Service Metadata Publishing (SMP) Version 2.0

Committee Specification ~~02~~03

~~16~~22 January ~~2020~~2021

This stage:
https://docs.oasis-open.org/bdxr/bdx-smp/v2.0/cs03/bdx-smp-v2.0-cs03.docx (Authoritative)
https://docs.oasis-open.org/bdxr/bdx-smp/v2.0/cs03/bdx-smp-v2.0-cs03.html
https://docs.oasis-open.org/bdxr/bdx-smp/v2.0/cs03/bdx-smp-v2.0-cs03.pdf

Previous stage:
https://docs.oasis-open.org/bdxr/bdx-smp/v2.0/cs02/bdx-smp-v2.0-cs02.docx (Authoritative)
https://docs.oasis-open.org/bdxr/bdx-smp/v2.0/cs02/bdx-smp-v2.0-cs02.html
https://docs.oasis-open.org/bdxr/bdx-smp/v2.0/cs02/bdx-smp-v2.0-cs02.pdf

~~**Previous stage:**~~
~~(Authoritative)~~

Latest stage:
https://docs.oasis-open.org/bdxr/bdx-smp/v2.0/bdx-smp-v2.0.docx (Authoritative)
https://docs.oasis-open.org/bdxr/bdx-smp/v2.0/bdx-smp-v2.0.html
https://docs.oasis-open.org/bdxr/bdx-smp/v2.0/bdx-smp-v2.0.pdf

Technical Committee:
OASIS Business Document Exchange (BDXR) TC

Chair:
Kenneth Bengtsson (kbengtsson@efact.pe), Individual member

Editors:
Kenneth Bengtsson (kbengtsson@efact.pe), Individual member
Erlend Klakegg Bergheim (erlend.klakegg.bergheim@difi.no), Difi-Agency for Public Management and eGovernment
Sander Fieten (sander@chasquis-consulting.com), Individual member
G. Ken Holman (gkholman@CraneSoftwrights.com), Crane Softwrights Ltd.

Additional artifacts:
This prose specification is one component of a Work Product that also includes:

- XML schemas: https://docs.oasis-open.org/bdxr/bdx-smp/v2.0/cs03/xsd/ and https://docs.oasis-open.org/bdxr/bdx-smp/v2.0/~~cs02~~cs03/xsdrt/
- Model documentation: https://docs.oasis-open.org/bdxr/bdx-smp/v2.0/cs03/mod/

Related work:
This specification replaces or supersedes:

- *Service Metadata Publishing (SMP) Version 1.0*. Edited by Jens Aabol, Kenneth Bengtsson, Erlend Klakegg Bergheim, Sander Fieten, and Sven Rasmussen. 01 August 2017. OASIS Standard. http://docs.oasis-open.org/bdxr/bdx-smp/v1.0/os/bdx-smp-v1.0-os.html.

This specification is related to:

- *Business Document Metadata Service Location Version 1.0.* Edited by Dale Moberg and Pim van der Eijk. Latest stage: http://docs.oasis-open.org/bdxr/BDX-Location/v1.0/BDX-Location-v1.0.html.

Declared XML namespaces:

- http://docs.oasis-open.org/bdxr/ns/SMP/2/ServiceGroup
- http://docs.oasis-open.org/bdxr/ns/SMP/2/ServiceMetadata
- http://docs.oasis-open.org/bdxr/ns/SMP/2/AggregateComponents
- http://docs.oasis-open.org/bdxr/ns/SMP/2/BasicComponents
- http://docs.oasis-open.org/bdxr/ns/SMP/2/ExtensionComponents
- http://docs.oasis-open.org/bdxr/ns/SMP/2/QualifiedDataTypes
- http://docs.oasis-open.org/bdxr/ns/SMP/2/UnqualifiedDataTypes

Abstract:

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 of the business document, such as types of documents the recipient 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.

Status:

This document was last revised or approved by the OASIS Business Document Exchange (BDXR) TC on the above date. The level of approval is also listed above. Check the "Latest stage" location noted above for possible later revisions of this document. Any other numbered Versions and other technical work produced by the Technical Committee (TC) are listed at https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=bdxr#technical.

TC members should send comments on this specification to the TC's email list. Others should send comments to the TC's public comment list, after subscribing to it by following the instructions at the "Send A Comment" button on the TC's web page at https://www.oasis-open.org/committees/bdxr/.

This specification is provided under the Non-Assertion Mode of the OASIS IPR Policy, the mode chosen when the Technical Committee was established. For information on whether any patents have been disclosed that may be essential to implementing this specification, and any offers of patent licensing terms, please refer to the Intellectual Property Rights section of the TC's web page (https://www.oasis-open.org/committees/bdxr/ipr.php).

Note that any machine-readable content (Computer Language Definitions) declared Normative for this Work Product is provided in separate plain text files. In the event of a discrepancy between any such plain text file and display content in the Work Product's prose narrative document(s), the content in the separate plain text file prevails.

Citation format:

When referencing this specification the following citation format should be used:

[BDX-SMP-v2.0]

Service Metadata Publishing (SMP) Version 2.0. Edited by Kenneth Bengtsson, Erlend Klakegg Bergheim, Sander Fieten, and G. Ken Holman. ~~16~~22 January ~~2020~~2021. OASIS Committee Specification ~~02~~03. https://docs.oasis-open.org/bdxr/bdx-smp/v2.0/cs03/bdx-smp-v2.0-cs03.html. Latest stage: https://docs.oasis-open.org/bdxr/bdx-smp/v2.0/bdx-smp-v2.0.html.

Notices

All capitalized terms in the following text have the meanings assigned to them in the OASIS Intellectual Property Rights Policy (the "OASIS IPR Policy"). The full Policy may be found at the OASIS website.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published, and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this section are included on all such copies and derivative works. However, this document itself may not be modified in any way, including by removing the copyright notice or references to OASIS, except as needed for the purpose of developing any document or deliverable produced by an OASIS Technical Committee (in which case the rules applicable to copyrights, as set forth in the OASIS IPR Policy, must be followed) or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by OASIS or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and OASIS DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY OWNERSHIP RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. OASIS AND ITS MEMBERS WILL NOT BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF ANY USE OF THIS DOCUMENT OR ANY PART THEREOF.

As stated in the OASIS IPR Policy, the following three paragraphs in brackets apply to OASIS Standards Final Deliverable documents (Committee Specifications, OASIS Standards, or Approved Errata).

[OASIS requests that any OASIS Party or any other party that believes it has patent claims that would necessarily be infringed by implementations of this OASIS ~~Committee Specification or OASIS Standard~~Standards Final Deliverable, to notify OASIS TC Administrator and provide an indication of its willingness to grant patent licenses to such patent claims in a manner consistent with the IPR Mode of the OASIS Technical Committee that produced this ~~specification.~~deliverable.]

[OASIS invites any party to contact the OASIS TC Administrator if it is aware of a claim of ownership of any patent claims that would necessarily be infringed by implementations of this ~~specification~~OASIS Standards Final Deliverable by a patent holder that is not willing to provide a license to such patent claims in a manner consistent with the IPR Mode of the OASIS Technical Committee that produced this ~~specification.~~OASIS Standards Final Deliverable. OASIS may include such claims on its website, but disclaims any obligation to do so~~.~~.]

