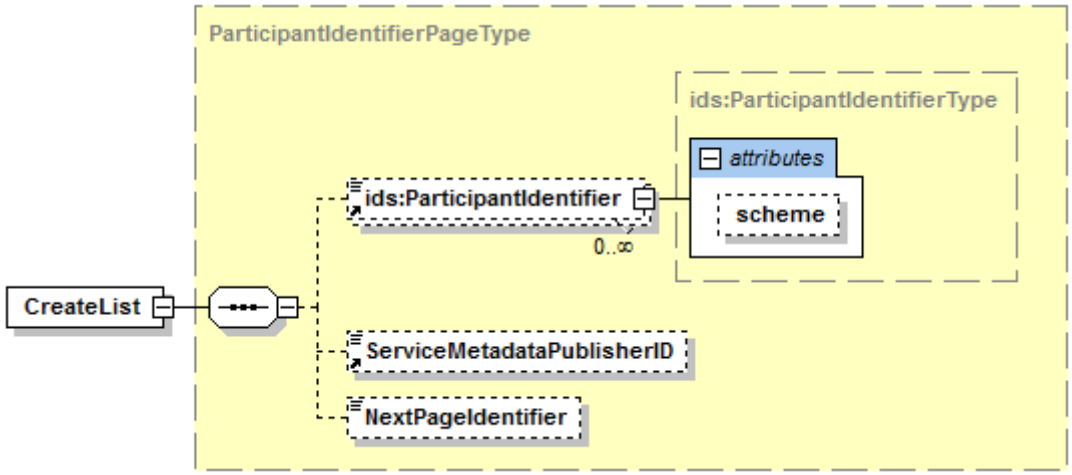
4.2.2.1.3. Create() Output

None

4.2.2.1.4. CreateList() Signature



4.2.2.1.5. CreateList() Input



Note: the NextPageIndentifier is absent.

argument	Description	Format/XSD/Xpath	Constraint
ParticipantIdentifier (0..n)	Business unique identifier of the Participant. Represents a business level endpoint key that uniquely identifies an end-user entity in the network. Examples of identifiers are company registration and VAT numbers, DUNS numbers, GLN numbers, email addresses, etc…	The format must comply with ISO 15459 constraints as defined in [REF6]. Example: 0088:4035811991014 Identifiers-1.0.xsd /*[local-name()='schema']/*[local-name()='complexType' and @name='ParticipantIdentifierType']	Must be unique. May not exceed 50 characters and must be at least 1 character long. May only contain ASCII characters. Participant identifier must be trimmed. If the scheme refers to the (default) PEPPOL participant identifier scheme (iso6523-actorid-upis) and the participant identifier is not a wildcard ('*'), then the issuing agency specified in the ParticipantIdentifier (as its leading part before ':') must be included in the official list (cf. §0 - "List of valid issuing agencies for PEPPOL") Remark: the participant identifier is case insensitive.

argument	Description	Format/XSD/Xpath	Constraint
ParticipantIdentifier.scheme (0..n)	The scheme of the participant identifier	Identifier schemes for all schemed identifier types (participants, documents, profiles, transports) may be defined outside of this specification. Any instance of a 4-cornered infrastructure may choose to define identifier schemes that match the type of documents, participants or profiles that are relevant to support in that instance. Examples: iso6523-actorid-upis, busdox-actorid-upis Identifiers-1.0.xsd /*[local-name()='schema']/*[local-name()='complexType' and @name='ProcessIdentifierType']/xs:simpleContent/xs:extension/xs:attribute	May not exceed 25 characters. Must match the two following pattern: [a-zA-Z0-9]+-[a-zA-Z0-9]+-[a-zA-Z0-9] Which defines the following parts: <domain>-<identifier Area>-<identifier type>
ServiceMetadataPublisherID	Unique identifier of the SMP	xs:string ServiceMetadataLocatorTypes-1.0.xsd /*[local-name()='schema']/*[local-name()='complexType' and @name='ServiceMetadataPublisherServiceForParticipantType']//*[local-name()='sequence']/*[local-name()='element' and @ref='ServiceMetadataPublisherID']	In ManageServiceMetadata service, this establishes the link with the managing SMP of the participant as defined by in the ManageBusinessIdentifier service.
NextPageIdentifier	This argument is not used in this context.	n/a	Must be null

Table 10 – CreateList() input

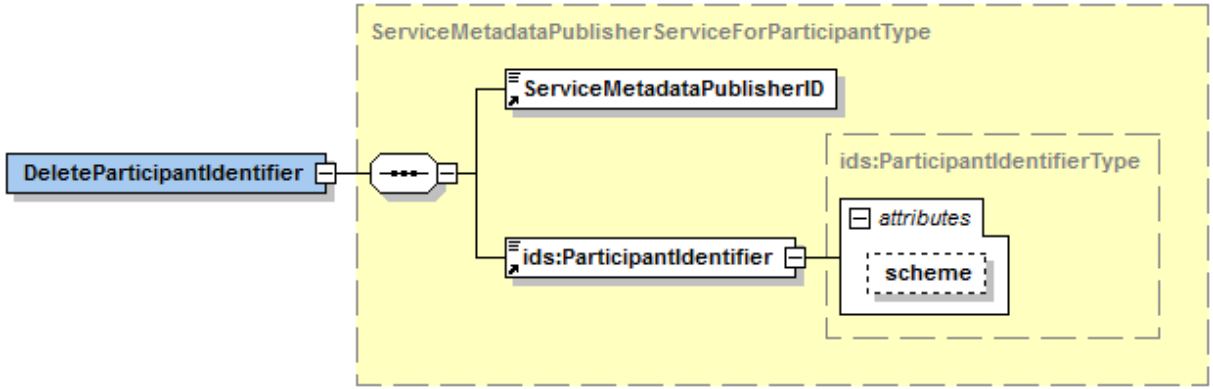
4.2.2.1.6. CreateList() Output

None.

4.2.2.1.7. Delete() Signature

Input: tns:deleteIn
msg:
messagePart (E lrs:DeleteParticipantIdentifier)
Output: tns:deleteOut
msg:
Fault: tns:notFoundFault
name: NotFoundFault
Fault: tns:unauthorizedFault
name: UnauthorizedFault
Fault: tns:internalErrorFault
name: InternalErrorFault
Fault: tns:badRequestFault
name: BadRequestFault

4.2.2.1.8. Delete() Input



argument	Description	Format/XSD/Xpath	Constraint
ServiceMetadataPublisherID	Unique identifier of the SMP	xs:string ServiceMetadataLocatorTypes-1.0.xsd /*[local-name()='schema']/*[local-name()='complexType' and @name='ServiceMetadataPublisherServiceForParticipantType']//*[local-name()='sequence']/*[local-name()='element' and @ref='ServiceMetadataPublisherID']	In ManageServiceMetadata service, this establishes the link with the managing SMP of the participant as defined by in the ManageBusinessIdentifier service.

argument	Description	Format/XSD/Xpath	Constraint
ParticipantIdentifier	Business unique identifier of the Participant. Represents a business level endpoint key that uniquely identifies an end-user entity in the network. Examples of identifiers are company registration and VAT numbers, DUNS numbers, GLN numbers, email addresses, etc…	The format must comply with ISO 15459 constraints as defined in [REF6]. Example: 0088:4035811991014 Identifiers-1.0.xsd /*[local-name()='schema']/*[local-name()='complexType' and @name='ParticipantIdentifierType']	Must be unique. May not exceed 50 characters and must be at least 1 character long. May only contain ASCII characters. Participant identifier must be trimmed. If the scheme refers to the (default) PEPPOL participant identifier scheme (iso6523-actorid-upis) and the participant identifier is not a wildcard ('*'), then the issuing agency specified in the ParticipantIdentifier (as its leading part before ':') must be included in the official list (cf. §0 - "List of valid issuing agencies for PEPPOL") Remark: the participant identifier is case insensitive.

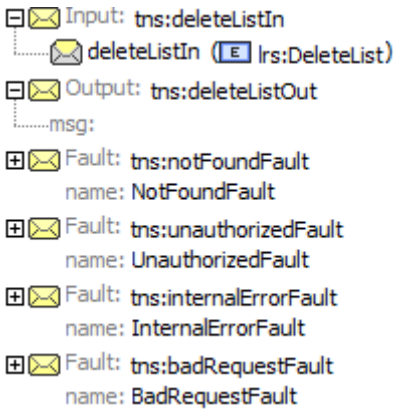
argument	Description	Format/XSD/Xpath	Constraint
ParticipantIdentifier.scheme	The scheme of the participant identifier	Identifier schemes for all schemed identifier types (participants, documents, profiles, transports) may be defined outside of this specification. Any instance of a 4-cornered infrastructure may choose to define identifier schemes that match the type of documents, participants or profiles that are relevant to support in that instance. Examples: iso6523-actorid-upis, busdox-actorid-upis Identifiers-1.0.xsd /*[local-name()='schema']/*[local-name()='complexType' and @name='ProcessIdentifierType']/xs:simpleContent/xs:extension/xs:attribute	May not exceed 25 characters. Must match the two following pattern: [a-zA-Z0-9]+-[a-zA-Z0-9]+-[a-zA-Z0-9] Which defines the following parts: <domain>-<identifier Area>-<identifier type>

Table 11 – Delete() input

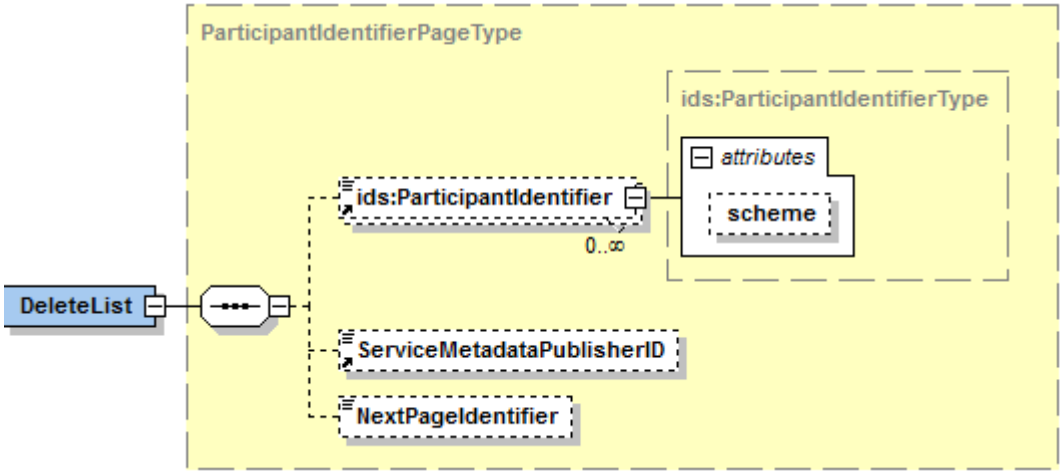
4.2.2.1.9. Delete() Output

None

4.2.2.1.10. DeleteList() Signature



4.2.2.1.11. DeleteList() Input



Note: the NextPageIndentifier is absent.

argument	Description	Format/XSD/Xpath	Constraint
ParticipantIdentifier (0..n)	Business unique identifier of the Participant. Represents a business level endpoint key that uniquely identifies an end-user entity in the network. Examples of identifiers are company registration and VAT numbers, DUNS numbers, GLN numbers, email addresses, etc…	The format must comply with ISO 15459 constraints as defined in [REF6]. Example: 0088:4035811991014 Identifiers-1.0.xsd /*[local-name()='schema']/*[local-name()='complexType' and @name='ParticipantIdentifierType']	Must be unique. May not exceed 50 characters and must be at least 1 character long. May only contain ASCII characters. Participant identifier must be trimmed. If the scheme refers to the (default) PEPPOL participant identifier scheme (iso6523-actorid-upis) and the participant identifier is not a wildcard ('*'), then the issuing agency specified in the ParticipantIdentifier (as its leading part before ':') must be included in the official list (cf. §0 - "List of valid issuing agencies for PEPPOL") Remark: the participant identifier is case insensitive.

argument	Description	Format/XSD/Xpath	Constraint
ParticipantIdentifier.scheme (0..n)	The scheme of the participant identifier	Identifier schemes for all schemed identifier types (participants, documents, profiles, transports) may be defined outside of this specification. Any instance of a 4-cornered infrastructure may choose to define identifier schemes that match the type of documents, participants or profiles that are relevant to support in that instance. Examples: iso6523-actorid-upis, busdox-actorid-upis Identifiers-1.0.xsd /*[local-name()='schema']/*[local-name()='complexType' and @name='ProcessIdentifierType']/xs:simpleContent/xs:extension/xs:attribute	May not exceed 25 characters. Must match the two following pattern: [a-zA-Z0-9]+-[a-zA-Z0-9]+-[a-zA-Z0-9] Which defines the following parts: <domain>-<identifier Area>-<identifier type>
ServiceMetadataPublisherID	Unique identifier of the SMP	xs:string ServiceMetadataLocatorTypes-1.0.xsd /*[local-name()='schema']/*[local-name()='complexType' and @name='ServiceMetadataPublisherServiceForParticipantType']//*[local-name()='sequence']/*[local-name()='element' and @ref='ServiceMetadataPublisherID']	In ManageServiceMetadata service, this establishes the link with the managing SMP of the participant as defined by in the ManageBusinessIdentifier service.
NextPageIdentifier	This argument is not used in this context.	n/a	Must be null

Table 12 – DeleteList() input

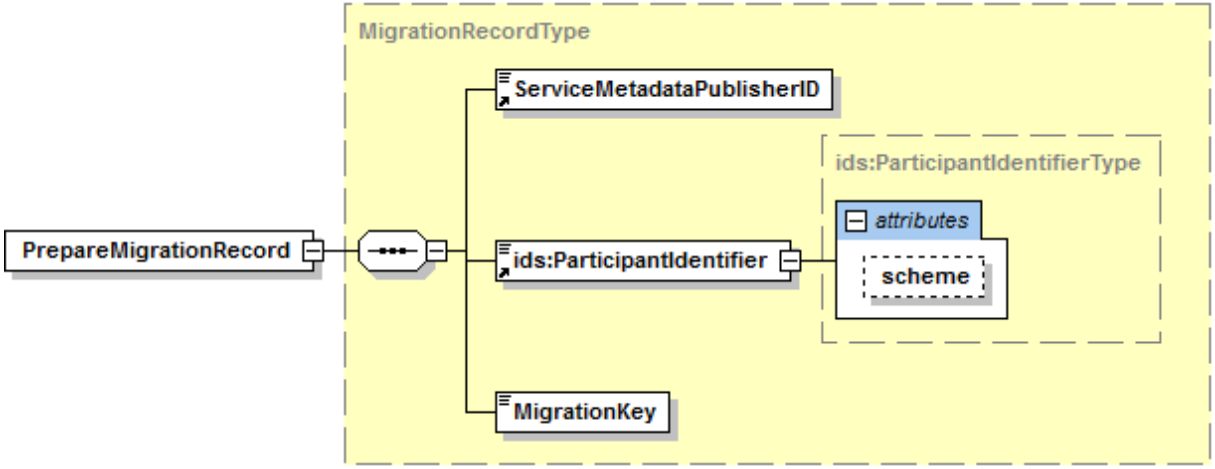
4.2.2.1.12. DeleteList() Output

None

4.2.2.1.13. PrepareToMigrate() Signature

Input: tns:prepareMigrateIn
prepareMigrateIn (E lrs:PrepareMigrationRecord)
Output: tns:prepareMigrateOut
msg:
Fault: tns:notFoundFault
name: NotFoundFault
Fault: tns:unauthorizedFault
name: UnauthorizedFault
Fault: tns:internalErrorFault
name: InternalErrorFault
Fault: tns:badRequestFault
name: BadRequestFault