[OASIS takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this ~~document~~OASIS Standards Final Deliverable or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Information on OASIS' procedures with respect to rights in any document or deliverable produced by an OASIS Technical Committee can be found on the OASIS website. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this OASIS ~~Committee Specification or OASIS Standard~~Standards Final Deliverable, can be obtained from the OASIS TC Administrator. OASIS makes no representation that any information or list of intellectual property rights will at any time be complete, or that any claims in such list are, in fact, Essential Claims~~.~~.]

The name "OASIS" is a trademark of OASIS, the owner and developer of this ~~specification~~document, and should be used only to refer to the organization and its official outputs. OASIS welcomes reference to, and implementation and use of, ~~specifications~~documents, while reserving the right to enforce its marks against misleading uses. Please see https://www.oasis-open.org/policies-guidelines/trademark for above guidance.

Table of Contents

1 Introduction

1.1 Service Metadata Publishing

This document describes the Service Metadata Publishing protocol (SMP) and its binding to a **[REST]** interface for Service Metadata Publication within a 4-corner network. It defines the data model for the messages exchanged between a Service Metadata Publisher and a client application wishing to discover the endpoint information necessary to send a business document to the intended recipient (defined in this specification as a "Participant"). A client application in this context can either be an end-user business application or a gateway in a 4-corner network (sometimes also referred to as an Access Point).

It also specifies how this endpoint discovery process is implemented using a REST transport interface. The SMP protocol itself however is open for binding to other transport protocols such as AS4, however such bindings are not specified in this specification.

SMP is typically used to discover endpoint information and capabilities between entities exchanging business documents in a 4-cornered network. In some 4-cornered networks, such as is the case in the European eHealth domain, business information is being exchanged in different structured forms than as documents. The term "document" used in this specification may in such networks be interpreted as referring to any resource that is being exchanged in the network.

1.2 IPR Policy

This specification is provided under the Non-Assertion Mode of the OASIS IPR Policy, the mode chosen when the Technical Committee was established. For information on whether any patents have been disclosed that may be essential to implementing this specification, and any offers of patent licensing terms, please refer to the Intellectual Property Rights section of the TC's web page (https://www.oasis-open.org/committees/bdxr/ipr.php).

1.3 Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

1.4 Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997. http://www.ietf.org/rfc/rfc2119.txt.

[RFC7231] "Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content", RFC 7231, June 2014. https://tools.ietf.org/html/rfc7231

[RFC7232] "Hypertext Transfer Protocol (HTTP/1.1): Conditional Requests", RFC 7232, June 2014, http://www.ietf.org/rfc/rfc7232.txt

[XML 1.0] "Extensible Markup Language (XML) 1.0 (Fifth Edition)", W3C Recommendations, 26 November 2008, http://www.w3.org/TR/xml/

[Unicode] "The Unicode Standard, Version 7.0.0", (Mountain View, CA: The Unicode Consortium, 2014. ISBN 978-1-936213-09-2) http://www.unicode.org/versions/Unicode7.0.0/

[XML-DSIG1] XML Signature Syntax and Processing Version 1.1, D. Eastlake, J. Reagle, D. Solo, F. Hirsch, M. Nyström, T. Roessler, K. Yiu, Editors, W3C Recommendation, April 11, 2013, http://www.w3.org/TR/2013/REC-xmldsig-core1-20130411/. Latest version available at http://www.w3.org/TR/xmldsig-core1/

[X509v3] ITU-T Recommendation X.509 version 3 (1997). "Information Technology - Open Systems Interconnection - The Directory Authentication Framework" ISO/IEC 9594-8:1997

[CCTS] UN/CEFACT Core Component Technical Specification, Version 2.01, http://www.unece.org/fileadmin/DAM/cefact/codesfortrade/CCTS/CCTS_V2-01_Final.pdf

[RFC3986] Berners-Lee, T., Fielding, R., Masinter, L., “Uniform Resource Identifier (URI): Generic Syntax”, RFC 3986, January 2005, http://tools.ietf.org/rfc/rfc3986

[C14N11] Canonical XML Version 1.1, W3C Recommendation, John Boyer and Glenn Marcy, 2 May 2008, http://www.w3.org/TR/2008/REC-xml-c14n11-20080502/

1.5 Non-Normative References

[REST] “Architectural Styles and the Design of Network-based Software Architectures”, http://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm

[BDXL] “Business Document Metadata Service Location (BDXL) Version 1.0“, Committee Specification, 10 June 2014, http://docs.oasis-open.org/bdxr/BDX-Location/v1.0/cs01/BDX-Location-v1.0-cs01.html

[ebCorePartyId] “OASIS ebCore Party Id Type Technical Specification Version 1.0. OASIS Committee Specification”, September 2010, https://docs.oasis-open.org/ebcore/PartyIdType/v1.0/PartyIdType-1.0.odt

[XAdES] XML Advanced Electronic Signatures. ETSI TS 101 903 V1.4.1, June 2009, http://uri.etsi.org/01903/v1.4.1/ts_101903v010401p.pdf.

2 SMP Protocol

2.1 The Service Discovery Process

2.1.1 Introduction

The SMP protocol is intended to discover the capabilities of Participants in a network of entities. It allows Participants and/or their Access Points in the network to find the technical endpoints of their trading partners. In the 4-corner architecture the technical endpoint, the so-called Access Point, can be provided to the Participant as a service by a separate entity.

In such a 4-cornered network, the discovery process is often a two-step process that starts with the lookup of the SMP service that holds the service metadata information about a Participant in the network. Each Participant is registered in the BDXL with one and only one Service Metadata Publisher. This lookup MAY be performed by the client using the Business Document Metadata Service Location protocol **[BDXL]**.

After retrieving the location of the SMP service, the client can then retrieve the metadata associated with the Participant. This metadata includes the information necessary to transmit the message to the recipient's network endpoint.

The diagram below represents the lookup flow for a sender contacting both the BDXL and the SMP:

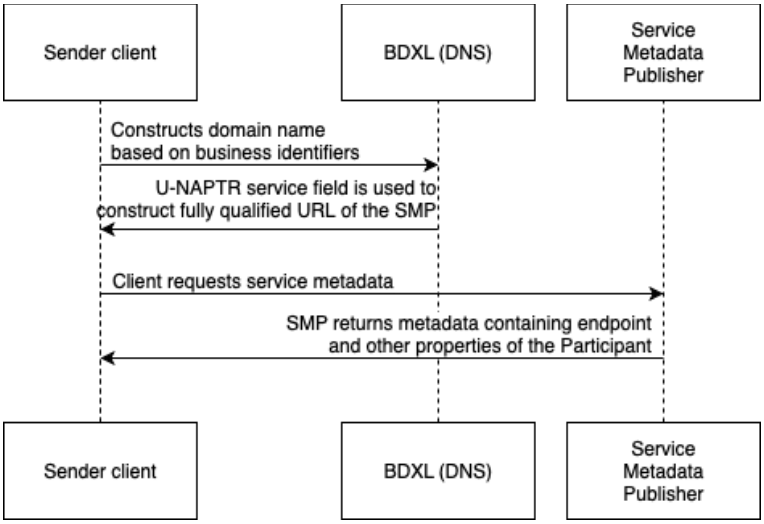

Fig 1: Participant lookup with Service Metadata

Note that the use of BDXL to discover an SMP service is OPTIONAL. Networks implementing SMP MAY define alternative mechanisms for discovering SMP services within the network.

To accelerate the discovery process, the sender client MAY cache the metadata retrieved from the SMP instead of performing a lookup for every transaction (see: 5.2.2 Caching of HTTP responses).

2.1.2 Discovering services associated with a Participant

In addition to the direct lookup of Service Metadata based on a Participant identifier and service type, a sender MAY want to discover what services are provided by a given Participant. Such discovery is relevant for applications supporting several equivalent business processes. Knowing the capabilities of the Participant is valuable information to a sender application and ultimately to an end user. E.g. the end user may be presented with a choice between a "simple" and an "elaborate" business process.

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

2.1.3 Service Metadata Publisher Redirection

In most scenarios, a Participant-lookup will only point to a single Service Metadata Publisher. However, there are cases where a Participant would want to use different Service Metadata Publishers for different document types or processes. This is supported by Service Metadata Publisher Redirection.

In this pattern, the sender is redirected by the Service Metadata Publisher to a secondary, remote Service Metadata Publisher where the actual Service Metadata can be found. A Redirect element within the response points to the SMP that holds further information about the destination SMP, as illustrated in the following diagram:

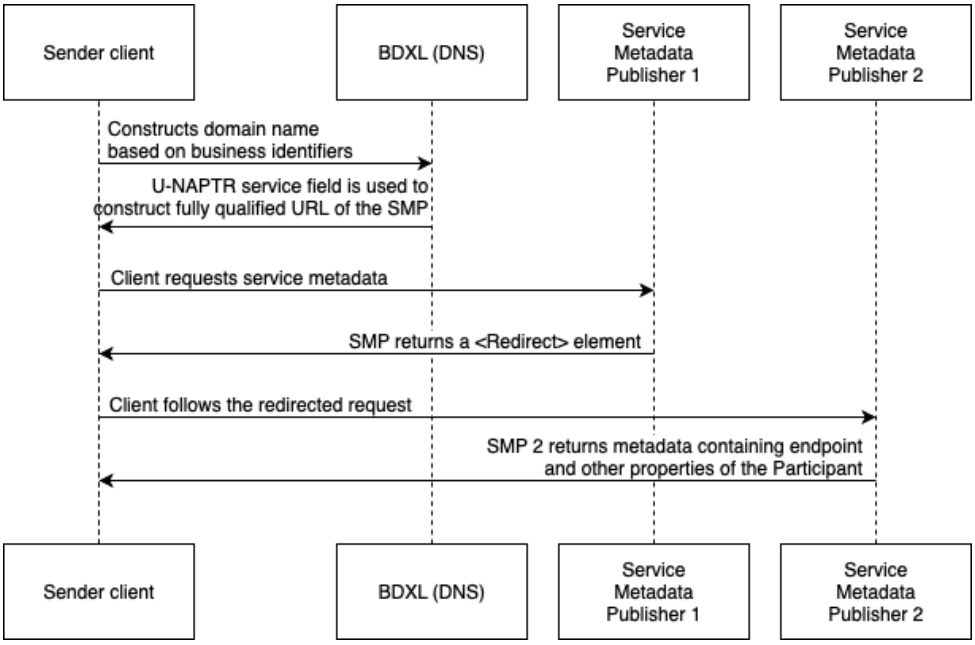

Fig 2: Service Metadata redirection

To avoid cyclic references, an SMP client SHOULD NOT follow a redirect instruction from an SMP service if already being redirected from another SMP service. Likewise, an SMP service that has been redirected to by another SMP service SHOULD NOT redirect further.

An SMP service MUST respond with either a service metadata element or an error code.

3 Identifiers

3.1 Introduction

This section defines what participant-, service- and process-identifiers are, and how they are represented within the SMP protocol.

3.2 Notational conventions

For describing the textual format of identifiers, the following conventions are used:

- Everything within the curly brackets { } can be substituted by specific values.
- Everything with square brackets [ ] represents optional content, whether literals or not.
- Everything outside the curly brackets MUST be treated as literals.

For example, for an identifier with the value *5798000000001*, the format definition

/{identifier}/service[/{service ID}]

Can be instantiated to either of the strings

/5798000000001/service

And

/5798000000001/service/urn:oasis:names:specification:ubl:schema:xsd:Invoice-2::Invoice##UBL-2.1

3.3 On the use of percent encoding in URLs

Identifiers may contain characters that need to be percent encoded when included in an URL to ensure the URL is valid (see sections 2.1 and 2.2 of **[RFC3986]**). When such identifier is used in a URL, the client MUST encode each path segment (i.e. the parts between slashes) individually. An SMP implementation MUST first parse the URL before decoding percent encoded characters (see also section 2.4 of **[RFC3986]**).

For example, this implies that for an URL in the form of */[{identifier scheme}::]{participant ID}/services/{service ID}*, the slash literals MUST NOT be URL encoded.

3.4 On scheme identifiers

Identifier schemes for all schemed identifier types (participants, services, processes) 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 services, participants or profiles that are relevant to support in that instance.

An example is the **[ebCorePartyId]** specification, which defines a mechanism for referencing participant identification schemes using a formal URN notation.

Another example is the participant identifier scheme being used in some European networks, *iso6523-actorid-upis*.

3.5 Case sensitivity handling

An identifier scheme MAY define its own requirements for case sensitivity handling. Unless defined differently by the identifier scheme, an identifier MUST be treated as case insensitive.

When a case insensitive identifier scheme is being used, an implementation MUST fold all characters to lower case.

3.6 Participant identifiers

3.6.1 Participant identifiers and schemes

A “participant identifier” is a business level endpoint key that uniquely identifies an end-user entity (“participant”) in a network. Examples of identifiers are company registration and VAT numbers, DUNS numbers, GLN numbers, email addresses etc. Participant identifiers are associated with groups of services or service metadata.

Participant identifiers SHOULD consist of a scheme identifier in addition to the participant identifier itself. The scheme identifier indicates the specification of the participant identifier format, i.e. its representation and meaning.

3.6.2 XML format for Participant identifiers

The *<ParticipantID>* element is used to represent participant identifiers and scheme information.

Example XML representation of a participant ID:

```
<ParticipantID
  schemeID="iso6523-actorid-upis">9908:810418052</ParticipantID>
```

In this example, the scheme identifier *iso6523-actorid-upis* tells the SMP client that the representation of the participant identifier *9908:810418052* follows the convention in some European networks for identifying participants in the network.

3.6.3 Using participant identifiers in URLs

The following format is used:

[{identifier scheme}::]{participant ID}

Where *identifier scheme* is the scheme of the identifier, and *participant ID* is the participant identifier itself, following the format indicated by the identifier scheme.