4.2.2.1.14. PrepareToMigrate() Input



argument	Description	Format/XSD/Xpath	Constraint
ServiceMetadataPublisherID	Unique identifier of the SMP	xs:string ServiceMetadataLocatorTypes-1.0.xsd /*[local-name()='schema']/*[local-name()='complexType' and @name='ServiceMetadataPublisherServiceForParticipantType']//*[local-name()='sequence']/*[local-name()='element' and @ref='ServiceMetadataPublisherID']	In ManageServiceMetadata service, this establishes the link with the managing SMP of the participant as defined by in the ManageBusinessIdentifier service.

argument	Description	Format/XSD/Xpath	Constraint
ParticipantIdentifier	Business unique identifier of the Participant. Represents a business level endpoint key that uniquely identifies an end-user entity in the network. Examples of identifiers are company registration and VAT numbers, DUNS numbers, GLN numbers, email addresses, etc…	The format must comply with ISO 15459 constraints as defined in [REF6]. Example: 0088:4035811991014 Identifiers-1.0.xsd /*[local-name()='schema']/*[local-name()='complexType' and @name='ParticipantIdentifierType']	Must be unique. May not exceed 50 characters and must be at least 1 character long. May only contain ASCII characters. Participant identifier must be trimmed. If the scheme refers to the (default) PEPPOL participant identifier scheme (iso6523-actorid-upis) and the participant identifier is not a wildcard ('*'), then the issuing agency specified in the ParticipantIdentifier (as its leading part before ':') must be included in the official list (cf. §0 - "List of valid issuing agencies for PEPPOL") Remark: the participant identifier is case insensitive.

argument	Description	Format/XSD/Xpath	Constraint
ParticipantIdentifier.scheme	The scheme of the participant identifier	Identifier schemes for all schemed identifier types (participants, documents, profiles, transports) may be defined outside of this specification. Any instance of a 4-cornered infrastructure may choose to define identifier schemes that match the type of documents, participants or profiles that are relevant to support in that instance. Examples: iso6523-actorid-upis, busdox-actorid-upis Identifiers-1.0.xsd /*[local-name()='schema']/*[local-name()='complexType' and @name='ProcessIdentifierType']/xs:simpleContent/xs:extension/xs:attribute	May not exceed 25 characters. Must match the two following pattern: [a-zA-Z0-9]+-[a-zA-Z0-9]+-[a-zA-Z0-9] Which defines the following parts: <domain>-<identifier Area>-<identifier type>
MigrationKey	String which is a unique key controlling the migration of the metadata for a given ParticipantIdentifier from one Service Metadata Publisher to another.	xs:string ServiceMetadataLocatorTypes-1.0.xsd /*[local-name()='schema']/*[local-name()='complexType' and @name='MigrationRecordType']/*[local-name()='sequence']/*[local-name()='element' and @name='MigrationKey']	The migration key is a code that must be passed out-of-band to the SMP which is taking over the publishing of the metadata for the participant identifier. This code must be unique; This code must not be null nor empty; This code must contain: • 8 characters minimum • 24 characters maximum • 2 Special Characters @#$%()[]{}*^-!~|+= • 2 Upper Case letters minimum

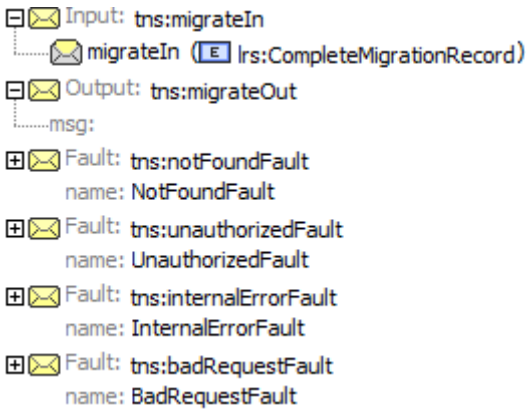
argument	Description	Format/XSD/Xpath	Constraint
			• 2 Lower Case letters minimum • 2 Numbers minimum • No white spaces

Table 13 – PrepareToMigrate() input

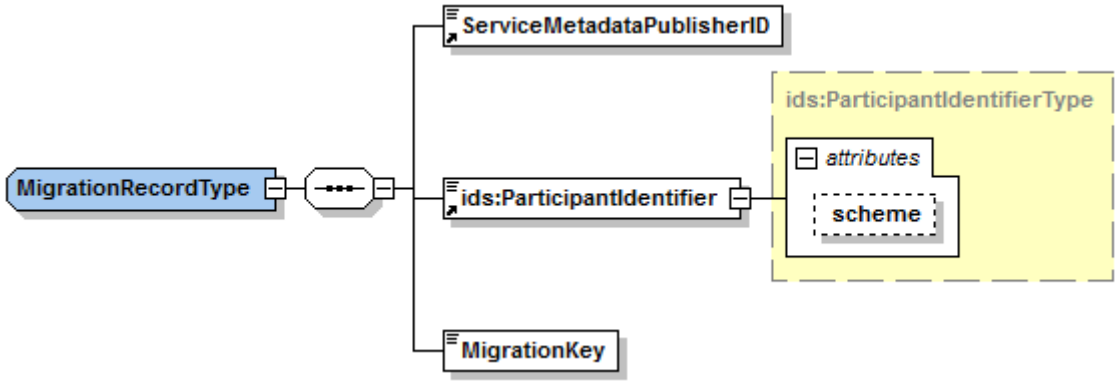
4.2.2.1.15. PrepareToMigrate() Output

None

4.2.2.1.16. Migrate() Signature



4.2.2.1.17. Migrate() Input



argument	Description	Format/XSD/Xpath	Constraint
ServiceMetadataPublisherID	Unique identifier of the SMP	xs:string ServiceMetadataLocatorTypes-1.0.xsd /*[local-name()='schema']/*[local-name()='complexType' and @name='ServiceMetadataPublisherServiceForParticipantType']//*[local-name()='sequence']/*[local-name()='element' and @ref='ServiceMetadataPublisherID']	In ManageServiceMetadata service, this establishes the link with the managing SMP of the participant as defined by in the ManageBusinessIdentifier service.

argument	Description	Format/XSD/Xpath	Constraint
ParticipantIdentifier	Business unique identifier of the Participant. Represents a business level endpoint key that uniquely identifies an end-user entity in the network. Examples of identifiers are company registration and VAT numbers, DUNS numbers, GLN numbers, email addresses, etc…	The format must comply with ISO 15459 constraints as defined in [REF6]. Example: 0088:4035811991014 Identifiers-1.0.xsd /*[local-name()='schema']/*[local-name()='complexType' and @name='ParticipantIdentifierType']	Must be unique. May not exceed 50 characters and must be at least 1 character long. May only contain ASCII characters. Participant identifier must be trimmed. If the scheme refers to the (default) PEPPOL participant identifier scheme (iso6523-actorid-upis) and the participant identifier is not a wildcard ('*'), then the issuing agency specified in the ParticipantIdentifier (as its leading part before ':') must be included in the official list (cf. §0 - "List of valid issuing agencies for PEPPOL") Remark: the participant identifier is case insensitive.

argument	Description	Format/XSD/Xpath	Constraint
ParticipantIdentifier.scheme	The scheme of the participant identifier	Identifier schemes for all schemed identifier types (participants, documents, profiles, transports) may be defined outside of this specification. Any instance of a 4-cornered infrastructure may choose to define identifier schemes that match the type of documents, participants or profiles that are relevant to support in that instance. Examples: iso6523-actorid-upis, busdox-actorid-upis Identifiers-1.0.xsd /*[local-name()='schema']/*[local-name()='complexType' and @name='ProcessIdentifierType']/xs:simpleContent/xs:extension/xs:attribute	May not exceed 25 characters. Must match the two following pattern: [a-zA-Z0-9]+-[a-zA-Z0-9]+-[a-zA-Z0-9] Which defines the following parts: <domain>-<identifier Area>-<identifier type>
MigrationKey	String which is a unique key controlling the migration of the metadata for a given ParticipantIdentifier from one Service Metadata Publisher to another.	xs:string ServiceMetadataLocatorTypes-1.0.xsd /*[local-name()='schema']/*[local-name()='complexType' and @name='MigrationRecordType']/*[local-name()='sequence']/*[local-name()='element' and @name='MigrationKey']	The migration key is a code that must be passed out-of-band to the SMP which is taking over the publishing of the metadata for the participant identifier. This code must be unique; This code must not be null nor empty; This code must contain: • 8 characters minimum • 24 characters maximum • 2 Special Characters @#$%()[]{}*^-!~|+= • 2 Upper Case letters minimum

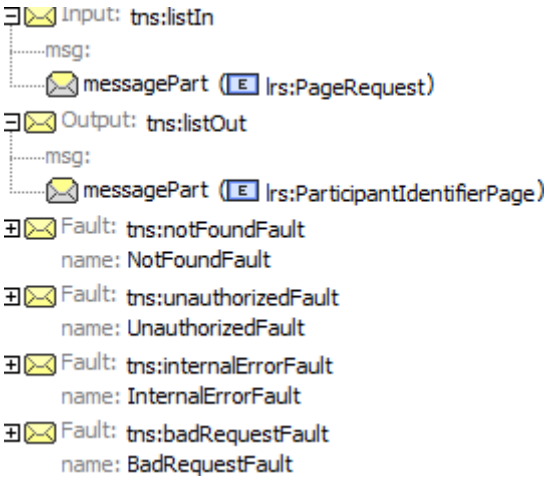
argument	Description	Format/XSD/Xpath	Constraint
			• 2 Lower Case letters minimum • 2 Numbers minimum • No white spaces

Table 14 – Migrate() input

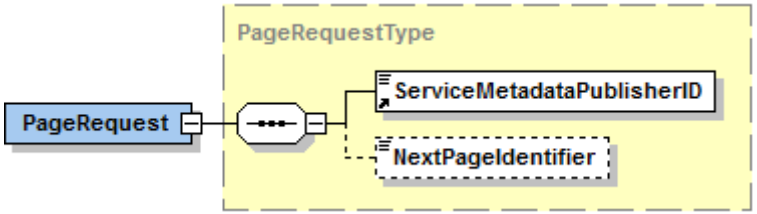
4.2.2.1.18. Migrate() Output

None

4.2.2.1.19. List() Signature



4.2.2.1.20. List() Input

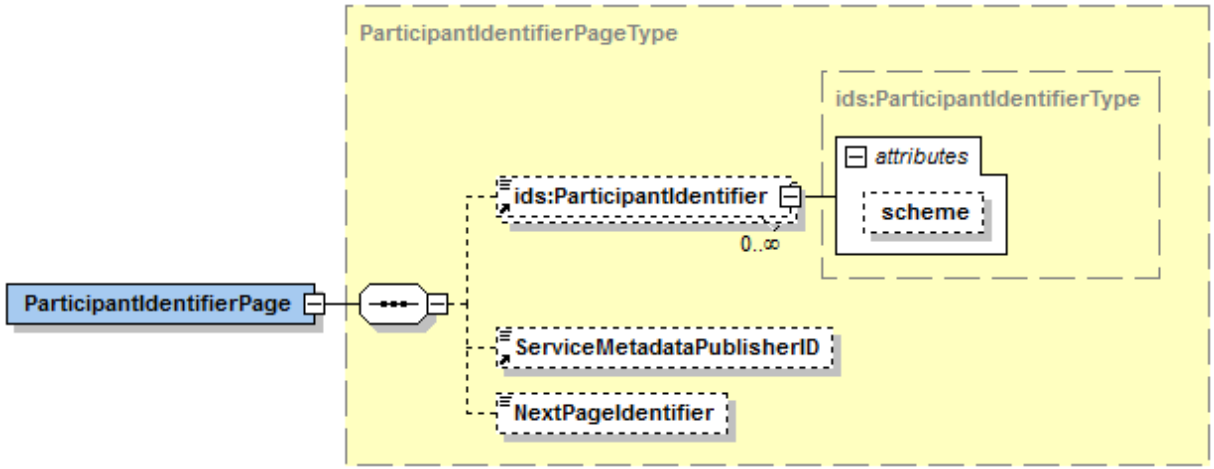


argument	Description	Format/XSD/Xpath	Constraint
ServiceMetadataPublisherID	Unique identifier of the SMP	xs:string ServiceMetadataLocatorTypes-1.0.xsd /*[local-name()='schema']/*[local-name()='complexType' and @name='ServiceMetadataPublisherServiceForParticipantType']//*[local-name()='sequence']/*[local-name()='element' and @ref='ServiceMetadataPublisherID']	In ManageServiceMetadata service, this establishes the link with the managing SMP of the participant as defined by in the ManageBusinessIdentifier service.
NextPageIdentifier	Identifier that controls the navigation between pages of a long list. As Input parameter this identifier represents the page of data to retrieve. If the NextPageIdentifier is absent, the first page is returned. As Output parameter, this	xs:string ServiceMetadataLocatorTypes-1.0.xsd /*[local-name()='schema']/*[local-name()='complexType' and @name='PageRequestType']/*[local-name()='sequence']/*[local-name()='element' and @name='NextPageIdentifier']	Must be a positive number. Must be null or empty for create or update operations (since the full list of values has to be provided at once).

argument	Description	Format/XSD/Xpath	Constraint
	value can be used as() Input at the next call to the same operation to retrieve the next page of data (to navigate forward). This parameter is used only for 'read' operations returning list of values.		

Table 15 – List() input

4.2.2.1.21. List() Output



argument	Description	Format/XSD/Xpath	Constraint
ParticipantIdentifier (0..n)	Business unique identifier of the Participant. Represents a business level endpoint key that uniquely identifies an end-user entity in the network. Examples of identifiers are company registration and VAT numbers, DUNS numbers, GLN numbers, email addresses, etc…	The format must comply with ISO 15459 constraints as defined in [REF6]. Example: 0088:4035811991014 Identifiers-1.0.xsd /*[local-name()='schema']/*[local-name()='complexType' and @name='ParticipantIdentifierType']	Must be unique. May not exceed 50 characters and must be at least 1 character long. May only contain ASCII characters. Participant identifier must be trimmed. If the scheme refers to the (default) PEPPOL participant identifier scheme (iso6523-actorid-upis) and the participant identifier is not a wildcard ('*'), then the issuing agency specified in the ParticipantIdentifier (as its leading part before ':') must be included in the official list (cf. §0 - "List of valid issuing agencies for PEPPOL") Remark: the participant identifier is case insensitive.

argument	Description	Format/XSD/Xpath	Constraint
ParticipantIdentifier.scheme (0..n)	The scheme of the participant identifier	Identifier schemes for all schemed identifier types (participants, documents, profiles, transports) may be defined outside of this specification. Any instance of a 4-cornered infrastructure may choose to define identifier schemes that match the type of documents, participants or profiles that are relevant to support in that instance. Examples: iso6523-actorid-upis, busdox-actorid-upis Identifiers-1.0.xsd /*[local-name()='schema']/*[local-name()='complexType' and @name='ProcessIdentifierType']/xs:simpleContent/xs:extension/xs:attribute	May not exceed 25 characters. Must match the two following pattern: [a-zA-Z0-9]+-[a-zA-Z0-9]+-[a-zA-Z0-9] Which defines the following parts: <domain>-<identifier Area>-<identifier type>
ServiceMetadataPublisherID	Unique identifier of the SMP	xs:string ServiceMetadataLocatorTypes-1.0.xsd /*[local-name()='schema']/*[local-name()='complexType' and @name='ServiceMetadataPublisherServiceForParticipantType']//*[local-name()='sequence']/*[local-name()='element' and @ref='ServiceMetadataPublisherID']	In ManageServiceMetadata service, this establishes the link with the managing SMP of the participant as defined by in the ManageBusinessIdentifier service.

argument	Description	Format/XSD/Xpath	Constraint
NextPageIdentifier	Identifier that controls the navigation between pages of a long list. As Input parameter this identifier represents the page of data to retrieve. If the NextPageIdentifier is absent, the first page is returned. As Output parameter, this value can be used as() Input at the next call to the same operation to retrieve the next page of data (to navigate forward). This parameter is used only for 'read' operations returning list of values.	xs:string ServiceMetadataLocatorTypes-1.0.xsd /*[local-name()='schema']/*[local-name()='complexType' and @name='PageRequestType']/*[local-name()='sequence']/*[local-name()='element' and @name='NextPageIdentifier']	Must be a positive number. Must be null or empty for create or update operations (since the full list of values has to be provided at once).

Table 16 – List() output

4.2.3. WSDL model for BDMSLService

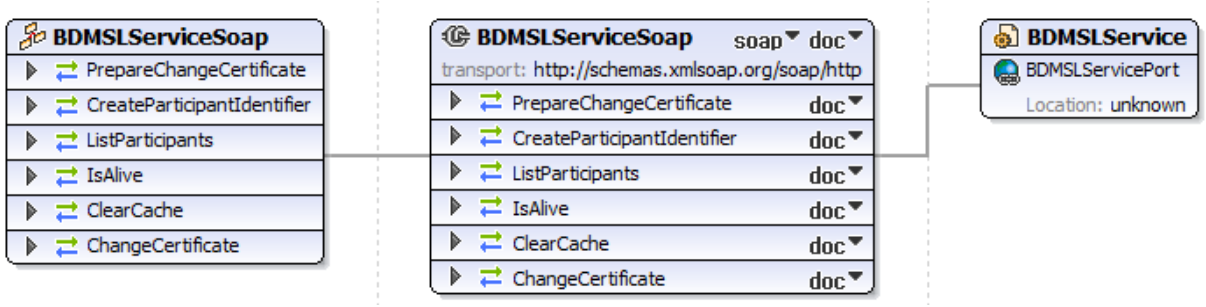


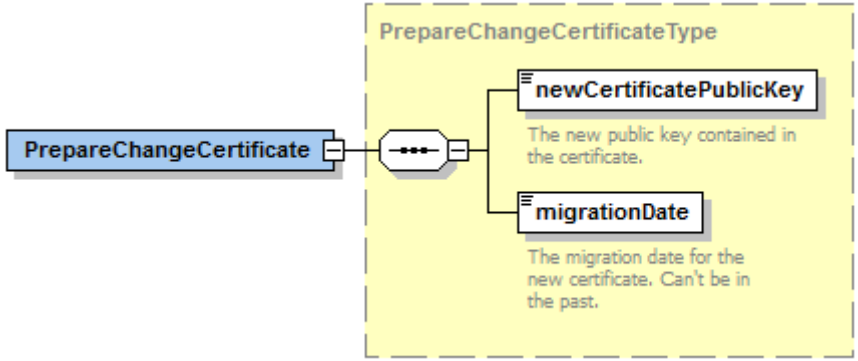
Figure 8 – WSDL BDMSLService

4.2.3.1. Operations Signatures

4.2.3.1.1. PrepareChangeCertificate() Signature

Input: tns:prepareChangeCertificateIn
msg:
messagePart (E lrs:PrepareChangeCertificate)
Output: tns:prepareChangeCertificateOut
msg:
Fault: tns:notFoundFault
name: NotFoundFault
Fault: tns:unauthorizedFault
name: UnauthorizedFault
Fault: tns:internalErrorFault
name: InternalErrorFault
Fault: tns:badRequestFault
name: BadRequestFault

4.2.3.1.2. PrepareChangeCertificate() Input



argument	Description	Format/XSD/Xpath	Constraint
newCertificatePublicKey	The new public key contained in the certificate	base64Binary BDMSLService-1.0.xsd /*[local-name()='schema']/*[local-name()='complexType' and @name='PrepareChangeCertificateType']/*[local-name()='sequence']/*[local-name()='element' and @name='newCertificatePublicKey']	Must be valid and belong to the list of authorized root certificate aliases.

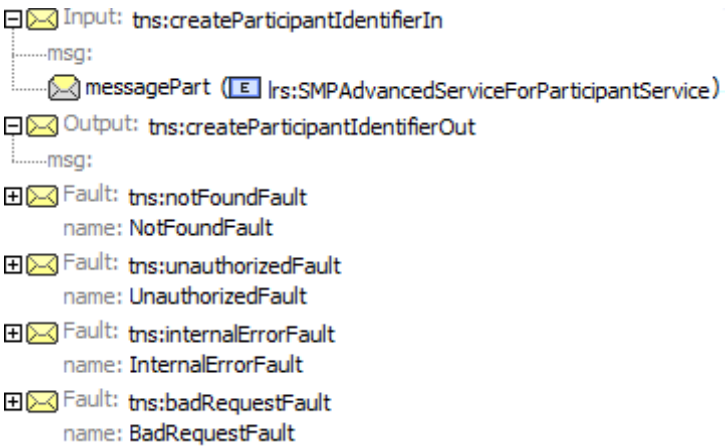
argument	Description	Format/XSD/Xpath	Constraint
migrationDate	The migration date for the new certificate	xs:date BDMSLService-1.0.xsd /*[local-name()='schema']/*[local-name()='complexType' and @name='PrepareChangeCertificateType']/*[local-name()='sequence']/*[local-name()='element' and @name='migrationDate']	May not be in the past. Must be in the validity period of the related new certificate (i.e. within NotBeforeCertificateDate and NotAfterCertificateDate attribute of the certificate). If migrationDate is empty, then the "Valid From" date is extracted from the certificate and is used as the migrationDate.

Table 17 – PrepareChangeCertificate() input

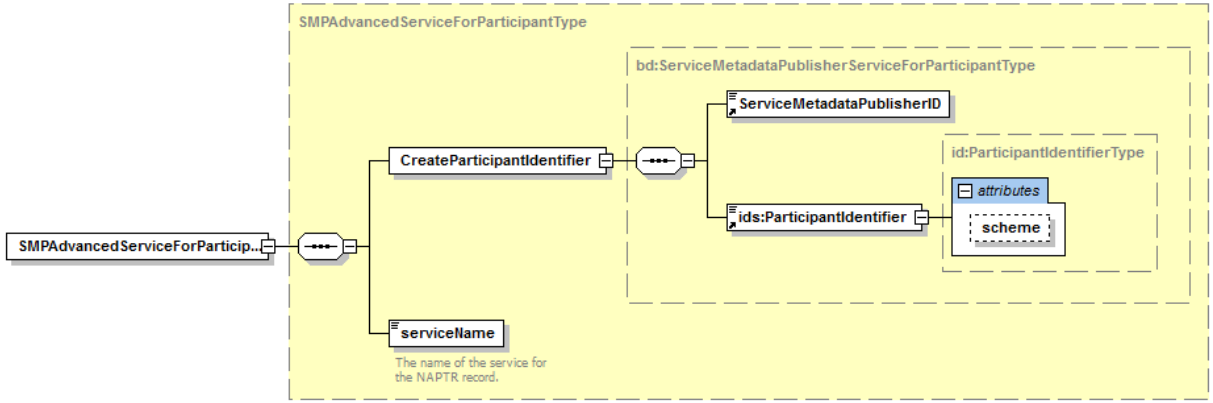
4.2.3.1.3. PrepareChangeCertificate() Output

None

4.2.3.1.4. CreateParticipantIdentifier() Signature



4.2.3.1.5. CreateParticipantIdentifier() Input



argument	Description	Format/XSD/Xpath	Constraint
ServiceMetadataPublisherID	Unique identifier of the SMP	xs:string ServiceMetadataLocatorTypes-1.0.xsd /*[local-name()='schema']/*[local-name()='complexType' and @name='ServiceMetadataPublisherServiceForParticipantType']//*[local-name()='sequence']/*[local-name()='element' and @ref='ServiceMetadataPublisherID']	In ManageServiceMetadata service, this establishes the link with the managing SMP of the participant as defined by in the ManageBusinessIdentifier service.

argument	Description	Format/XSD/Xpath	Constraint
ParticipantIdentifier	Business unique identifier of the Participant. Represents a business level endpoint key that uniquely identifies an end-user entity in the network. Examples of identifiers are company registration and VAT numbers, DUNS numbers, GLN numbers, email addresses, etc…	The format must comply with ISO 15459 constraints as defined in [REF6]. Example: 0088:4035811991014 Identifiers-1.0.xsd /*[local-name()='schema']/*[local-name()='complexType' and @name='ParticipantIdentifierType']	Must be unique. May not exceed 50 characters and must be at least 1 character long. May only contain ASCII characters. Participant identifier must be trimmed. If the scheme refers to the (default) PEPPOL participant identifier scheme (iso6523-actorid-upis) and the participant identifier is not a wildcard ('*'), then the issuing agency specified in the ParticipantIdentifier (as its leading part before ':') must be included in the official list (cf. §0 - "List of valid issuing agencies for PEPPOL") Remark: the participant identifier is case insensitive.

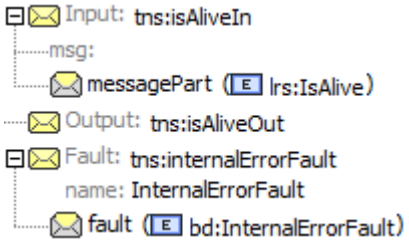
argument	Description	Format/XSD/Xpath	Constraint
ParticipantIdentifier.scheme	The scheme of the participant identifier	Identifier schemes for all schemed identifier types (participants, documents, profiles, transports) may be defined outside of this specification. Any instance of a 4-cornered infrastructure may choose to define identifier schemes that match the type of documents, participants or profiles that are relevant to support in that instance. Examples: iso6523-actorid-upis, busdox-actorid-upis Identifiers-1.0.xsd /*[local-name()='schema']/*[local-name()='complexType' and @name='ProcessIdentifierType']/xs:simpleContent/xs:extension/xs:attribute	May not exceed 25 characters. Must match the two following pattern: [a-zA-Z0-9]+-[a-zA-Z0-9]+-[a-zA-Z0-9] Which defines the following parts: <domain>-<identifier Area>-<identifier type>
serviceName	The name of the service for the NAPTR record.	xs:string BDMSLService-1.0.xsd /*[local-name()='schema']/*[local-name()='complexType' and @name='SMPAdvancedServiceForParticipantType']/*[local-name()='sequence']/*[local-name()='element' and @name='serviceName']	No constraint is enforced by the BDMSL

Table 18 – CreateParticipantIdentifier() input

4.2.3.1.6. CreateParticipantIdentifier() Output

None

4.2.3.1.7. IsAlive() Signature



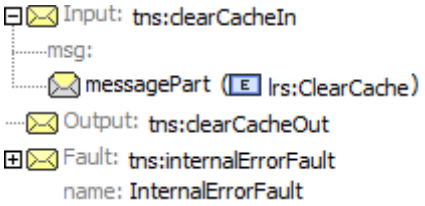
4.2.3.1.8. IsAlive() Input

(empty)

4.2.3.1.9. IsAlive() Output

None

4.2.3.1.10. ClearCache() Signature



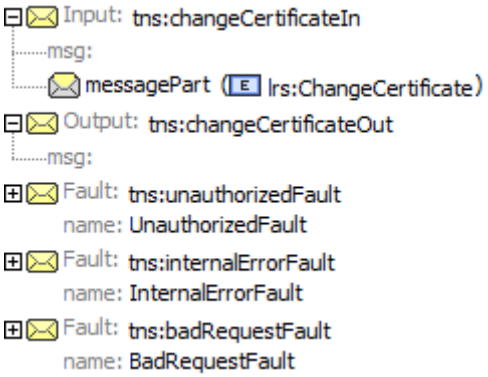
4.2.3.1.11. ClearCache() Input

none

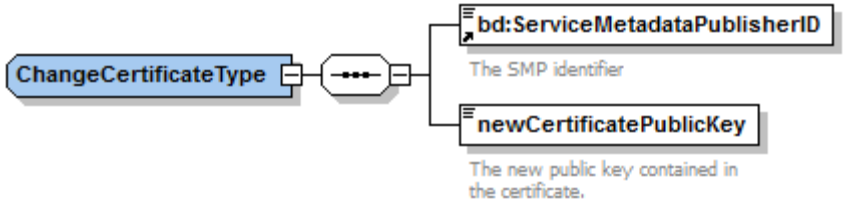
4.2.3.1.12. ClearCache() Output

none

4.2.3.1.13. ChangeCertificate() Signature



4.2.3.1.14. ChangeCertificate() Input



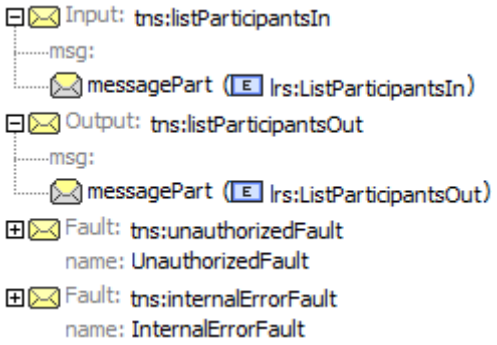
argument	Description	Format/XSD/Xpath	Constraint
ServiceMetadataPublisherID	Unique identifier of the SMP	xs:string ServiceMetadataLocatorTypes-1.0.xsd /*[local-name()='schema']/*[local-name()='complexType' and @name='ServiceMetadataPublisherServiceForParticipantType']//*[local-name()='sequence']/*[local-name()='element' and @ref='ServiceMetadataPublisherID']	In ManageServiceMetadata service, this establishes the link with the managing SMP of the participant as defined by in the ManageBusinessIdentifier service.
newCertificatePublicKey	The new public key contained in the certificate	base64Binary BDMSLService-1.0.xsd /*[local-name()='schema']/*[local-name()='complexType' and @name='PrepareChangeCertificateType']/*[local-name()='sequence']/*[local-name()='element' and @name='newCertificatePublicKey']	Must be valid and belong to the list of authorized root certificate aliases.