In a URL, the string represented by *[{identifier scheme}::]{participant ID}* MUST be percent encoded as specified in section 3.3.

*Non-normative example using the **[ebCorePartyId]** URN format, assuming an ISO 6523 International Code Designator 9908 with the participant identifier 810418052:*

urn:oasis:names:tc:ebcore:partyid-type:iso6523:9908::810418052

In percent encoded form:

urn%3Aoasis%3Anames%3Atc%3Aebcore%3Apartyid-type%3Aiso6523%3A9908%3A%3A810418052

And the same non-normative example using the Universal Participant Identifier Format as being used in some European networks:

iso6523-actorid-upis::9908:810418052

In percent encoded form:

iso6523-actorid-upis%3A%3A9908%3A810418052

3.7 Service identifiers

3.7.1 Service and document schemes defined by SMP

3.7.1.1 Introduction

Services and document types are represented by an identifier (typically identifying the document type) and a scheme identifier which represents the scheme or format of the identifier itself. It is outside the scope of this document to list identifier schemes that may be valid in a given context.

3.7.1.2 Representing QName/Subtype Identifier

This specification defines the *QName/Subtype Identifier* scheme, which is identified by the following URI:

bdx-docid-qns

Values of this scheme are based on a concatenation of the service or document type namespace, root element, and OPTIONAL (and document type or service dependent) subtype:

{rootNamespace}::{documentElementLocalName}[##{Subtype identifier}]

For example, in the case of a UBL 2.1 invoice, this service or document can then be identified by

01 ***Root namespace:*** *urn:oasis:names:specification:ubl:schema:xsd:Invoice-2*
02 ***Document element local name:*** *Invoice*
03 ***Subtype identifier:*** *UBL-2.1 (since several versions of the Invoice schema may use the same namespace + document element name)*

The service identifier will then be:

urn:oasis:names:specification:ubl:schema:xsd:Invoice-2::Invoice##UBL-2.1

3.7.1.3 Representing JSON Identifier

This specification defines the *JSON Identifier scheme*, which is identified by the following URI:

bdx-docid-json

Values of this scheme are based on a concatenation of the root schema and OPTIONAL (and document type or service dependent) subtype:

{rootSchema}[##{Subtype identifier}]

For example, in the case of a person document, this service or document can then be identified by

01 ***Root schema:*** *https://example.com/person.schema.json*
02 ***Subtype identifier:*** *vcard-1.0 (specifying a specific specification of which the schema is used.)*

The service identifier will then be:

https://example.com/person.schema.json##vcard-1.0

3.7.2 XML Representation of service identifiers

The *<ServiceID>* element is used to represent service identifiers and scheme information.

Example XML representation of a Service ID (QName/Subtype identifier)::

```
<ServiceID schemeID="bdx-docid-qns"
>urn:oasis:names:specification:ubl:schema:xsd:Invoice-2::Invoice##UBL-2.1</ServiceID>
```

Example XML representation of a Service ID (JSON Identifier):

```
<ServiceID schemeID="bdx-docid-json"
>https://example.com/person.schema.json##vcard-1.0</ServiceID>
```

Where the schemeID attribute indicates the scheme of the service identifier.

3.7.3 URL representation of service identifiers

When representing service identifiers in URLs, the service identifier itself will be prefixed with the scheme identifier.

The generic format of this is:

[{identifier scheme}::]{service ID}

In the case that the *QName/Subtype Identifier Scheme* defined in section 3.7.1.2 is used, the complete format is:

`bdx-docid-qns::{rootNamespace}::{documentElementLocalName}[##{Subtype identifier}]`

As a non-normative example, in the case of a UBL 2.1 invoice, this service can then be identified by:

- ***Identifier scheme:*** *bdx-docid-qns*
- ***Root namespace:*** *urn:oasis:names:specification:ubl:schema:xsd:Invoice-2*
- ***Document element local name:*** *Invoice*
- ***Subtype identifier:*** *UBL-2.1 (since several versions of the Invoice schema may use the same namespace + document element name)*

The service identifier will then be:

`bdx-docid-qns::urn:oasis:names:specification:ubl:schema:xsd:Invoice-2::Invoice##UBL-2.1`

Rules for parsing this example identifier:

- *The text up until the first `::` is the identifier scheme identifier*
- *The text between the first `::` and second `::` is the root namespace*
- *The text between the second occurrence of `::` and next occurrence of `##` OR end of the string is the document element local name*
- *The text following the first `##` after the document element local name (if any) is the subtype identifier*

This string MUST be percent encoded if used in an URL. In that case, the above identifier will then read as:

`bdx-docid-qns%3A%3Aurn%3Aoasis%3Anames%3Aspecification%3Aubl%3Aschema%3Axsd%3AInvoice-2%3A%3AInvoice%23%23UBL-2.1`

Note the limitation that XML service types with the following characteristics MUST NOT be referenced using Service Metadata Publishing when using the `QName/Subtype Identifier` scheme:

- Services with only local names (i.e. without namespaces)
- Services that need to be identified with a subtype identifier, and where the subtype part of the identifier does not correspond to a specific, mandatory attribute value or element value in the document that is based on XML Schema simple content.

When using a service type with such characteristics, a different scheme identifier MUST be used.

4 Data Model

4.1 Class diagram

The **[CCTS]**-modeled classes of information in the SMP model and the relationships between them are depicted in the below class diagram:

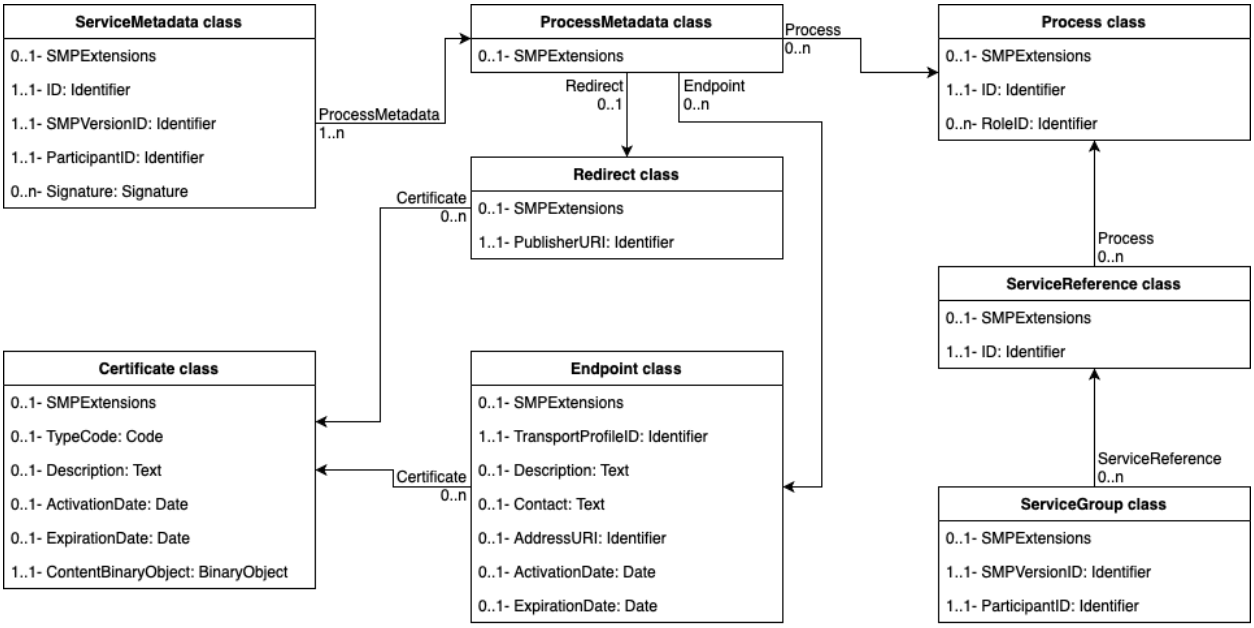

Fig 3: SMP class diagram

OPTIONAL extensions MAY be introduced at the beginning of every major entity (see section 4.4.1 for information about extensions). The relationship between the CCTS-modelled classes and the non-CCTS extensions are depicted in the below diagram:

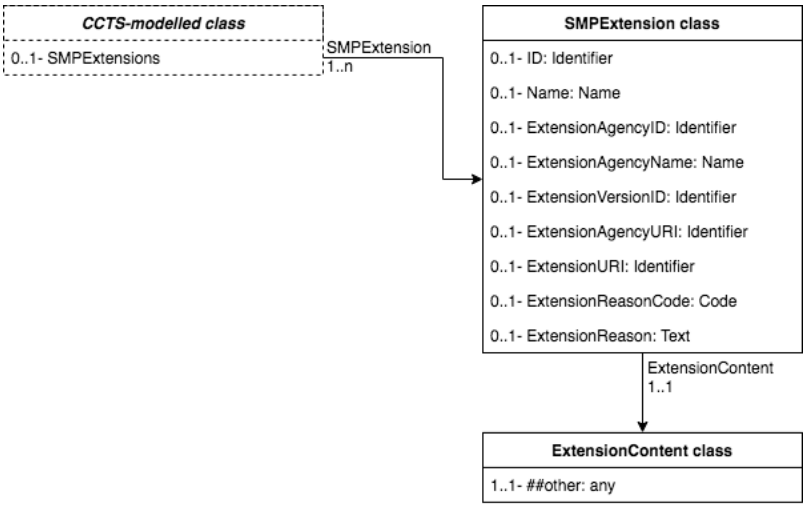

Fig 4: SMP extensions class diagram

4.2 CCTS and non-CCTS information

The information described in the SMP data model is documented in two parts:

The Basic SMP information related to ServiceGroup and ServiceMetadata as well as their underlying classes is modeled as CCTS classes.

Additional information related to extensions and signatures is not modeled as CCTS classes. Such information is realized in the schema expressions as additional document constraints and is documented in the Additional SMP information section.

4.3 Basic SMP information

4.3.1 The ServiceGroup class

The CCTS-modeled objects in the ServiceGroup class are as follows:

Name **(Unqualified data type)**	**Cardinality**	**Description**
SMPExtensions	0..1	A container for all extensions present at the ServiceGroup level.
SMPVersionID (Identifier)	1..1	The version of the Service Metadata Publishing specification in use. For SMP 2.0, this value MUST be set to “2.0”.
ParticipantID (Identifier)	1..1	Represents a business level endpoint key that uniquely identifies a Participant in the network. Examples of identifiers are company registration and VAT numbers, DUNS numbers, GLN numbers, email addresses etc. See section 3.6 of this specification for information on this data type.
ServiceReference (ServiceReference class)	0..n	The ServiceReference structure holds a list of references to ServiceMetadata structures. From this list, a sender can follow the references to get each ServiceMetadata structure.

4.3.2 The ServiceMetadata class

The CCTS-modeled objects in the ServiceMetadata class are as follows:

Name **(Unqualified data type)**	**Cardinality**	**Description**
SMPExtensions	0..1	A container for all extensions present at the ServiceMetadata level.

ID
(Identifier)
	1..1	A service identifier representing a specific service or document type. An example XML representation of a ServiceID using a UBL 2.1 Invoice document using the default “bdx-docid-qns” scheme is:
		`<smb:ServiceID schemeID="bdx-docid-qns">`
		`urn:oasis:names:specification:ubl:schema:xsd:Invoice-2::Invoice##UBL-2.1`
		`</smb:ServiceID>`
		See section 3.7 of this specification for information on this data type.

Name (Unqualified data type)	Cardinality	Description
SMPVersionID (Identifier)	1..1	The version of the Service Metadata Publishing specification in use. For SMP 2.0, this value MUST be set to “2.0”.
ParticipantID (Identifier)	1..1	Represents a business level endpoint key that uniquely identifies a Participant in the network. Examples of identifiers are company registration and VAT numbers, DUNS numbers, GLN numbers, email addresses etc. See section 3.6 of this specification for information on this data type.
ProcessMetadata (ProcessMetadata class)	1..n	Metadata containing information about how to locate the network endpoint for the given Participant.

4.3.3 The ServiceReference class

The CCTS-modeled objects in the ServiceReference class are as follows:

Name (Unqualified data type)	Cardinality	Description
SMPExtensions	0..1	A container for all extensions present at the ServiceReference level.

ID (Identifier)	1..1	A service identifier representing a specific service or document type. An example XML representation of an ID using a UBL 2.1 Invoice document using the default “bdx-docid-qns” scheme is:
		`<smb:ID schemeID="bdx-docid-qns">`
		`urn:oasis:names:specification:ubl:schema:xsd:`
		`Invoice-2::Invoice##UBL-2.1`
		`</smb:ID>`
		See section 3.7 of this specification for information on this data type.

Process (Process class)	0..n	Information about the process of which the service of the participant is a part.

4.3.4 The ProcessMetadata class

ProcessMetadata MUST have as a child element either a Redirect element or one or more Endpoint elements. The ProcessMetadata MUST NOT contain an Endpoint element and a Redirect element at the same time.

The CCTS-modeled objects in the ProcessMetadata class are as follows:

Name (Unqualified data type)	Cardinality	Description
SMPExtensions	0..1	A container for all extensions present at the ProcessMetadata level.
Endpoint	0..n	The Endpoint element contains information about the network endpoint of the Participant.

Name (Unqualified data type)	Cardinality	Description
(Endpoint class)		
Redirect (Redirect class)	0..1	The presence of a Redirect element indicates that a client MUST follow the URL in the PublisherURI element of the Redirect class.
Process (Process class)	0..n	Information about the process of which the service of the participant is a part.