Table 19 – ChangeCertificate() input

4.2.3.1.15. ChangeCertificate() Output

None

4.2.3.1.16. ListParticipants() Signature

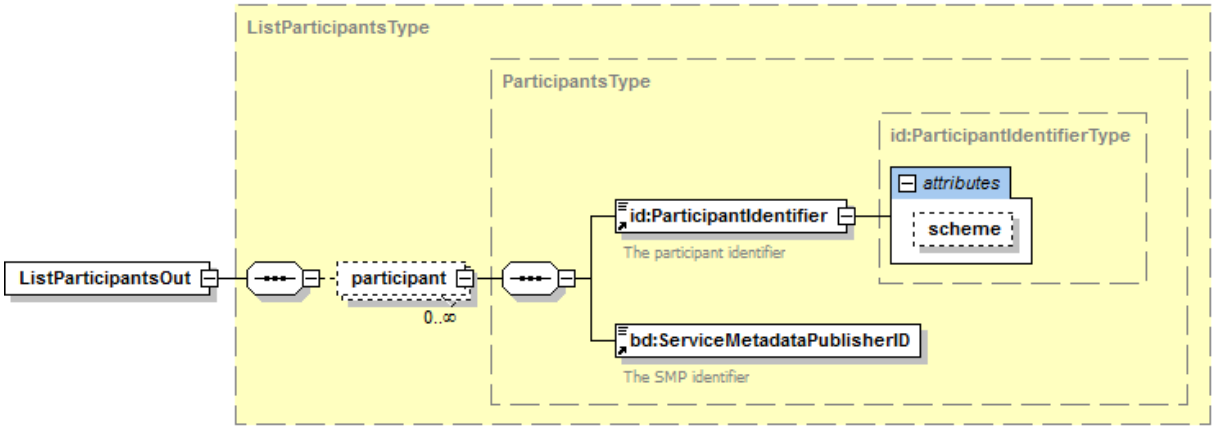


4.2.3.1.17. ListParticipants() Input

ListParticipantsIn

(empty)

4.2.3.1.18. ListParticipants() Output



argument	Description	Format/XSD/Xpath	Constraint
ParticipantIdentifier (0..n)	Business unique identifier of the Participant. Represents a business level endpoint key that uniquely identifies an end-user entity in the network. Examples of identifiers are company registration and VAT numbers, DUNS numbers, GLN numbers, email addresses, etc…	The format must comply with ISO 15459 constraints as defined in [REF6]. Example: 0088:4035811991014 Identifiers-1.0.xsd /*[local-name()='schema']/*[local-name()='complexType' and @name='ParticipantIdentifierType']	Must be unique. May not exceed 50 characters and must be at least 1 character long. May only contain ASCII characters. Participant identifier must be trimmed. If the scheme refers to the (default) PEPPOL participant identifier scheme (iso6523-actorid-upis) and the participant identifier is not a wildcard ('*'), then the issuing agency specified in the ParticipantIdentifier (as its leading part before ':') must be included in the official list (cf. §0 - "List of valid issuing agencies for PEPPOL") Remark: the participant identifier is case insensitive.

argument	Description	Format/XSD/Xpath	Constraint
ParticipantIdentifier.scheme (0..n)	The scheme of the participant identifier	Identifier schemes for all schemed identifier types (participants, documents, profiles, transports) may be defined outside of this specification. Any instance of a 4-cornered infrastructure may choose to define identifier schemes that match the type of documents, participants or profiles that are relevant to support in that instance. Examples: iso6523-actorid-upis, busdox-actorid-upis Identifiers-1.0.xsd /*[local-name()='schema']/*[local-name()='complexType' and @name='ProcessIdentifierType']/xs:simpleContent/xs:extension/xs:attribute	May not exceed 25 characters. Must match the two following pattern: [a-zA-Z0-9]+-[a-zA-Z0-9]+-[a-zA-Z0-9] Which defines the following parts: <domain>-<identifier Area>-<identifier type>
ServiceMetadataPublisherID	Unique identifier of the SMP	xs:string ServiceMetadataLocatorTypes-1.0.xsd /*[local-name()='schema']/*[local-name()='complexType' and @name='ServiceMetadataPublisherServiceForParticipantType']//*[local-name()='sequence']/*[local-name()='element' and @ref='ServiceMetadataPublisherID']	In ManageServiceMetadata service, this establishes the link with the managing SMP of the participant as defined by in the ManageBusinessIdentifier service.

Table 20 – ListParticipants() input

4.2.4. Faults

4.2.4.1. Faults generic specifications

All operations above may return all or some of the possible following faults:

- notFoundFault: the target element(s) (MetadataPublisher, Participant …) on which the operation must be performed is not present into the configuration.
- unauthorizedFault: the user does not have the permission to execute that operation
- badRequestFault: the structure of the request is not well-formed
- internaltFault: any other error occurred during the processing of the request

4.2.4.2. Error Codes

Whenever a fault occurs, more details on the source of the error will provided in the SOAP fault with the applicable error code as listed in the table below:

Error code	Description
100	SMP not found error
101	Unauthorized error
102	Certificate authentication issue
103	The root alias is not found in the list of trusted issuers in the database
104	The certificate is revoked
105	Generic technical error
106	Bad request error
107	DNS communication problem
108	Problem with the SIG0 Signature
109	Bad configuration
110	Participant not found error
111	Migration data not found
112	Duplicate participant error
113	Error when deleting a SMP
114	The deletion failed because a migration is planned for the given participant or SMP

Table 21 – Error Codes

4.2.4.3. Faults specific usages

The table below shows the applicability of these errors for all operations specified in this document:

	ManageServiceMetadata				ManageBusinessIdentifier							BDMSLService					
	Create	Read	Update	Delete	Create	CreateList	Delete	DeleteList	PrepareToMigrate	Migrate	List	PrepareChangeCertificate	CreateParticipantIdentifier	IsAlive	ClearCache	ChangeCertificate	ListParticipants
notFoundFault:		X	X	X	X	X	X	X	X	X	X	X	X			X	
unauthorizedFault:	X	X	X	X	X	X	X	X	X	X	X	X	X			X	X
internalErrorFault:	X	X	X	X	X	X	X	X	X	X	X	X	X		X	X	X
badRequestFault:	X	X	X	X	X	X	X	X	X	X	X	X	X			X	

Table 22 – Fault specific usage

4.2.4.4. Sample SOAP errors

The following are SOAP faults samples as they are returned to the requester in case of error encountered by the BDMSL.

Sample "NotFoundFault":

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <soap:Fault>
      <faultcode>soap:Server</faultcode>
      <faultstring>[ERR-100] The SMP 'testSMPPrepareToMigrate520' doesn't exist.
[14ojYP8Op4vWs78XmcVtLBVFsbPat1mOE9h8HChQW0O48XbOOZXu!-
2114654990!1469472374542]</faultstring>
      <detail>
        <NotFoundFault xmlns:ns2="http://busdox.org/transport/identifiers/1.0/"
xmlns="http://busdox.org/serviceMetadata/locator/1.0/">
          <FaultMessage>[ERR-100] The SMP 'testSMPPrepareToMigrate520' doesn't exist.</FaultMessage>
        </NotFoundFault>
      </detail>
    </soap:Fault>
  </soap:Body>
</soap:Envelope>
```

Sample "BadRequestFault":

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <soap:Fault>
      <faultcode>soap:Server</faultcode>
      <faultstring[ERR-106] Participant Identifier value "9999" is illegal .
[NtsjYc25LzhHUNmQ4_Z6Bv8En5sB2e9WjFhMZrBTUcLHT5QlsR99!-
2114654990!1469472427449]</faultstring>
      <detail>
        <BadRequestFault xmlns:ns2="http://busdox.org/transport/identifiers/1.0/"
xmlns="http://busdox.org/serviceMetadata/locator/1.0/">
          <FaultMessage>[ERR-106] Participant Identifier Value contains the illegal issuing agency
'0185'</FaultMessage>
        </BadRequestFault>
      </detail>
    </soap:Fault>
  </soap:Body>
</soap:Envelope>
```

5. ANNEXES

5.1. Interface Message standards

The documents listed in this paragraph can all be downloaded from eDelivery repository at this location:

https://ec.europa.eu/cefdigital/code/projects/EDELIVERY/repos/bdmsl/browse/bdmsl-webapp/src/main/webapp/WEB-INF/wsdl/

5.1.1. WSDL's

5.1.1.1. ManageBusinessIdentifierService

```
<!--
(C) Copyright 2018 - European Commission | CEF eDelivery

Licensed under the EUPL, Version 1.2 (the "License");
You may not use this file except in compliance with the License.
You may obtain a copy of the License at

https://joinup.ec.europa.eu/sites/default/files/custom-page/attachment/eupl_v1.2_en.pdf

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
-->

<wsdl:definitions xmlns:tns="http://busdox.org/serviceMetadata/ManageBusinessIdentifierService/1.0/"
xmlns:soap11="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:lrs="http://busdox.org/serviceMetadata/locator/1.0/" xmlns:s="http://www.w3.org/2001/XMLSchema"
xmlns:http="http://schemas.xmlsoap.org/wsdl/http/" name="ManageBusinessIdentifierService"
targetNamespace="http://busdox.org/serviceMetadata/ManageBusinessIdentifierService/1.0/"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
    <wsdl:documentation xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"/>
    <wsdl:types>
        <s:schema elementFormDefault="qualified"
targetNamespace="http://busdox.org/serviceMetadata/ManageBusinessIdentifierService/1.0/Schema/">
            <s:import namespace="http://busdox.org/serviceMetadata/locator/1.0/" schemaLocation="ServiceMetadataLocatorTypes-
1.0.xsd"/>
        </s:schema>
    </wsdl:types>
    <wsdl:message name="createIn">
        <wsdl:documentation xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"/>
        <wsdl:part name="messagePart" element="lrs:CreateParticipantIdentifier"/>
    </wsdl:message>
    <wsdl:message name="createOut">
        <wsdl:documentation xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"/>
    </wsdl:message>
    <wsdl:message name="deleteIn">
        <wsdl:documentation xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"/>
        <wsdl:part name="messagePart" element="lrs:DeleteParticipantIdentifier"/>
    </wsdl:message>
    <wsdl:message name="deleteOut">
        <wsdl:documentation xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"/>
    </wsdl:message>
    <wsdl:message name="listIn">
        <wsdl:documentation xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"/>
        <wsdl:part name="messagePart" element="lrs:PageRequest"/>
    </wsdl:message>
    <wsdl:message name="listOut">
        <wsdl:documentation xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"/>
```

```
        <wsdl:part name="messagePart" element="lrs:ParticipantIdentifierPage"/>
    </wsdl:message>
    <wsdl:message name="prepareMigrateIn">
        <wsdl:part name="prepareMigrateIn" element="lrs:PrepareMigrationRecord"/>
    </wsdl:message>
    <wsdl:message name="prepareMigrateOut">
        <wsdl:documentation xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"/>
    </wsdl:message>
    <wsdl:message name="migrateIn">
        <wsdl:part name="migrateIn" element="lrs:CompleteMigrationRecord"/>
    </wsdl:message>
    <wsdl:message name="migrateOut">
        <wsdl:documentation xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"/>
    </wsdl:message>
    <wsdl:message name="createListIn">
        <wsdl:part name="createListIn" element="lrs:CreateList"/>
    </wsdl:message>
    <wsdl:message name="createListOut">
        <wsdl:documentation xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"/>
    </wsdl:message>
    <wsdl:message name="deleteListIn">
        <wsdl:part name="deleteListIn" element="lrs:DeleteList"/>
    </wsdl:message>
    <wsdl:message name="deleteListOut">
        <wsdl:documentation xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"/>
    </wsdl:message>
    <wsdl:message name="badRequestFault">
        <wsdl:part name="fault" element="lrs:BadRequestFault"/>
    </wsdl:message>
    <wsdl:message name="internalErrorFault">
        <wsdl:part name="fault" element="lrs:InternalErrorFault"/>
    </wsdl:message>
    <wsdl:message name="notFoundFault">
        <wsdl:part name="fault" element="lrs:NotFoundFault"/>
    </wsdl:message>
    <wsdl:message name="unauthorizedFault">
        <wsdl:part name="fault" element="lrs:UnauthorizedFault"/>
    </wsdl:message>
    <wsdl:portType name="ManageBusinessIdentifierServiceSoap">
        <wsdl:documentation xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"/>
        <wsdl:operation name="Create">
            <wsdl:documentation xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"/>
            <wsdl:input message="tns:createIn"/>
            <wsdl:output message="tns:createOut"/>
            <wsdl:fault message="tns:notFoundFault" name="NotFoundFault"/>
            <wsdl:fault message="tns:unauthorizedFault" name="UnauthorizedFault"/>
            <wsdl:fault message="tns:internalErrorFault" name="InternalErrorFault"/>
            <wsdl:fault message="tns:badRequestFault" name="BadRequestFault"/>
        </wsdl:operation>
        <wsdl:operation name="Delete">
            <wsdl:documentation xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"/>
            <wsdl:input message="tns:deleteIn"/>
            <wsdl:output message="tns:deleteOut"/>
            <wsdl:fault message="tns:notFoundFault" name="NotFoundFault"/>
            <wsdl:fault message="tns:unauthorizedFault" name="UnauthorizedFault"/>
            <wsdl:fault message="tns:internalErrorFault" name="InternalErrorFault"/>
            <wsdl:fault message="tns:badRequestFault" name="BadRequestFault"/>
        </wsdl:operation>
        <wsdl:operation name="List">
            <wsdl:documentation xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"/>
            <wsdl:input message="tns:listIn"/>
            <wsdl:output message="tns:listOut"/>
            <wsdl:fault message="tns:notFoundFault" name="NotFoundFault"/>
            <wsdl:fault message="tns:unauthorizedFault" name="UnauthorizedFault"/>
            <wsdl:fault message="tns:internalErrorFault" name="InternalErrorFault"/>
            <wsdl:fault message="tns:badRequestFault" name="BadRequestFault"/>
        </wsdl:operation>
        <wsdl:operation name="PrepareToMigrate">
            <wsdl:input message="tns:prepareMigrateIn"/>
            <wsdl:output message="tns:prepareMigrateOut"/>
            <wsdl:fault message="tns:notFoundFault" name="NotFoundFault"/>
            <wsdl:fault message="tns:unauthorizedFault" name="UnauthorizedFault"/>
```

```
            <wsdl:fault message="tns:internalErrorFault" name="InternalErrorFault"/>
            <wsdl:fault message="tns:badRequestFault" name="BadRequestFault"/>
        </wsdl:operation>
        <wsdl:operation name="Migrate">
            <wsdl:input message="tns:migrateIn"/>
            <wsdl:output message="tns:migrateOut"/>
            <wsdl:fault message="tns:notFoundFault" name="NotFoundFault"/>
            <wsdl:fault message="tns:unauthorizedFault" name="UnauthorizedFault"/>
            <wsdl:fault message="tns:internalErrorFault" name="InternalErrorFault"/>
            <wsdl:fault message="tns:badRequestFault" name="BadRequestFault"/>
        </wsdl:operation>
        <wsdl:operation name="CreateList">
            <wsdl:input message="tns:createListIn"/>
            <wsdl:output message="tns:createListOut"/>
            <wsdl:fault message="tns:notFoundFault" name="NotFoundFault"/>
            <wsdl:fault message="tns:unauthorizedFault" name="UnauthorizedFault"/>
            <wsdl:fault message="tns:internalErrorFault" name="InternalErrorFault"/>
            <wsdl:fault message="tns:badRequestFault" name="BadRequestFault"/>
        </wsdl:operation>
        <wsdl:operation name="DeleteList">
            <wsdl:input message="tns:deleteListIn"/>
            <wsdl:output message="tns:deleteListOut"/>
            <wsdl:fault message="tns:notFoundFault" name="NotFoundFault"/>
            <wsdl:fault message="tns:unauthorizedFault" name="UnauthorizedFault"/>
            <wsdl:fault message="tns:internalErrorFault" name="InternalErrorFault"/>
            <wsdl:fault message="tns:badRequestFault" name="BadRequestFault"/>
        </wsdl:operation>
    </wsdl:portType>
    <wsdl:binding name="ManageBusinessIdentifierServiceSoap" type="tns:ManageBusinessIdentifierServiceSoap">
        <soap11:binding transport="http://schemas.xmlsoap.org/soap/http"/>
        <wsdl:operation name="Create">
            <soap11:operation soapAction="http://busdox.org/serviceMetadata/ManageBusinessIdentifierService/1.0/      :createIn"
style="document"/>
            <wsdl:input>
                <soap11:body use="literal"/>
            </wsdl:input>
            <wsdl:output>
                <soap11:body use="literal"/>
            </wsdl:output>
            <wsdl:fault name="NotFoundFault">
                <soap:fault name="NotFoundFault" use="literal"/>
            </wsdl:fault>
            <wsdl:fault name="UnauthorizedFault">
                <soap:fault name="UnauthorizedFault" use="literal"/>
            </wsdl:fault>
            <wsdl:fault name="InternalErrorFault">
                <soap:fault name="InternalErrorFault" use="literal"/>
            </wsdl:fault>
            <wsdl:fault name="BadRequestFault">
                <soap:fault name="BadRequestFault" use="literal"/>
            </wsdl:fault>
        </wsdl:operation>
        <wsdl:operation name="CreateList">
            <soap11:operation soapAction="http://busdox.org/serviceMetadata/ManageBusinessIdentifierService/1.0/      :createListIn"
style="document"/>
            <wsdl:input>
                <soap11:body use="literal"/>
            </wsdl:input>
            <wsdl:output>
                <soap11:body use="literal"/>
            </wsdl:output>
            <wsdl:fault name="NotFoundFault">
                <soap:fault name="NotFoundFault" use="literal"/>
            </wsdl:fault>
            <wsdl:fault name="UnauthorizedFault">
                <soap:fault name="UnauthorizedFault" use="literal"/>
            </wsdl:fault>
            <wsdl:fault name="InternalErrorFault">
                <soap:fault name="InternalErrorFault" use="literal"/>
            </wsdl:fault>
            <wsdl:fault name="BadRequestFault">
                <soap:fault name="BadRequestFault" use="literal"/>
```

```
            </wsdl:fault>
        </wsdl:operation>
        <wsdl:operation name="Delete">
            <soap11:operation soapAction="http://busdox.org/serviceMetadata/ManageBusinessIdentifierService/1.0/      :deleteIn"
style="document"/>
            <wsdl:input>
                <soap11:body use="literal"/>
            </wsdl:input>
            <wsdl:output>
                <soap11:body use="literal"/>
            </wsdl:output>
            <wsdl:fault name="NotFoundFault">
                <soap:fault name="NotFoundFault" use="literal"/>
            </wsdl:fault>
            <wsdl:fault name="UnauthorizedFault">
                <soap:fault name="UnauthorizedFault" use="literal"/>
            </wsdl:fault>
            <wsdl:fault name="InternalErrorFault">
                <soap:fault name="InternalErrorFault" use="literal"/>
            </wsdl:fault>
            <wsdl:fault name="BadRequestFault">
                <soap:fault name="BadRequestFault" use="literal"/>
            </wsdl:fault>
        </wsdl:operation>
        <wsdl:operation name="DeleteList">
            <soap11:operation soapAction="http://busdox.org/serviceMetadata/ManageBusinessIdentifierService/1.0/      :deleteListIn"
style="document"/>
            <wsdl:input>
                <soap11:body use="literal"/>
            </wsdl:input>
            <wsdl:output>
                <soap11:body use="literal"/>
            </wsdl:output>
            <wsdl:fault name="NotFoundFault">
                <soap:fault name="NotFoundFault" use="literal"/>
            </wsdl:fault>
            <wsdl:fault name="UnauthorizedFault">
                <soap:fault name="UnauthorizedFault" use="literal"/>
            </wsdl:fault>
            <wsdl:fault name="InternalErrorFault">
                <soap:fault name="InternalErrorFault" use="literal"/>
            </wsdl:fault>
            <wsdl:fault name="BadRequestFault">
                <soap:fault name="BadRequestFault" use="literal"/>
            </wsdl:fault>
        </wsdl:operation>
        <wsdl:operation name="List">
            <soap11:operation soapAction="http://busdox.org/serviceMetadata/ManageBusinessIdentifierService/1.0/      :listIn"
style="document"/>
            <wsdl:input>
                <soap11:body use="literal"/>
            </wsdl:input>
            <wsdl:output>
                <soap11:body use="literal"/>
            </wsdl:output>
            <wsdl:fault name="NotFoundFault">
                <soap:fault name="NotFoundFault" use="literal"/>
            </wsdl:fault>
            <wsdl:fault name="UnauthorizedFault">
                <soap:fault name="UnauthorizedFault" use="literal"/>
            </wsdl:fault>
            <wsdl:fault name="InternalErrorFault">
                <soap:fault name="InternalErrorFault" use="literal"/>
            </wsdl:fault>
            <wsdl:fault name="BadRequestFault">
                <soap:fault name="BadRequestFault" use="literal"/>
            </wsdl:fault>
        </wsdl:operation>
        <wsdl:operation name="PrepareToMigrate">
            <soap11:operation soapAction="http://busdox.org/serviceMetadata/ManageBusinessIdentifierService/1.0/
:prepareMigrateIn" style="document"/>
            <wsdl:input>
```

```
                <soap11:body use="literal"/>
            </wsdl:input>
            <wsdl:output>
                <soap11:body use="literal"/>
            </wsdl:output>
            <wsdl:fault name="NotFoundFault">
                <soap:fault name="NotFoundFault" use="literal"/>
            </wsdl:fault>
            <wsdl:fault name="UnauthorizedFault">
                <soap:fault name="UnauthorizedFault" use="literal"/>
            </wsdl:fault>
            <wsdl:fault name="InternalErrorFault">
                <soap:fault name="InternalErrorFault" use="literal"/>
            </wsdl:fault>
            <wsdl:fault name="BadRequestFault">
                <soap:fault name="BadRequestFault" use="literal"/>
            </wsdl:fault>
        </wsdl:operation>
        <wsdl:operation name="Migrate">
            <soap11:operation soapAction="http://busdox.org/serviceMetadata/ManageBusinessIdentifierService/1.0/      :migrateIn"
style="document"/>
            <wsdl:input>
                <soap11:body use="literal"/>
            </wsdl:input>
            <wsdl:output>
                <soap11:body use="literal"/>
            </wsdl:output>
            <wsdl:fault name="NotFoundFault">
                <soap:fault name="NotFoundFault" use="literal"/>
            </wsdl:fault>
            <wsdl:fault name="UnauthorizedFault">
                <soap:fault name="UnauthorizedFault" use="literal"/>
            </wsdl:fault>
            <wsdl:fault name="InternalErrorFault">
                <soap:fault name="InternalErrorFault" use="literal"/>
            </wsdl:fault>
            <wsdl:fault name="BadRequestFault">
                <soap:fault name="BadRequestFault" use="literal"/>
            </wsdl:fault>
        </wsdl:operation>
    </wsdl:binding>
    <wsdl:service name="ManageBusinessIdentifierService">
        <wsdl:port name="ManageBusinessIdentifierServicePort" binding="tns:ManageBusinessIdentifierServiceSoap">
            <soap:address location="unknown"/>
        </wsdl:port>
    </wsdl:service>
</wsdl:definitions>
```