4.3.5 The Process class

The CCTS-modeled objects in the Process class are as follows:

Name (Unqualified data type)	Cardinality	Description
SMPExtensions	0..1	A container for all extensions present at the Process level.
ID (Identifier)	1..1	The identifier of the process. A process is identified by a string that (with the exception of `bdx:noprocess`) is defined outside of this specification. For example, the CEN workshop on Business Interoperability Interfaces (BII) has chosen to indicate a UBL-based ”simple procurement” process (or ”profile” in UBL terminology) with the identifier “BII07”, and a UBL-based basic invoice exchange profile with the identifier “BII04”. This document defines one process identifier, which represents documents that are not sent under any specific process: `bdx:noprocess` A process identifier specification policy MAY define its own requirements for case sensitivity handling. Unless defined differently in such specification or policy, the process identifier MUST be treated as case insensitive.
RoleID (Identifier)	0..n	May be used to indicate the role of the participant in a process where more than two roles are defined or where distinguishing implicit roles based on service identifier is not possible.

4.3.6 The Endpoint class

The CCTS-modeled objects in the Endpoint class are as follows:

Name (Unqualified data type)	Cardinality	Description
SMPExtensions	0..1	A container for all extensions present at the Endpoint level.
TransportProfileID (Identifier)	1..1	Indicates the type of transport method that is being used between access points.

Name (Unqualified data type)	Cardinality	Description
Description (Text)	0..1	A human readable description of the endpoint.
Contact (Text)	0..1	Represents a link to human readable contact information. This might also be an email address.
AddressURI (Identifier)	0..1	The address of an endpoint, as a URL.
ActivationDate (Date)	0..1	Activation date of the service. Senders SHOULD ignore services that are not yet activated. Data type of ServiceActivationDate date is `xsd:date`. The service activation period includes the date specified in ActivationDate date. ActivationDate MUST be a date before the ExpirationDate.
ExpirationDate (Date)	0..1	Expiration date of the service. Senders SHOULD ignore services that are expired. Data type of ExpirationDate date is `xsd:date`. The active service period is exclusive of the ExpirationDate date. ExpirationDate MUST be a date after the ActivationDate.
Certificate (Certificate class)	0..n	One or more certificates used to validate the communication with an endpoint.

4.3.7 The Redirect class

The CCTS-modeled objects in the Redirect class are as follows:

Name (Unqualified data type)	Cardinality	Description
SMPExtensions	0..1	A container for all extensions present at the Redirect level.
PublisherURI (Identifier)	1..1	A client MUST follow the URL in the PublisherURI element to get to the SMP holding the information.
Certificate (Certificate class)	0..1	The certificate used to validate information signed by the destination SMP.

4.3.8 The Certificate class

The CCTS-modeled objects in the Certificate class are as follows:

Name (Unqualified data type)	Cardinality	Description
SMPExtensions	0..1	A container for all extensions present at the Certificate level.
TypeCode (Code)	0..1	The use of the certificate being provided, expressed as a user or domain defined code.

Name (Unqualified data type)	Cardinality	Description
Description (Text)	0..1	An optional and informal description of the certificate.
ActivationDate (Date)	0..1	The date from which the embedded certificate can be used, extracted from the certificate itself or set explicitly by the endpoint. In either case the certificate should not be considered for use before the activation date.
ExpirationDate (Date)	0..1	The date from which the embedded certificate can no longer be used, extracted from the certificate itself or set explicitly by the endpoint. In either case the certificate should not be considered for use from the expiration date onwards.
ContentBinaryObject (BinaryObject)	1..1	Holds the complete certificate of the recipient endpoint or SMP. Specifying specific certificate formats is outside the scope of this specification. It is up to the implementing communities to agree on local certificate practice.

4.4 Additional SMP information

4.4.1 Extensions

4.4.1.1 On the use of extensions

For each major entity, extension points have been added with the OPTIONAL <SMPExtensions> element. Semantics and use child elements of the <SMPExtensions> element are known as “custom extension elements”. Extension points MAY be used for OPTIONAL extensions of service metadata. When using extensions in a global context, this implies that:

- Cardinality at extension points is by definition unbounded. An SMP publishing service MAY introduce as many extensions at each extension point as wanted.
- SMP publishing services MUST NOT produce metadata that contain extensions necessary for a Client to understand in order to make use of this metadata. 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.

Notwithstanding the above, when SMP extensions are used in a private context, such as between two entities exchanging business documents or by a community in a closed infrastructure, the use of extensions MAY be made mandatory as long as such requirement for mandatory use of extensions only applies to business document exchange within the private context where it has been defined and that all participating parties agree on the use and mandatory status of the extension(s).

The extension point, when it exists, MUST contain one or more user-defined extensions, with each extension wrapped with OPTIONAL extension metadata identifying properties of the extension.

4.4.1.2 Extension information

Name (Unqualified data type)	Cardinality	Description
SMPExtension	1..n	A single extension for private use.

Name (Unqualified data type)	Cardinality	Description
ID (Identifier)	0..1	An identifier for the Extension assigned by the creator of the extension.
Name (Name)	0..1	A name for the Extension assigned by the creator of the extension.
ExtensionAgencyID (Identifier)	0..1	An agency that maintains one or more Extensions.
ExtensionAgencyName (Name)	0..1	The name of the agency that maintains the Extension.
ExtensionVersionID (Identifier)	0..1	The version of the Extension.
ExtensionAgencyURI (Identifier)	0..1	A URI for the Agency that maintains the Extension.
ExtensionURI (Identifier)	0..1	A URI for the Extension.
ExtensionReasonCode (Code)	0..1	A code for reason the Extension is being included.
ExtensionReason (Text)	0..1	A description of the reason for the Extension.
ExtensionContent	1..1	The definition of the extension content. Any valid XML structure can be inserted here.

4.4.2 Signature information

Using the W3C XML Digital Signature **[XML-DSIG1]**, zero or more signatures can be added to the SMP ServiceGroup and to the SMP ServiceMetadata.

The signatures MUST be grouped as the final children of the ServiceGroup and ServiceMetadata elements.

See section 5.6 for more information about the use of digital signatures in the REST binding to secure the meta-data retrieval.

5 Service Metadata Publishing REST binding

5.1 Introduction

This section describes the REST binding of the Service Metadata Publishing protocol. Note that the implementation of the SMP protocol is not limited to the REST binding and future specifications MAY define additional bindings to other transport protocols, like for example AS4.

5.2 The use of HTTP 1.x

5.2.1 General use of HTTP 1.x

An implementation of the REST binding MUST support the use of GET and HEAD as specified in **[RFC7231]**, and MUST set the HTTP "content-type" header and give it a value of "application/xml". A business document exchange infrastructure MAY set restrictions on what ports are allowed.

An implementation of SMP MAY choose to manage resources through the HTTP POST, PUT and DELETE verbs. It is however up to each implementation to choose how to manage records, and the use of HTTP POST, PUT and DELETE is not mandated or regulated by this specification.

HTTP GET operations MUST return the following HTTP status codes:

HTTP status code	Meaning
200	MUST be returned if the resource is retrieved correctly.
404	Code 404 MUST be returned if a specific resource could not be found. This could for example be the result of a request containing a Participant Identifier that does not exist.
5xx	5xx codes MUST be returned if the service experiences an internal processing error.

An SMP implementation MAY support other HTTP status codes as well.

An SMP implementation MUST NOT use redirection in the manner indicated by the HTTP 3xx codes. Clients are not required to support active redirection.

An SMP implementation SHOULD respond in accordance with **[RFC7231]** to a request using the HTTP HEAD method.

5.2.2 Caching of HTTP responses

When using HTTP for SMP lookup, client-side caching MAY be introduced using headers "Last-Modified" and "If-Modified-Since" as defined in **[RFC7232]**. An SMP server MAY implement support of caching, and an SMP client MAY implement caching in case it is supported by an SMP server. Implementing caching or support of caching MUST NOT be imposed. Strategy for invalidation is not specified here, and MUST be implemented in accordance with **[RFC7232]**.

Only the "Last-Modified" and "If-Modified-Since" headers are supported for caching SMP responses. No other HTTP headers in **[RFC7232]** or elsewhere are used for client-side caching in SMP.

Sample of "Last-Modified" in response (server):

```
Last-Modified: Tue, 01 Dec 2018 19:14:44 GMT
```

Sample of "If-Modified-Since" in request (client):

```
If-Modified-Since: Tue, 01 Dec 2018 19:14:44 GMT
```

5.3 The use of XML and encoding

XML documents returned by HTTP GET MUST be well-formed according to **[XML 1.0]** and MUST be UTF-8 encoded (**[Unicode]**). They MUST contain an XML declaration starting with "<?xml" that includes the encoding attribute set to UTF-8.

5.4 Resources

The REST interface comprises 2 types of resources:

Resource	URI	Me-thod	XML resource root element	HTTP Status	Description of returned content
ServiceGroup	./bdxr-smp-2/[{identifier scheme}::]{participant id} See section 3.6 for {participant id} format	GET	<ServiceGroup>	200; 500; 404	Holds the Participant Identifier of the recipient, and a list of references to individual ServiceMetadata resources that are associated with that participant identifier.
ServiceMetadata	./bdxr-smp-2/[{identifier scheme}::]{participant id}/services/{service ID} See section 3.7 for {service ID} format	GET	<ServiceMetadata>	200; 500; 404	Holds all of the metadata about a Service, or a redirection URL to another Service Metadata Publisher holding this information.

Note that the resources MAY reside in the root directory of the SMP server, but MAY also instead reside in any subdirectory of the SMP server.

5.5 Referencing the SMP REST binding

For referencing the SMP REST binding, for example from Business Document Metadata Service Location records, the following identifier SHOULD be used for the version 2 of the SMP REST binding:

oasis-bdxr-smp-2

5.6 Security

5.6.1 General

At the transport level a Service Metadata Publishing service MAY either be secured or unsecured depending on the specific requirements and policies of a business document exchange infrastructure. Likewise, client-side authentication MAY be supported by a Service Metadata Publishing service pending infrastructure requirements and policies.

5.6.2 Message signature

5.6.2.1 Use of XML signatures

The message returned by a Service Metadata Publisher service MAY be signed using one or more **[XML-DSIG1]** XML-Signatures.

The signature MUST be an enveloped XML signature represented via a <ds:Signature> element embedded in the ServiceGroup element or in the ServiceMetadata element. The <ds:Signature> element MUST be constructed according to the following rules:

- The <Reference> MUST use exactly one Transform http://www.w3.org/2000/09/xmldsig#enveloped-signature;
- The <ds:KeyInfo> element MUST contain a <ds:X509Data> element with a <ds:X509Certificate> sub-element containing the signer's X.509 certificate as Base64 encoded X509 DER value;
- The canonicalization algorithm MUST be http://www.w3.org/2006/12/xml-c14n11;
- The SignatureMethod SHOULD be http://www.w3.org/2001/04/xmldsig-more#rsa-sha256;
- The DigestMethod SHOULD be http://www.w3.org/2001/04/xmlenc#sha256.

5.6.2.2 Verifying the signature

When verifying the signature, the SMP client has access to the full certificate as a Base64 encoded X509 DER value within the <Signature> element. The consumer MAY verify the signature by

a) extracting the certificate from the <ds:X509Data> element,
b) verify that it has been issued by the trusted root,
c) perform a validation of the signature, and
d) perform the required certificate validation steps (which might include checking expiration/activation dates and revocation lists).

5.6.2.3 Verifying the signature of the destination SMP

For the redirect scheme, the destination SMP signing certificate SHOULD be stored at the redirecting SMP. In addition to the regular signature validation performed by the client of the destination SMP resources, the client SHOULD also validate that the identifier of the destination SMP signing certificate corresponds to the unique identifier which the redirecting SMP claims belongs to the destination SMP.

5.6.2.4 XAdES

[XAdES] defines a set of forms that extends XMLDSig and allows adding some validation data to the signature. A compliant implementation of XAdES guarantees wide acceptance in implementing legal regulations, and supports signature validation best practices in general.

XAdES schema files have been added to the SMP distribution as a convenience and users MAY choose to include an OPTIONAL XAdES extension inside of a digital signature. The presence of these schema files does not oblige the use of XAdES.

6 Conformance

A Service Metadata Publishing implementation exhibits core conformance when complying with all of the following criteria:

01 The implementation MUST NOT violate any document constraints expressed by the schemas ServiceGroup-2.0.xsd and ServiceMetadata-2.0.xsd
02 When using digital signatures, the signing and verification must strictly adhere to the rules defined in section 5.6.2
03 The service discovery process MUST be executed as defined in section 2.1
04 The implementation MUST implement the REST binding as defined in sections 5.2, 5.3, 5.4 and 5.5 of this specification

This specification allows extensions. The use of extensions SHALL NOT contradict nor cause non-conformance with this specification.

Appendix A. ServiceGroup example (non-normative)