5.1.1.2. ManageServiceMetadataService

```
<?xml version="1.0" encoding="utf-8"?>
<!--
(C) Copyright 2018 - European Commission | CEF eDelivery

Licensed under the EUPL, Version 1.2 (the "License");
You may not use this file except in compliance with the License.
You may obtain a copy of the License at

https://joinup.ec.europa.eu/sites/default/files/custom-page/attachment/eupl_v1.2_en.pdf

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
-->
<wsdl:definitions xmlns:tns="http://busdox.org/serviceMetadata/ManageServiceMetadataService/1.0/"
xmlns:soap11="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:lrs="http://busdox.org/serviceMetadata/locator/1.0/" xmlns:s="http://www.w3.org/2001/XMLSchema"
xmlns:http="http://schemas.xmlsoap.org/wsdl/http/" name="ManageServiceMetadataService"
targetNamespace="http://busdox.org/serviceMetadata/ManageServiceMetadataService/1.0/"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
```

```
    <wsdl:documentation xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"/>
    <wsdl:types>
        <s:schema elementFormDefault="qualified"
targetNamespace="http://busdox.org/serviceMetadata/ManageServiceMetadataService/1.0/Schema/">
            <s:import namespace="http://busdox.org/serviceMetadata/locator/1.0/" schemaLocation="ServiceMetadataLocatorTypes-
1.0.xsd"/>
        </s:schema>
    </wsdl:types>
    <wsdl:message name="createIn">
        <wsdl:documentation xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"/>
        <wsdl:part name="messagePart" element="lrs:CreateServiceMetadataPublisherService"/>
    </wsdl:message>
    <wsdl:message name="createOut">
        <wsdl:documentation xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"/>
    </wsdl:message>
    <wsdl:message name="readIn">
        <wsdl:documentation xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"/>
        <wsdl:part name="messagePart" element="lrs:ReadServiceMetadataPublisherService"/>
    </wsdl:message>
    <wsdl:message name="readOut">
        <wsdl:documentation xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"/>
        <wsdl:part name="messagePart" element="lrs:ServiceMetadataPublisherService"/>
    </wsdl:message>
    <wsdl:message name="updateIn">
        <wsdl:documentation xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"/>
        <wsdl:part name="messagePart" element="lrs:UpdateServiceMetadataPublisherService"/>
    </wsdl:message>
    <wsdl:message name="updateOut">
        <wsdl:documentation xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"/>
    </wsdl:message>
    <wsdl:message name="deleteIn">
        <wsdl:documentation xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"/>
        <wsdl:part name="messagePart" element="lrs:ServiceMetadataPublisherID"/>
    </wsdl:message>
    <wsdl:message name="deleteOut">
        <wsdl:documentation xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"/>
    </wsdl:message>
    <wsdl:message name="badRequestFault">
        <wsdl:part name="fault" element="lrs:BadRequestFault"/>
    </wsdl:message>
    <wsdl:message name="internalErrorFault">
        <wsdl:part name="fault" element="lrs:InternalErrorFault"/>
    </wsdl:message>
    <wsdl:message name="notFoundFault">
        <wsdl:part name="fault" element="lrs:NotFoundFault"/>
    </wsdl:message>
    <wsdl:message name="unauthorizedFault">
        <wsdl:part name="fault" element="lrs:UnauthorizedFault"/>
    </wsdl:message>
    <wsdl:portType name="ManageServiceMetadataServiceSoap">
        <wsdl:documentation xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"/>
        <wsdl:operation name="Create">
            <wsdl:documentation xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"/>
            <wsdl:input message="tns:createIn"/>
            <wsdl:output message="tns:createOut"/>
            <wsdl:fault message="tns:unauthorizedFault" name="UnauthorizedFault"/>
            <wsdl:fault message="tns:internalErrorFault" name="InternalErrorFault"/>
            <wsdl:fault message="tns:badRequestFault" name="BadRequestFault"/>
        </wsdl:operation>
        <wsdl:operation name="Read">
            <wsdl:documentation xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"/>
            <wsdl:input message="tns:readIn"/>
            <wsdl:output message="tns:readOut"/>
            <wsdl:fault message="tns:notFoundFault" name="NotFoundFault"/>
            <wsdl:fault message="tns:unauthorizedFault" name="UnauthorizedFault"/>
            <wsdl:fault message="tns:internalErrorFault" name="InternalErrorFault"/>
            <wsdl:fault message="tns:badRequestFault" name="BadRequestFault"/>
        </wsdl:operation>
        <wsdl:operation name="Update">
            <wsdl:documentation xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"/>
            <wsdl:input message="tns:updateIn"/>
            <wsdl:output message="tns:updateOut"/>
```

```
            <wsdl:fault message="tns:notFoundFault" name="NotFoundFault"/>
            <wsdl:fault message="tns:unauthorizedFault" name="UnauthorizedFault"/>
            <wsdl:fault message="tns:internalErrorFault" name="InternalErrorFault"/>
            <wsdl:fault message="tns:badRequestFault" name="BadRequestFault"/>
        </wsdl:operation>
        <wsdl:operation name="Delete">
            <wsdl:documentation xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"/>
            <wsdl:input message="tns:deleteIn"/>
            <wsdl:output message="tns:deleteOut"/>
            <wsdl:fault message="tns:notFoundFault" name="NotFoundFault"/>
            <wsdl:fault message="tns:unauthorizedFault" name="UnauthorizedFault"/>
            <wsdl:fault message="tns:internalErrorFault" name="InternalErrorFault"/>
            <wsdl:fault message="tns:badRequestFault" name="BadRequestFault"/>
        </wsdl:operation>
    </wsdl:portType>
    <wsdl:binding name="ManageServiceMetadataServiceSoap" type="tns:ManageServiceMetadataServiceSoap">
        <soap11:binding transport="http://schemas.xmlsoap.org/soap/http"/>
        <wsdl:operation name="Create">
            <soap11:operation soapAction="http://busdox.org/serviceMetadata/ManageServiceMetadataService/1.0/:createIn"
style="document"/>
            <wsdl:input>
                <soap11:body use="literal"/>
            </wsdl:input>
            <wsdl:output>
                <soap11:body use="literal"/>
            </wsdl:output>
            <wsdl:fault name="UnauthorizedFault">
                <soap:fault name="UnauthorizedFault" use="literal"/>
            </wsdl:fault>
            <wsdl:fault name="InternalErrorFault">
                <soap:fault name="InternalErrorFault" use="literal"/>
            </wsdl:fault>
            <wsdl:fault name="BadRequestFault">
                <soap:fault name="BadRequestFault" use="literal"/>
            </wsdl:fault>
        </wsdl:operation>
        <wsdl:operation name="Read">
            <soap11:operation soapAction="http://busdox.org/serviceMetadata/ManageServiceMetadataService/1.0/:readIn"
style="document"/>
            <wsdl:input>
                <soap11:body use="literal"/>
            </wsdl:input>
            <wsdl:output>
                <soap11:body use="literal"/>
            </wsdl:output>
            <wsdl:fault name="NotFoundFault">
                <soap:fault name="NotFoundFault" use="literal"/>
            </wsdl:fault>
            <wsdl:fault name="UnauthorizedFault">
                <soap:fault name="UnauthorizedFault" use="literal"/>
            </wsdl:fault>
            <wsdl:fault name="InternalErrorFault">
                <soap:fault name="InternalErrorFault" use="literal"/>
            </wsdl:fault>
            <wsdl:fault name="BadRequestFault">
                <soap:fault name="BadRequestFault" use="literal"/>
            </wsdl:fault>
        </wsdl:operation>
        <wsdl:operation name="Update">
            <soap11:operation soapAction="http://busdox.org/serviceMetadata/ManageServiceMetadataService/1.0/:updateIn"
style="document"/>
            <wsdl:input>
                <soap11:body use="literal"/>
            </wsdl:input>
            <wsdl:output>
                <soap11:body use="literal"/>
            </wsdl:output>
            <wsdl:fault name="NotFoundFault">
                <soap:fault name="NotFoundFault" use="literal"/>
            </wsdl:fault>
            <wsdl:fault name="UnauthorizedFault">
                <soap:fault name="UnauthorizedFault" use="literal"/>
```

```
            </wsdl:fault>
            <wsdl:fault name="InternalErrorFault">
                <soap:fault name="InternalErrorFault" use="literal"/>
            </wsdl:fault>
            <wsdl:fault name="BadRequestFault">
                <soap:fault name="BadRequestFault" use="literal"/>
            </wsdl:fault>
        </wsdl:operation>
        <wsdl:operation name="Delete">
            <soap11:operation soapAction="http://busdox.org/serviceMetadata/ManageServiceMetadataService/1.0/:deleteIn"
style="document"/>
            <wsdl:input>
                <soap11:body use="literal"/>
            </wsdl:input>
            <wsdl:output>
                <soap11:body use="literal"/>
            </wsdl:output>
            <wsdl:fault name="NotFoundFault">
                <soap:fault name="NotFoundFault" use="literal"/>
            </wsdl:fault>
            <wsdl:fault name="UnauthorizedFault">
                <soap:fault name="UnauthorizedFault" use="literal"/>
            </wsdl:fault>
            <wsdl:fault name="InternalErrorFault">
                <soap:fault name="InternalErrorFault" use="literal"/>
            </wsdl:fault>
            <wsdl:fault name="BadRequestFault">
                <soap:fault name="BadRequestFault" use="literal"/>
            </wsdl:fault>
        </wsdl:operation>
    </wsdl:binding>
    <wsdl:service name="ManageServiceMetadataService">
        <wsdl:port name="ManageServiceMetadataServicePort" binding="tns:ManageServiceMetadataServiceSoap">
            <soap:address location="unknown"/>
        </wsdl:port>
    </wsdl:service>
</wsdl:definitions>
```