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceGroup xmlns:smb="http://docs.oasis-
open.org/bdxr/ns/SMP/2/BasicComponents"
              xmlns:ext="http://docs.oasis-
open.org/bdxr/ns/SMP/2/ExtensionComponents"
              xmlns:sma="http://docs.oasis-
open.org/bdxr/ns/SMP/2/AggregateComponents"
              xmlns="http://docs.oasis-open.org/bdxr/ns/SMP/2/ServiceGroup">
    <smb:SMPVersionID>2.0</smb:SMPVersionID>
    <smb:ParticipantID schemeID="iso6523-actorid-
upis">9908:810418052</smb:ParticipantID>
    <sma:ServiceReference>
        <smb:ID schemeID="bdx-docid-
qns">urn:oasis:names:specification:ubl:schema:xsd:Invoice-
2::Invoice##urn:www.cenbii.eu:transaction:biitrns010:ver2.0:extended:urn:www.pe
ppol.eu:bis:peppol5a:ver2.0:extended:urn:www.difi.no:ehf:faktura:ver2.0::2.1</s
mb:ID>
        <sma:Process>
            <smb:ID schemeID="cenbii-procid-
ubl">urn:www.cenbii.eu:profile:bii05:ver2.0</smb:ID>
        </sma:Process>
    </sma:ServiceReference>
    <sma:ServiceReference>
        <smb:ID schemeID="bdx-docid-
qns">urn:oasis:names:specification:ubl:schema:xsd:CreditNote-
2::CreditNote##urn:www.cenbii.eu:transaction:biitrns014:ver2.0:extended:urn:www
.peppol.eu:bis:peppol5a:ver2.0:extended:urn:www.difi.no:ehf:kreditnota:ver2.0::
2.1</smb:ID>
        <sma:Process>
            <smb:ID schemeID="cenbii-procid-
ubl">urn:www.cenbii.eu:profile:bii05:ver2.0</smb:ID>
        </sma:Process>
    </sma:ServiceReference>
</ServiceGroup>
```

Appendix B. ServiceMetadata example (non-normative)

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ServiceMetadata xmlns:smb="http://docs.oasis-
open.org/bdxr/ns/SMP/2/BasicComponents" xmlns:ext="http://docs.oasis-
open.org/bdxr/ns/SMP/2/ExtensionComponents" xmlns:sma="http://docs.oasis-
open.org/bdxr/ns/SMP/2/AggregateComponents" xmlns="http://docs.oasis-
open.org/bdxr/ns/SMP/2/ServiceMetadata">
  <smb:SMPVersionID>2.0</smb:SMPVersionID>
  <smb:ID schemeID="bdx-docid-
qns">urn:oasis:names:specification:ubl:schema:xsd:Invoice-
2::Invoice##urn:www.cenbii.eu:transaction:biitrns010:ver2.0:extended:urn:www.pe
ppol.eu:bis:peppol5a:ver2.0:extended:urn:www.difi.no:ehf:faktura:ver2.0::2.1</s
mb:ID>
  <smb:ParticipantID schemeID="iso6523-actorid-
upis">9908:810418052</smb:ParticipantID>
  <sma:ProcessMetadata>
    <sma:Process>
      <smb:ID schemeID="cenbii-procid-
ubl">urn:www.cenbii.eu:profile:bii05:ver2.0</smb:ID>
    </sma:Process>
    <sma:Endpoint>
      <smb:TransportProfileID>bdx-transport-as2-ver1p0</smb:TransportProfileID>
      <smb:Description>contact@example.com</smb:Description>
      <smb:Contact>Access point for testing</smb:Contact>
      <smb:AddressURI>https://ap.example.com/as2</smb:AddressURI>
      <smb:ActivationDate>2018-04-12</smb:ActivationDate>
      <smb:ExpirationDate>2020-04-12</smb:ExpirationDate>
      <sma:Certificate>
        <smb:Description>CN=EXAMPLE AP,C=NO</smb:Description>
        <smb:ActivationDate>2018-04-12</smb:ActivationDate>
        <smb:ExpirationDate>2020-04-12</smb:ExpirationDate>
        <smb:ContentBinaryObject mimeCode="application/base64">
          MIICwDCCAaigAwIBAgIEWs7kiDANBgkqhkiG9w0BAQsFADAiMQswCQYDVQQGEwJO
          TzETMBEGA1UEAwwKRVhBTVBMRSBBUDAeFw0xODA0MTIwNDQ2MDBaFw0yMDA0MTIw
          NDQ2MDBaMCIxCzAJBgNVBAYTAk5PMRMwEQYDVQQDDApFWEFNUExFIEFQMIIBIjAN
          BgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAtuG5qwA2sNvC9dj4purG8hkSVB9p
          CWVHyO9buRrSCC+r2UxSF7Lnmr8Hjii0uIdJeFyYv0Vj9d4CjpYyEeYU2QG96wi+
          w2KdE28HMZFNMwy0iV9vIkbq0esJCcAXQ4C3rPQ4e9F1Tw4oKnS6rEWCw8i8lsKE
          iS/dzIFUa/BVtgjqHvs3siON4k1Y7BU93rZViz8ZM6LB4eA7rYU4le6a8rGKbHa6
          BMSIoKgBuKr8XY9lKb2OVg65+LWTESpPkKiKRikzZhAw+mEVYmljvFwBLSp0IEKW
          3qogVYGA0jz+kWWGOChk58SKsnqhbANoIucz+axxoJOLl0A5328qM7aRTwIDAQAB
          MA0GCSqGSIb3DQEBCwUAA4IBAQBW7VCYhUmRR7xW+QOhUxI//ISjupDdcQ/Jl7hH
          CrUghjL7FmNnJKNqBAwrxcAfdwXwRltWzNT9E1btekfyw4+QL34w20kZ7SNLioZU
          lxVviaoLsf0f70TMPOBGGv/uyV26l5VMBK40FXvcFwDQ5VNiJOYrsxpF//Hh/t76
          QMij6glyLUmYA1aS9Am0zAB5ld+U7HtJAEL6SXinPrPDR1ofcRgx3FzY5pq0PCn9
          EA005L6X4eGkI3HqwpcDzYwDC29pPSfnNP50khfFJMCnT6kKhCkPJYQhcZexGJ2U
          Ad5OU7Gui/WnmjM80x9qHBv2RIIQggpMy838WjPbw11gMOo+
        </smb:ContentBinaryObject>
      </sma:Certificate>
    </sma:Endpoint>
  </sma:ProcessMetadata>
</ServiceMetadata>
```

Appendix C. Major changes from SMP 1.0 (non-normative)

SMP 2.0 contains a number of improvements over the previous version 1.0. In the revision of the specification, the BDXR Technical Committee has been receiving feedback from existing SMP users and communities as well as from potential users in order produce a work product that fully responds to current and future expectations.

The key changes from the previous SMP version are in improved security in general, as well as in a more complete and elaborate data model, specifically:

- The previous static XML data model has been refactored to be more flexible and modular, so as to support a wider range of business scenarios. In the refactoring of the data model we have introduced new features, such as the inclusion of participant roles and the support of multiple certificates.
- The XML data model is now building on the UN/CEFACT Core Component Technical Specification (**[CCTS]**) to align with other XML implementations and to make implementation easier by reusing existing building blocks.
- The extension model has been improved to align with other OASIS work products.

The REST model is left mainly unchanged to facilitate an easy transformation from SMP 1.0 to SMP 2.0 for existing users, however the terminology has been changed to enhance the understanding of users working with services and other resources different from “documents”. While in that process, a new identifier scheme for “services” has been introduced in parallel to the existing `bdx-docid-qns` used for documents.

Appendix D. Acknowledgments (non-normative)

The following individuals have participated in the creation of this specification and are gratefully acknowledged:

Participants:
Jens Aabol, Difi-Agency for Public Management and eGovernment
Todd Albers, Federal Reserve Bank of Minneapolis
Oriol Bausa Peris, Individual
Kenneth Bengtsson, Individual
Erlend Klakegg Bergheim, Difi-Agency for Public Management and eGovernment
Mikkel Brun, Tradeshift Network Ltd.
Ger Clancy, IBM
Kees Duvekot, RFS Holland Holding B.V.
Pim van der Eijk, Sonnenglanz Consulting
Sander Fieten, Individual
Martin Forsberg, Swedish Association of Local Authorities & Regions
Ken Holman, Crane Softwrights Ltd.
Levine Naidoo, Individual
Klaus Pedersen, Difi-Agency for Public Management and eGovernment
Sven Rasmussen, Danish Agency for Digitisation, Ministry of Finance
Steven Ryan, Individual
Matt Vickers, Xero