5.1.1.3. BDMSLService

```
<?xml version="1.0" encoding="utf-8"?>
<!--
(C) Copyright 2018 - European Commission | CEF eDelivery

Licensed under the EUPL, Version 1.2 (the "License");
You may not use this file except in compliance with the License.
You may obtain a copy of the License at

https://joinup.ec.europa.eu/sites/default/files/custom-page/attachment/eupl_v1.2_en.pdf


Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
-->

<wsdl:definitions xmlns:tns="ec:services:wsdl:BDMSL:1.0" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:lrs="ec:services:wsdl:BDMSL:data:1.0" xmlns:s="http://www.w3.org/2001/XMLSchema"
xmlns:http="http://schemas.xmlsoap.org/wsdl/http/" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns:bd="http://busdox.org/serviceMetadata/locator/1.0/" xmlns:ns="http://schemas.xmlsoap.org/soap/encoding/"
name="BDMSLService" targetNamespace="ec:services:wsdl:BDMSL:1.0">
    <wsdl:documentation xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"/>
    <wsdl:types>
        <s:schema elementFormDefault="qualified" targetNamespace="ec:services:wsdl:BDMSL:1.0">
            <s:import namespace="ec:services:wsdl:BDMSL:data:1.0" schemaLocation="BDMSLService-1.0.xsd"/>
        </s:schema>
        <s:schema elementFormDefault="qualified"
targetNamespace="http://busdox.org/serviceMetadata/ManageServiceMetadataService/1.0/Schema/">
            <s:import namespace="http://busdox.org/serviceMetadata/locator/1.0/" schemaLocation="ServiceMetadataLocatorTypes-
1.0.xsd"/>
        </s:schema>
```

```
        <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
            <xsd:import namespace="http://busdox.org/serviceMetadata/locator/1.0/" schemaLocation="ServiceMetadataLocatorTypes-
1.0.xsd">
        </xsd:import>
        </xsd:schema>
    </wsdl:types>
    <wsdl:message name="prepareChangeCertificateIn">
        <wsdl:documentation xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"/>
        <wsdl:part name="messagePart" element="lrs:PrepareChangeCertificate"/>
    </wsdl:message>
    <wsdl:message name="prepareChangeCertificateOut">
        <wsdl:documentation xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"/>
    </wsdl:message>
    <wsdl:message name="createParticipantIdentifierIn">
        <wsdl:documentation xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"/>
        <wsdl:part name="messagePart" element="lrs:SMPAdvancedServiceForParticipantService"/>
    </wsdl:message>
    <wsdl:message name="createParticipantIdentifierOut">
        <wsdl:documentation xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"/>
    </wsdl:message>
    <wsdl:message name="listParticipantsIn">
        <wsdl:documentation xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"/>
        <wsdl:part name="messagePart" element="lrs:ListParticipantsIn"/>
    </wsdl:message>
    <wsdl:message name="listParticipantsOut">
        <wsdl:documentation xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"/>
        <wsdl:part name="messagePart" element="lrs:ListParticipantsOut"/>
    </wsdl:message>
    <wsdl:message name="isAliveIn">
        <wsdl:documentation xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"/>
        <wsdl:part name="messagePart" element="lrs:IsAlive"/>
    </wsdl:message>
    <wsdl:message name="isAliveOut"/>
    <wsdl:message name="clearCacheIn">
        <wsdl:documentation xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"/>
        <wsdl:part name="messagePart" element="lrs:ClearCache"/>
    </wsdl:message>
    <wsdl:message name="clearCacheOut"/>
    <wsdl:message name="badRequestFault">
        <wsdl:part name="fault" element="bd:BadRequestFault"/>
    </wsdl:message>
    <wsdl:message name="internalErrorFault">
        <wsdl:part name="fault" element="bd:InternalErrorFault"/>
    </wsdl:message>
    <wsdl:message name="notFoundFault">
        <wsdl:part name="fault" element="bd:NotFoundFault"/>
    </wsdl:message>
    <wsdl:message name="unauthorizedFault">
        <wsdl:part name="fault" element="bd:UnauthorizedFault"/>
    </wsdl:message>
    <wsdl:message name="changeCertificateIn">
        <wsdl:documentation xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"/>
        <wsdl:part name="messagePart" element="lrs:ChangeCertificate"/>
    </wsdl:message>
    <wsdl:message name="changeCertificateOut">
        <wsdl:documentation xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"/>
    </wsdl:message>
    <wsdl:message name="ChangeCertificateFault">
        <wsdl:part name="parameters" element="bd:UnauthorizedFault"/>
    </wsdl:message>
    <wsdl:message name="ChangeCertificateFault1">
        <wsdl:part name="parameters" element="bd:BadRequestFault"/>
    </wsdl:message>
    <wsdl:message name="ChangeCertificateFault2">
        <wsdl:part name="parameters" element="bd:InternalErrorFault"/>
    </wsdl:message>
    <wsdl:portType name="BDMSLServiceSoap">
        <wsdl:documentation xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"/>
        <wsdl:operation name="PrepareChangeCertificate">
            <wsdl:documentation xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"/>
            <wsdl:input message="tns:prepareChangeCertificateIn"/>
            <wsdl:output message="tns:prepareChangeCertificateOut"/>
```

```
            <wsdl:fault name="NotFoundFault" message="tns:notFoundFault"/>
            <wsdl:fault name="UnauthorizedFault" message="tns:unauthorizedFault"/>
            <wsdl:fault name="InternalErrorFault" message="tns:internalErrorFault"/>
            <wsdl:fault name="BadRequestFault" message="tns:badRequestFault"/>
        </wsdl:operation>
        <wsdl:operation name="CreateParticipantIdentifier">
            <wsdl:documentation xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"/>
            <wsdl:input message="tns:createParticipantIdentifierIn"/>
            <wsdl:output message="tns:createParticipantIdentifierOut"/>
            <wsdl:fault name="NotFoundFault" message="tns:notFoundFault"/>
            <wsdl:fault name="UnauthorizedFault" message="tns:unauthorizedFault"/>
            <wsdl:fault name="InternalErrorFault" message="tns:internalErrorFault"/>
            <wsdl:fault name="BadRequestFault" message="tns:badRequestFault"/>
        </wsdl:operation>
        <wsdl:operation name="ListParticipants">
            <wsdl:documentation xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"/>
            <wsdl:input message="tns:listParticipantsIn"/>
            <wsdl:output message="tns:listParticipantsOut"/>
            <wsdl:fault name="UnauthorizedFault" message="tns:unauthorizedFault"/>
            <wsdl:fault name="InternalErrorFault" message="tns:internalErrorFault"/>
        </wsdl:operation>
        <wsdl:operation name="IsAlive">
            <wsdl:documentation xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"/>
            <wsdl:input message="tns:isAliveIn"/>
            <wsdl:output message="tns:isAliveOut"/>
            <wsdl:fault name="InternalErrorFault" message="tns:internalErrorFault"/>
        </wsdl:operation>
        <wsdl:operation name="ClearCache">
            <wsdl:documentation xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"/>
            <wsdl:input message="tns:clearCacheIn"/>
            <wsdl:output message="tns:clearCacheOut"/>
            <wsdl:fault name="InternalErrorFault" message="tns:internalErrorFault"/>
        </wsdl:operation>
        <wsdl:operation name="ChangeCertificate">
            <wsdl:documentation xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"/>
            <wsdl:input message="tns:changeCertificateIn"/>
            <wsdl:output message="tns:changeCertificateOut"/>
            <wsdl:fault name="UnauthorizedFault" message="tns:unauthorizedFault"/>
            <wsdl:fault name="InternalErrorFault" message="tns:internalErrorFault"/>
            <wsdl:fault name="BadRequestFault" message="tns:badRequestFault"/>
        </wsdl:operation>
    </wsdl:portType>
    <wsdl:binding name="BDMSLServiceSoap" type="tns:BDMSLServiceSoap">
        <soap:binding transport="http://schemas.xmlsoap.org/soap/http"/>
        <wsdl:operation name="PrepareChangeCertificate">
            <soap:operation soapAction="ec:services:wsdl:BDMSL:1.0:prepareChangeCertificateIn" style="document"/>
            <wsdl:input>
                <soap:body use="literal"/>
            </wsdl:input>
            <wsdl:output>
                <soap:body use="literal"/>
            </wsdl:output>
            <wsdl:fault name="NotFoundFault">
                <soap:fault name="NotFoundFault" use="literal"/>
            </wsdl:fault>
            <wsdl:fault name="UnauthorizedFault">
                <soap:fault name="UnauthorizedFault" use="literal"/>
            </wsdl:fault>
            <wsdl:fault name="InternalErrorFault">
                <soap:fault name="InternalErrorFault" use="literal"/>
            </wsdl:fault>
            <wsdl:fault name="BadRequestFault">
                <soap:fault name="BadRequestFault" use="literal"/>
            </wsdl:fault>
        </wsdl:operation>
        <wsdl:operation name="CreateParticipantIdentifier">
            <soap:operation soapAction="ec:services:wsdl:BDMSL:1.0:createParticipantIdentifierIn" style="document"/>
            <wsdl:input>
                <soap:body use="literal"/>
            </wsdl:input>
            <wsdl:output>
                <soap:body use="literal"/>
```

```
            </wsdl:output>
            <wsdl:fault name="NotFoundFault">
                <soap:fault name="NotFoundFault" use="literal"/>
            </wsdl:fault>
            <wsdl:fault name="UnauthorizedFault">
                <soap:fault name="UnauthorizedFault" use="literal"/>
            </wsdl:fault>
            <wsdl:fault name="InternalErrorFault">
                <soap:fault name="InternalErrorFault" use="literal"/>
            </wsdl:fault>
            <wsdl:fault name="BadRequestFault">
                <soap:fault name="BadRequestFault" use="literal"/>
            </wsdl:fault>
        </wsdl:operation>
        <wsdl:operation name="ListParticipants">
            <soap:operation soapAction="ec:services:wsdl:BDMSL:1.0:listParticipantsIn" style="document"/>
            <wsdl:input>
                <soap:body use="literal"/>
            </wsdl:input>
            <wsdl:output>
                <soap:body use="literal"/>
            </wsdl:output>
            <wsdl:fault name="UnauthorizedFault">
                <soap:fault name="UnauthorizedFault" use="literal"/>
            </wsdl:fault>
            <wsdl:fault name="InternalErrorFault">
                <soap:fault name="InternalErrorFault" use="literal"/>
            </wsdl:fault>
        </wsdl:operation>
        <wsdl:operation name="IsAlive">
            <soap:operation soapAction="ec:services:wsdl:BDMSL:1.0:isAliveIn" style="document"/>
            <wsdl:input>
                <soap:body use="literal"/>
            </wsdl:input>
            <wsdl:output>
                <soap:body use="literal"/>
            </wsdl:output>
        </wsdl:operation>
        <wsdl:operation name="ClearCache">
            <soap:operation soapAction="ec:services:wsdl:BDMSL:1.0:clearCacheIn" style="document"/>
            <wsdl:input>
                <soap:body use="literal"/>
            </wsdl:input>
            <wsdl:output>
                <soap:body use="literal"/>
            </wsdl:output>
            <wsdl:fault name="InternalErrorFault">
                <soap:fault name="InternalErrorFault" use="literal"/>
            </wsdl:fault>
        </wsdl:operation>
        <wsdl:operation name="ChangeCertificate">
            <soap:operation soapAction="ec:services:wsdl:BDMSL:1.0:changeCertificateIn" style="document"/>
            <wsdl:input>
                <soap:body use="literal"/>
            </wsdl:input>
            <wsdl:output>
                <soap:body use="literal"/>
            </wsdl:output>
            <wsdl:fault name="UnauthorizedFault">
                <soap:fault name="UnauthorizedFault" use="literal"/>
            </wsdl:fault>
            <wsdl:fault name="InternalErrorFault">
                <soap:fault name="InternalErrorFault" use="literal"/>
            </wsdl:fault>
            <wsdl:fault name="BadRequestFault">
                <soap:fault name="BadRequestFault" use="literal"/>
            </wsdl:fault>
        </wsdl:operation>
    </wsdl:binding>
    <wsdl:service name="BDMSLService">
        <wsdl:port name="BDMSLServicePort" binding="tns:BDMSLServiceSoap">
            <soap:address location="unknown"/>
```

```
        </wsdl:port>
    </wsdl:service>
</wsdl:definitions>
```

5.1.2. XSD's

5.1.2.1. Identifiers-1.0.xsd

```
<?xml version="1.0" encoding="utf-8"?>
<!--
(C) Copyright 2018 - European Commission | CEF eDelivery

Licensed under the EUPL, Version 1.2 (the "License");
You may not use this file except in compliance with the License.
You may obtain a copy of the License at

https://joinup.ec.europa.eu/sites/default/files/custom-page/attachment/eupl_v1.2_en.pdf

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
-->
<xs:schema xmlns="http://busdox.org/transport/identifiers/1.0/" xmlns:xs="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://busdox.org/transport/identifiers/1.0/" elementFormDefault="qualified" id="Identifiers">
    <xs:annotation>
        <xs:documentation>
    Common identifiers for WSDLs and Schemas
  </xs:documentation>
    </xs:annotation>
    <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:element name="MessageIdentifier" type="MessageIdentifierType"/>
    <xs:element name="ChannelIdentifier" type="ChannelIdentifierType"/>
    <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:simpleType name="MessageIdentifierType">
        <xs:restriction base="xs:string"/>
    </xs:simpleType>
    <xs:simpleType name="ChannelIdentifierType">
        <xs:restriction base="xs:string"/>
    </xs:simpleType>
</xs:schema>
```

5.1.2.2. ServiceGroupReferenceList.xsd

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

```
<!--
(C) Copyright 2018 - European Commission | CEF eDelivery

Licensed under the EUPL, Version 1.2 (the "License");
You may not use this file except in compliance with the License.
You may obtain a copy of the License at

https://joinup.ec.europa.eu/sites/default/files/custom-page/attachment/eupl_v1.2_en.pdf


Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
-->

<xs:schema xmlns="http://busdox.org/serviceMetadata/publishing/1.0/" xmlns:xs="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://busdox.org/serviceMetadata/publishing/1.0/" elementFormDefault="qualified"
id="ServiceGroupReferenceList">
    <xs:include schemaLocation="ServiceMetadataPublishingTypes-1.0.xsd"/>
    <xs:element name="ServiceGroupReferenceList" type="ServiceGroupReferenceListType"/>
    <xs:complexType name="ServiceGroupReferenceListType">
        <xs:sequence>
            <xs:element name="ServiceGroupReference" type="ServiceGroupReferenceType" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="ServiceGroupReferenceType">
        <xs:simpleContent>
            <xs:extension base="xs:string">
                <xs:attribute name="href" type="xs:anyURI"/>
            </xs:extension>
        </xs:simpleContent>
    </xs:complexType>
    <xs:element name="CompleteServiceGroup" type="CompleteServiceGroupType"/>
    <xs:complexType name="CompleteServiceGroupType">
        <xs:sequence>
            <xs:element ref="ServiceGroup"/>
            <xs:element ref="ServiceMetadata" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>
</xs:schema>
```

5.1.2.3. ServiceMetadataLocatorTypes-1.0.xsd

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

<!--
(C) Copyright 2018 - European Commission | CEF eDelivery

Licensed under the EUPL, Version 1.2 (the "License");
You may not use this file except in compliance with the License.
You may obtain a copy of the License at

https://joinup.ec.europa.eu/sites/default/files/custom-page/attachment/eupl_v1.2_en.pdf


Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
-->

<xs:schema xmlns="http://busdox.org/serviceMetadata/locator/1.0/" xmlns:ids="http://busdox.org/transport/identifiers/1.0/"
xmlns:xs="http://www.w3.org/2001/XMLSchema" targetNamespace="http://busdox.org/serviceMetadata/locator/1.0/"
elementFormDefault="qualified" id="ServiceMetadataPublisherService">
    <xs:import namespace="http://busdox.org/transport/identifiers/1.0/" schemaLocation="Identifiers-1.0.xsd"/>
    <xs:element name="ServiceMetadataPublisherID" type="xs:string"/>
    <xs:element name="CreateServiceMetadataPublisherService" type="ServiceMetadataPublisherServiceType"/>
    <xs:element name="ReadServiceMetadataPublisherService" type="ServiceMetadataPublisherServiceType"/>
```

```
    <xs:element name="UpdateServiceMetadataPublisherService" type="ServiceMetadataPublisherServiceType"/>
    <xs:element name="ServiceMetadataPublisherService" type="ServiceMetadataPublisherServiceType"/>
    <xs:complexType name="ServiceMetadataPublisherServiceType">
        <xs:sequence>
            <xs:element name="PublisherEndpoint" type="PublisherEndpointType"/>
            <xs:element ref="ServiceMetadataPublisherID"/>
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="PublisherEndpointType">
        <xs:sequence>
            <xs:element name="LogicalAddress" type="xs:anyURI"/>
            <xs:element name="PhysicalAddress" type="xs:string"/>
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="ServiceMetadataPublisherServiceForParticipantType">
        <xs:sequence>
            <xs:element ref="ServiceMetadataPublisherID"/>
            <xs:element ref="ids:ParticipantIdentifier"/>
        </xs:sequence>
    </xs:complexType>
    <xs:element name="CreateParticipantIdentifier" type="ServiceMetadataPublisherServiceForParticipantType"/>
    <xs:element name="DeleteParticipantIdentifier" type="ServiceMetadataPublisherServiceForParticipantType"/>
    <xs:element name="ParticipantIdentifierPage" type="ParticipantIdentifierPageType"/>
    <xs:element name="CreateList" type="ParticipantIdentifierPageType"/>
    <xs:element name="DeleteList" type="ParticipantIdentifierPageType"/>
    <xs:complexType name="ParticipantIdentifierPageType">
        <xs:sequence>
            <xs:element ref="ids:ParticipantIdentifier" minOccurs="0" maxOccurs="unbounded"/>
            <xs:element ref="ServiceMetadataPublisherID" minOccurs="0"/>
            <xs:element name="NextPageIdentifier" type="xs:string" minOccurs="0"/>
        </xs:sequence>
    </xs:complexType>
    <xs:element name="PageRequest" type="PageRequestType"/>
    <xs:complexType name="PageRequestType">
        <xs:sequence>
            <xs:element ref="ServiceMetadataPublisherID"/>
            <xs:element name="NextPageIdentifier" type="xs:string" minOccurs="0"/>
        </xs:sequence>
    </xs:complexType>
    <xs:element name="PrepareMigrationRecord" type="MigrationRecordType"/>
    <xs:element name="CompleteMigrationRecord" type="MigrationRecordType"/>
    <xs:complexType name="MigrationRecordType">
        <xs:sequence>
            <xs:element ref="ServiceMetadataPublisherID"/>
            <xs:element ref="ids:ParticipantIdentifier"/>
            <xs:element name="MigrationKey" type="xs:string"/>
        </xs:sequence>
    </xs:complexType>
    <xs:element name="BadRequestFault" type="FaultType"/>
    <xs:element name="InternalErrorFault" type="FaultType"/>
    <xs:element name="NotFoundFault" type="FaultType"/>
    <xs:element name="UnauthorizedFault" type="FaultType"/>
    <xs:complexType name="FaultType">
        <xs:sequence>
            <xs:element name="FaultMessage" type="xs:string" minOccurs="0"/>
        </xs:sequence>
    </xs:complexType>
</xs:schema>
```

5.1.2.4. ServiceMetadataPublishingTypes-1.0.xsd

```
<?xml version="1.0" encoding="utf-8"?>
<!--
(C) Copyright 2018 - European Commission | CEF eDelivery

Licensed under the EUPL, Version 1.2 (the "License");
You may not use this file except in compliance with the License.
You may obtain a copy of the License at

https://joinup.ec.europa.eu/sites/default/files/custom-page/attachment/eupl_v1.2_en.pdf
```

```
Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
-->
<xs:schema xmlns="http://busdox.org/serviceMetadata/publishing/1.0/" xmlns:ids="http://busdox.org/transport/identifiers/1.0/"
xmlns:ds="http://www.w3.org/2000/09/xmldsig#" xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:wsa="http://www.w3.org/2005/08/addressing" targetNamespace="http://busdox.org/serviceMetadata/publishing/1.0/"
elementFormDefault="qualified" id="ServiceMetadataPublishing">
    <xs:import namespace="http://www.w3.org/2000/09/xmldsig#" schemaLocation="xmldsig-core-schema.xsd"/>
    <xs:import namespace="http://busdox.org/transport/identifiers/1.0/" schemaLocation="Identifiers-1.0.xsd"/>
    <xs:import namespace="http://www.w3.org/2005/08/addressing" schemaLocation="ws-addr.xsd"/>
    <xs:element name="ServiceGroup" type="ServiceGroupType"/>
    <xs:element name="ServiceMetadata" type="ServiceMetadataType"/>
    <xs:element name="SignedServiceMetadata" type="SignedServiceMetadataType"/>
    <xs:complexType name="SignedServiceMetadataType">
        <xs:annotation>
            <xs:documentation>The SignedServiceMetadata structure is a ServiceMetadata structure
   that has been signed by the ServiceMetadataPublisher, according to governance policies.</xs:documentation>
        </xs:annotation>
        <xs:sequence>
            <xs:element ref="ServiceMetadata">
                <xs:annotation>
                    <xs:documentation>The ServiceMetadata element covered by the Signature.</xs:documentation>
                </xs:annotation>
            </xs:element>
            <xs:element ref="ds:Signature">
                <xs:annotation>
                    <xs:documentation>Represents an enveloped XML() Signature over the SignedServiceMetadata
element.</xs:documentation>
                </xs:annotation>
            </xs:element>
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="ServiceMetadataType">
        <xs:annotation>
            <xs:documentation>
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 sender
Access Point needs to know in order to send a message to that service.
   </xs:documentation>
        </xs:annotation>
        <xs:sequence>
            <xs:choice>
                <xs:element name="ServiceInformation" type="ServiceInformationType">
                    <xs:annotation>
                        <xs:documentation>Contains service information for an actual service registration, rather than a redirect to
another SMP</xs:documentation>
                    </xs:annotation>
                </xs:element>
                <xs:element name="Redirect" type="RedirectType">
                    <xs:annotation>
                        <xs:documentation>
For recipients that want to associate more than one SMP with their participant identifier,
they may redirect senders to an alternative SMP for specific document types. To achieve
this, the ServiceMetadata element defines the optional element 'Redirect'. This element
holds the URL of the alternative SMP, as well as the Subject Unique Identifier of the
destination SMPs certificate used to sign its resources.
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.
        </xs:documentation>
                    </xs:annotation>
                </xs:element>
```

```
            </xs:choice>
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="ServiceInformationType">
        <xs:sequence>
            <xs:element ref="ids:ParticipantIdentifier">
                <xs:annotation>
                    <xs:documentation>The participant identifier. Comprises the identifier, and an identifier scheme. This identifier
MUST have the same value of the {id} part of the URI of the enclosing ServiceMetadata resource.</xs:documentation>
                </xs:annotation>
            </xs:element>
            <xs:element ref="ids:DocumentIdentifier">
                <xs:annotation>
                    <xs:documentation>Represents the type of document that the recipient is able to handle.</xs:documentation>
                </xs:annotation>
            </xs:element>
            <xs:element name="ProcessList" type="ProcessListType">
                <xs:annotation>
                    <xs:documentation>Represents the processes that a specific document type can participate in, and endpoint address
and binding information. Each process element describes a specific business process that accepts this type of document as() Input and
holds a list of endpoint addresses (in the case that the service supports multiple transports) of services that implement the business
process, plus information about the transport used for each endpoint.</xs:documentation>
                </xs:annotation>
            </xs:element>
            <xs:element name="Extension" type="ExtensionType" minOccurs="0">
                <xs:annotation>
                    <xs:documentation>The extension element may contain any XML element. Clients MAY ignore this
element.</xs:documentation>
                </xs:annotation>
            </xs:element>
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="ProcessListType">
        <xs:annotation>
            <xs:documentation>List of processes</xs:documentation>
        </xs:annotation>
        <xs:sequence>
            <xs:element name="Process" type="ProcessType" maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="ProcessType">
        <xs:sequence>
            <xs:element ref="ids:ProcessIdentifier">
                <xs:annotation>
                    <xs:documentation>The identifier of the process.</xs:documentation>
                </xs:annotation>
            </xs:element>
            <xs:element name="ServiceEndpointList" type="ServiceEndpointList">
                <xs:annotation>
                    <xs:documentation>List of one or more endpoints that support this process.</xs:documentation>
                </xs:annotation>
            </xs:element>
            <xs:element name="Extension" type="ExtensionType" minOccurs="0">
                <xs:annotation>
                    <xs:documentation>The extension element may contain any XML element. Clients MAY ignore this
element.</xs:documentation>
                </xs:annotation>
            </xs:element>
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="ServiceEndpointList">
        <xs:annotation>
            <xs:documentation>Contains a list of all endpoint</xs:documentation>
        </xs:annotation>
        <xs:sequence>
            <xs:element name="Endpoint" type="EndpointType" maxOccurs="unbounded">
                <xs:annotation>
                    <xs:documentation>Endpoint represents the technical endpoint and address type of the recipient, as an
URL.</xs:documentation>
                </xs:annotation>
            </xs:element>
        </xs:sequence>
```

```
    </xs:complexType>
    <xs:complexType name="EndpointType">
        <xs:sequence>
            <xs:element ref="wsa:EndpointReference">
                <xs:annotation>
                    <xs:documentation>The address of an endpoint, as an WS-Addressing Endpoint Reference</xs:documentation>
                </xs:annotation>
            </xs:element>
            <xs:element name="RequireBusinessLevelSignature" type="xs:boolean">
                <xs:annotation>
                    <xs:documentation>Set to &amp;quot;true&amp;quot; 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, such as the START 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 APs.</xs:documentation>
                </xs:annotation>
            </xs:element>
            <xs:element name="MinimumAuthenticationLevel" type="xs:string" minOccurs="0">
                <xs:annotation>
                    <xs:documentation>Indicates the minimum authentication level that recipient requires. The specific semantics of this field is defined in a specific instance of the BUSDOX infrastructure. It could for example reflect the value of the &amp;quot;urn:eu:busdox:attribute:assurance-level&amp;quot; SAML attribute defined in the START specification.</xs:documentation>
                </xs:annotation>
            </xs:element>
            <xs:element name="ServiceActivationDate" type="xs:dateTime" minOccurs="0">
                <xs:annotation>
                    <xs:documentation>Activation date of the service. Senders should ignore services that are not yet activated.</xs:documentation>
                </xs:annotation>
            </xs:element>
            <xs:element name="ServiceExpirationDate" type="xs:dateTime" minOccurs="0">
                <xs:annotation>
                    <xs:documentation>Expiration date of the service. Senders should ignore services that are expired.</xs:documentation>
                </xs:annotation>
            </xs:element>
            <xs:element name="Certificate" type="xs:string">
                <xs:annotation>
                    <xs:documentation>Holds the complete signing certificate of the recipient AP, as a PEM base 64 encoded X509 DER formatted value.</xs:documentation>
                </xs:annotation>
            </xs:element>
            <xs:element name="ServiceDescription" type="xs:string">
                <xs:annotation>
                    <xs:documentation>A human readable description of the service</xs:documentation>
                </xs:annotation>
            </xs:element>
            <xs:element name="TechnicalContactUrl" type="xs:anyURI">
                <xs:annotation>
                    <xs:documentation>Represents a link to human readable contact information. This might also be an email address.</xs:documentation>
                </xs:annotation>
            </xs:element>
            <xs:element name="TechnicalInformationUrl" type="xs:anyURI" minOccurs="0">
                <xs:annotation>
                    <xs:documentation>A URL to human readable documentation of the service format. This could for example be a web site containing links to XML Schemas, WSDLs, Schematrons and other relevant resources.</xs:documentation>
                </xs:annotation>
            </xs:element>
            <xs:element name="Extension" type="ExtensionType" minOccurs="0">
                <xs:annotation>
                    <xs:documentation>The extension element may contain any XML element. Clients MAY ignore this element.</xs:documentation>
                </xs:annotation>
            </xs:element>
        </xs:sequence>
        <xs:attribute name="transportProfile" type="xs:string">
            <xs:annotation>
                <xs:documentation>Indicates the type of BUSDOX transport that is being used between access points, e.g. the BUSDOX START profile. This specification defines the following identifier URI which denotes the BUSDOX START transport: &amp;quot;busdox-transport-start&amp;quot;</xs:documentation>
            </xs:annotation>
```

```
        </xs:attribute>
    </xs:complexType>
    <xs:complexType name="ServiceGroupType">
        <xs:annotation>
            <xs:documentation>The ServiceGroup structure 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 SignedServiceMetadata resources in the ServiceList
    structure.</xs:documentation>
        </xs:annotation>
        <xs:sequence>
            <xs:element ref="ids:ParticipantIdentifier">
                <xs:annotation>
                    <xs:documentation>Represents the business level endpoint key and key type, e.g. a DUNS or GLN number that is
associated with a group of services. </xs:documentation>
                </xs:annotation>
            </xs:element>
            <xs:element name="ServiceMetadataReferenceCollection" type="ServiceMetadataReferenceCollectionType">
                <xs:annotation>
                    <xs:documentation>The ServiceMetadataReferenceCollection structure holds a list of references to
SignedServiceMetadata structures. From this list, a sender can follow the references to get each SignedServiceMetadata
structure.</xs:documentation>
                </xs:annotation>
            </xs:element>
            <xs:element name="Extension" type="ExtensionType" minOccurs="0">
                <xs:annotation>
                    <xs:documentation>The extension element may contain any XML element. Clients MAY ignore this
element.</xs:documentation>
                </xs:annotation>
            </xs:element>
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="ServiceMetadataReferenceCollectionType">
        <xs:annotation>
            <xs:documentation>Contains the URL to a specific SignedServiceMetadata instance. Note
    that references MUST refer to SignedServiceMetadata records that are signed by the
    certificate of the SMP. It must not point to SignedServiceMetadata resources published
    by external SMPs.</xs:documentation>
        </xs:annotation>
        <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:annotation>
                <xs:documentation>Contains the URL to a specific SignedServiceMetadata instance.</xs:documentation>
            </xs:annotation>
        </xs:attribute>
    </xs:complexType>
    <xs:complexType name="RedirectType">
        <xs:sequence>
            <xs:element name="CertificateUID" type="xs:string">
                <xs:annotation>
                    <xs:documentation>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.</xs:documentation>
                </xs:annotation>
            </xs:element>
            <xs:element name="Extension" type="ExtensionType" minOccurs="0">
                <xs:annotation>
                    <xs:documentation>The extension element may contain any XML element. Clients MAY ignore this
element.</xs:documentation>
                </xs:annotation>
            </xs:element>
        </xs:sequence>
        <xs:attribute name="href" type="xs:anyURI">
            <xs:annotation>
                <xs:documentation>The destination URL of the redirect.</xs:documentation>
            </xs:annotation>
        </xs:attribute>
```

```
    </xs:complexType>
    <xs:complexType name="ExtensionType">
        <xs:annotation>
            <xs:documentation>
Child elements of the &lt;smp:Extension&gt; element are known as &quot;custom
extension elements&quot;. Extension points may be used for optional extensions
of service metadata. This implies:
* Extension elements added to a specific Service Metadata resource MUST be ignorable
by any client of the transport infrastructure. The ability to parse and adjust client
behavior based on an extension element MUST NOT be a prerequisite for a client to
locate a service, or to make a successful request at the referenced service.
* A client MAY ignore any extension element added to specific service metadata
resource instances.
    </xs:documentation>
        </xs:annotation>
        <xs:sequence>
            <!-- TODO processContents="skip" will be added after 1.1.0 -->
            <xs:any/>
        </xs:sequence>
    </xs:complexType>
</xs:schema>
```

5.1.2.5. BDMSLService-1.0.xsd

```
<?xml version="1.0" encoding="utf-8"?>
<!--
(C) Copyright 2018 - European Commission | CEF eDelivery

Licensed under the EUPL, Version 1.2 (the "License");
You may not use this file except in compliance with the License.
You may obtain a copy of the License at

https://joinup.ec.europa.eu/sites/default/files/custom-page/attachment/eupl_v1.2_en.pdf


Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
-->

<xs:schema xmlns="ec:services:wsdl:BDMSL:data:1.0" xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:bd="http://busdox.org/serviceMetadata/locator/1.0/" xmlns:id="http://busdox.org/transport/identifiers/1.0/"
targetNamespace="ec:services:wsdl:BDMSL:data:1.0" elementFormDefault="qualified" id="BDMSLTypes">
    <xs:import namespace="http://busdox.org/serviceMetadata/locator/1.0/" schemaLocation="ServiceMetadataLocatorTypes-1.0.xsd"/>
    <xs:import namespace="http://busdox.org/transport/identifiers/1.0/" schemaLocation="Identifiers-1.0.xsd"/>
    <xs:element name="PrepareChangeCertificate" type="PrepareChangeCertificateType"/>
    <xs:element name="ChangeCertificate" type="ChangeCertificateType"/>
    <xs:element name="SMPAdvancedServiceForParticipantService" type="SMPAdvancedServiceForParticipantType"/>
    <xs:element name="ListParticipantsOut" type="ListParticipantsType"/>
    <xs:element name="ClearCache" type="ClearCacheType"/>
    <xs:element name="IsAlive" type="IsAliveType"/>
    <xs:element name="ListParticipantsIn" type="ListParticipantsInType"/>
    <xs:complexType name="PrepareChangeCertificateType">
        <xs:sequence>
            <xs:element name="newCertificatePublicKey" type="xs:string">
                <xs:annotation>
                    <xs:documentation>The new public key contained in the certificate.</xs:documentation>
                </xs:annotation>
            </xs:element>
            <xs:element name="migrationDate" type="xs:date">
                <xs:annotation>
                    <xs:documentation>The migration date for the new certificate. Can't be in the past.
              </xs:documentation>
                </xs:annotation>
            </xs:element>
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="ChangeCertificateType">
        <xs:sequence>
            <xs:element ref="bd:ServiceMetadataPublisherID">
```

```
                <xs:annotation>
                    <xs:documentation>The SMP identifier</xs:documentation>
                </xs:annotation>
            </xs:element>
            <xs:element name="newCertificatePublicKey" type="xs:base64Binary">
                <xs:annotation>
                    <xs:documentation>The new public key contained in the certificate.</xs:documentation>
                </xs:annotation>
            </xs:element>
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="SMPAdvancedServiceForParticipantType">
        <xs:sequence>
            <xs:element name="CreateParticipantIdentifier" type="bd:ServiceMetadataPublisherServiceForParticipantType"/>
            <xs:element name="serviceName" type="xs:string">
                <xs:annotation>
                    <xs:documentation>The name of the service for the NAPTR record.</xs:documentation>
                </xs:annotation>
            </xs:element>
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="ListParticipantsType">
        <xs:sequence>
            <xs:element name="participant" type="ParticipantsType" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="ParticipantsType">
        <xs:sequence>
            <xs:element ref="id:ParticipantIdentifier">
                <xs:annotation>
                    <xs:documentation>The participant identifier</xs:documentation>
                </xs:annotation>
            </xs:element>
            <xs:element ref="bd:ServiceMetadataPublisherID">
                <xs:annotation>
                    <xs:documentation>The SMP identifier</xs:documentation>
                </xs:annotation>
            </xs:element>
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="ListParticipantsInType"/>
    <xs:complexType name="IsAliveType"/>
    <xs:complexType name="ClearCacheType"/>
</xs:schema>
```

5.2. List of valid issuing agencies for PEPPOL

Scheme ID	ICD value	Issuing Organization	Deprecated	Since	Last Modification	Structure of code	Display requirements	PEPPOL Examples
FR:SIRENE	0002	Institut National de la Statistique et des Etudes Economiques, (I.N.S.E.E.)	false	1.0.0	1.1.1 - Changed from FR:SIRET to FR:SIRENE (see ISU Jira ISU-231)	1) Number of characters: 9 characters ("SIREN") 14 " 9+5 ("SIRET"), The 9 character number designates an organization, The 14 character number designates a specific establishment of the organization designated by the first 9 characters. 2) Check digits: 9th & 14th character respectively	The 9 figure code number (SIREN) is written in groups of 3 characters. Example: 784 301 772, The 14 figure code number is written in 3 groups of 3 characters and a single group of 5. Example: 784 301 772 00025	
SE:ORGNR	0007	The National Tax Board	false	1.0.0		1) 10 digits. 1st digit = Group number, 2nd - 9th digit = Ordinalnumber1st digit, = Group number, 10th digit = Check digit, 2) Last digit.	Single group of 10 digits.	
FR:SIRET	0009	DU PONT DE NEMOURS	false	1.1.1		1) 14 digits, 2) None	In four groups, Groups 1 - 3 = three digits each, Group 4 = five digits	
FI:OVT	0037	National Board of Taxes, (Verohallitus)	false	1.0.0		1) ICD 4 Digits, Organization code upto 11 characters, Organization name upto 250 characters, 2) None - Example: 00371234567800001 - 0037 Country code for Finland (ISO 6523 International Code Designator (ICD) value) - 12345678 Business ID without hyphen - 00001 Optional specifier for organisation unit (assigned by the organisation itself)	None	

Scheme ID	ICD value	Issuing Organization	Deprecated	Since	Last Modification	Structure of code	Display requirements	PEPPOL Examples
DUNS	0060	Dun and Bradstreet Ltd	false	1.0.0		1) 8 digits, 1st-7th digit = number, 8th digit = check number, 2) digit	Single group of 8 digits	OVT identifier conforming to standard ISO6523. - Constant 0037 (Finnish tax administration organisation code) - Finnish local tax ID, 8 characters with initial zero and no hyphen - Free-format 5 characters, for example profit center. Example: 003710948874
GLN	0088	EAN International	false	1.0.0		1) Eight identification digits and a check digit. A two digit prefix will be added in the future but it will not be used to calculate the check digit. 2) The Organization name is not part of the D-U-N-S number.	IIIIIIIIC where all characters are the digits 0, to 9, I = an identification digit and C = the check digit. When the prefix (P) is added the display requirement will be eleven digits, PPIIIIIIIIC.	
DK:P	0096	Danish Chamber of Commerce	false	1.0.0		1) 13 digits including check digits, 2) None	None	
IT:FTI	0097	FTI - Ediforum Italia	false	1.0.0		Character repertoire, The EDI identifier consists of digits only. The identifier has a fixed length. No separators are required. Structure: [123] [123456] [123456] [12], 17, < >, A B C D, A: numerical value allocated by the RA to the regional sub-authority, (3 digits), B: numerical value allocated by the sub-authority to the registered organization (mandatory part of	None	

Scheme ID	ICD value	Issuing Organization	Deprecated	Since	Last Modification	Structure of code	Display requirements	PEPPOL Examples
						the identifier; 6 digits), C: numerical value used by the registered organization (free part; 6 digits), D: numerical check digit calculated by the registered organization; (2 digits), Check digit computation, The check digit is modular 97 computed on ABC as one number.		
NL:KVK	0106	Vereniging van Kamers van Koophandel en Fabrieken in Nederland, Scheme	false	1.1.2				
IT:SIA	0135	SIA-Società Interbancaria per l'Automazione S.p.A.	false	1.0.0		Structure of EDI identifier, Character repertoire, The EDI identifier consists of digits only. The identifier has a fixed length. No separators are required. Structure: [1234567] [123] [1] [12345], min 11- max 16, < >, A B C D, A: numerical value (7 digits) assigned by Uffico Provinciale IVA (local branch of Ministry of Finance); B: numerical value a (3 digits) identifying the County; C: numerical check digit (1 digit); D: optional numerical value (up to 5 digits0 used by the registered organization (free part). Check digit computation, The check digit algorithm is the one published in the Gazzetta Ufficiale no 345 of December 29 1976.	None	
IT:SECETI	0142	Servizi Centralizzati SECETI S.p.A.	false	1.0.0		First field: ICD: 4 digits, Second field: sequence of digits	None	
DIGST	0184	DIGSTORG	false	1.2.1		Defined by Danish Agency for Digitisation		

Scheme ID	ICD value	Issuing Organization	Deprecated	Since	Last Modification	Structure of code	Display requirements	PEPPOL Examples
DK:CPR	9901	Danish Ministry of the Interior and Health	false	1.0.0		1) First field: ICD: 4 digits, Second field: sequence of digits	None	
DK:CVR	9902	The Danish Commerce and Companies Agency	false	1.0.0				7603770123
DK:SE	9904	Danish Ministry of Taxation, Central Customs and Tax Administration	false	1.0.0				DK26769388
DK:VANS	9905	Danish VANS providers	false	1.0.0				DK26769388
IT:VAT	9906	Ufficio responsabile gestione partite IVA	false	1.0.0				IT06363391001
IT:CF	9907	TAX Authority	false	1.0.0				RSSBBR69C48F839A NOTE: The "CF" is a Fiscal Code that can be "personal" or for a "legal entity". The CF for legal entities is like the Italian VAT code (IT:VAT)
NO:ORGNR	9908	Enhetsregisteret ved Bronnoysundregisterne	false	1.0.0				
NO:VAT	9909	Enhetsregisteret ved Bronnoysundregisterne	true	1.0.0	1.1.0 - deprecated			990399123
HU:VAT	9910		false	1.0.0				990399123MVA
EU:VAT	9912	National ministries of Economy	true	1.0.0	1.1.0 - deprecated			
EU:REID	9913	Business Registers Network	false	1.0.0				
AT:VAT	9914	Österreichische Umsatzsteuer-Identifikationsnummer	false	1.0.0				
AT:GOV	9915	Österreichisches Verwaltungs bzw. Organisationskennzeichen	false	1.0.0				ATU12345678

Scheme ID	ICD value	Issuing Organization	Deprecated	Since	Last Modification	Structure of code	Display requirements	PEPPOL Examples
AT:CID	9916	Firmenidentifikationsnummer der Statistik Austria	true	1.0.0	1.0.2 - deprecated			
IS:KT	9917	Icelandic National Registry	false	1.0.0				
IBAN	9918	SOCIETY FOR WORLDWIDE INTERBANK FINANCIAL, TELECOMMUNICATION S.W.I.F.T	false	1.0.1				
AT:KUR	9919	Kennziffer des Unternehmensregisters	false	1.0.2		9 characters in total; letter, number x3, letter, number x3, letter		
ES:VAT	9920	Agencia Española de Administración Tributaria	false	1.0.2				
IT:IPA	9921	Indice delle Pubbliche Amministrazioni	false	1.1.0				
AD:VAT	9922	Andorra VAT number	false	1.1.0				
AL:VAT	9923	Albania VAT number	false	1.1.0				
BA:VAT	9924	Bosnia and Herzegovina VAT number	false	1.1.0				
BE:VAT	9925	Belgium VAT number	false	1.1.0				
BG:VAT	9926	Bulgaria VAT number	false	1.1.0				
CH:VAT	9927	Switzerland VAT number	false	1.1.0				
CY:VAT	9928	Cyprus VAT number	false	1.1.0				
CZ:VAT	9929	Czech Republic VAT number	false	1.1.0				
DE:VAT	9930	Germany VAT number	false	1.1.0				
EE:VAT	9931	Estonia VAT number	false	1.1.0				
GB:VAT	9932	United Kingdom VAT number	false	1.1.0				
GR:VAT	9933	Greece VAT number	false	1.1.0				
HR:VAT	9934	Croatia VAT number	false	1.1.0				
IE:VAT	9935	Ireland VAT number	false	1.1.0				

Scheme ID	ICD value	Issuing Organization	Deprecated	Since	Last Modification	Structure of code	Display requirements	PEPPOL Examples
LI:VAT	9936	Liechtenstein VAT number	false	1.1.0				
LT:VAT	9937	Lithuania VAT number	false	1.1.0				
LU:VAT	9938	Luxemburg VAT number	false	1.1.0				
LV:VAT	9939	Latvia VAT number	false	1.1.0				
MC:VAT	9940	Monaco VAT number	false	1.1.0				
ME:VAT	9941	Montenegro VAT number	false	1.1.0				
MK:VAT	9942	Macedonia, the former Yugoslav Republic of VAT number	false	1.1.0				
MT:VAT	9943	Malta VAT number	false	1.1.0				
NL:VAT	9944	Netherlands VAT number	false	1.1.0				
PL:VAT	9945	Poland VAT number	false	1.1.0				
PT:VAT	9946	Portugal VAT number	false	1.1.0				
RO:VAT	9947	Romania VAT number	false	1.1.0				
RS:VAT	9948	Serbia VAT number	false	1.1.0				
SI:VAT	9949	Slovenia VAT number	false	1.1.0				
SK:VAT	9950	Slovakia VAT number	false	1.1.0				
SM:VAT	9951	San Marino VAT number	false	1.1.0				
TR:VAT	9952	Turkey VAT number	false	1.1.0				
VA:VAT	9953	Holy See (Vatican City State) VAT number	false	1.1.0				
NL:OIN	9954	Dutch Originator's Identification Number	false	1.1.3	1.1.4 - renamed from ION to OIN			
SE:VAT	9955	Swedish VAT number	false	1.2.0				
BE:CBE	9956	Belgian Crossroad Bank of Enterprises	false	1.2.1		Format: 9.999.999.999 - Check: 99 = 97 - (9.999.999.9 modulo 97)		
FR:VAT	9957	French VAT number	false	1.2.1				

Table 23 – PEPPOL issuing agencies

6. ANNEXE 1 – DOCUMENT PARTS

UC v1.0.xlsx

Drawings - Interface Control Document for

PEPPOL Code Lists 1 2 1-15072016.xls

7. LIST OF FIGURES

List of TablesTable 1 - Documented services 6

8. CONTACT INFORMATION

CEF Support Team

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

By phone: +32 2 299 09 09

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

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