



Service Metadata Publishing (SMP) Version 1.0

Committee Specification 03

30 June 2016

Specification URIs

This version:

<http://docs.oasis-open.org/bdxx/bdx-smp/v1.0/cs03/bdx-smp-v1.0-cs03.doc> (Authoritative)
<http://docs.oasis-open.org/bdxx/bdx-smp/v1.0/cs03/bdx-smp-v1.0-cs03.html>
<http://docs.oasis-open.org/bdxx/bdx-smp/v1.0/cs03/bdx-smp-v1.0-cs03.pdf>

Previous version:

<http://docs.oasis-open.org/bdxx/bdx-smp/v1.0/csprd02/bdx-smp-v1.0-csprd02.doc> (Authoritative)
<http://docs.oasis-open.org/bdxx/bdx-smp/v1.0/csprd02/bdx-smp-v1.0-csprd02.html>
<http://docs.oasis-open.org/bdxx/bdx-smp/v1.0/csprd02/bdx-smp-v1.0-csprd02.pdf>

Latest version:

<http://docs.oasis-open.org/bdxx/bdx-smp/v1.0/bdx-smp-v1.0.doc> (Authoritative)
<http://docs.oasis-open.org/bdxx/bdx-smp/v1.0/bdx-smp-v1.0.html>
<http://docs.oasis-open.org/bdxx/bdx-smp/v1.0/bdx-smp-v1.0.pdf>

Technical Committee:

OASIS Business Document Exchange (BDXR) TC

Chair:

Kenneth Bengtsson (kenneth@alfa1lab.com), Alfa1lab

Editors:

Jens Aabol (jea@difi.no), Difi-Agency for Public Management and eGovernment
Kenneth Bengtsson (kenneth@alfa1lab.com), Alfa1lab
Erlend Klakegg Bergheim (erlend.klakegg.bergheim@difi.no), Difi-Agency for Public Management and eGovernment
Sander Fieten (sander@fieten-it.com) Individual
Sven Rasmussen (svrra@digst.dk), Danish Agency for Digitisation, Ministry of Finance

Additional artifacts:

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

- XML schema: <http://docs.oasis-open.org/bdxx/bdx-smp/v1.0/cs03/schemas/bdx-smp-201605.xsd>

Related work:

This specification is related to:

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

Declared XML namespace:

- <http://docs.oasis-open.org/bdxx/ns/SMP/2016/05>

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 (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.

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 version" 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/bdxt/>.

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/bdxt/ipr.php>).

Citation format:

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

[SMP-v1.0]

Service Metadata Publishing (SMP) Version 1.0. Edited by Jens Aabol, Kenneth Bengtsson, Erlend Klakegg Bergheim, Sander Fieten, and Sven Rasmussen. 30 June 2016. OASIS Committee Specification 03. <http://docs.oasis-open.org/bdxt/bdx-smp/v1.0/cs03/bdx-smp-v1.0-cs03.html>. Latest version: <http://docs.oasis-open.org/bdxt/bdx-smp/v1.0/bdx-smp-v1.0.html>.

Notices

Copyright © OASIS Open 2016. All Rights Reserved.

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 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, 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.

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 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 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 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, 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, and should be used only to refer to the organization and its official outputs. OASIS welcomes reference to, and implementation and use of, specifications, 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.....	6
1.1	Introduction.....	6
1.2	Goals and non-goals.....	6
1.3	Terminology.....	6
1.4	Normative References.....	6
1.5	Non-Normative References.....	7
2	SMP Protocol.....	8
2.1	The Service Discovery Process.....	8
2.1.1	Introduction.....	8
2.1.2	Discovering services associated with a Participant Identifier.....	8
2.1.3	Service Metadata Publisher Redirection.....	9
2.2	Interface model.....	9
2.3	Data model.....	10
2.3.1	Data model.....	10
2.3.2	On extension points.....	10
2.3.2.1	Use of extensions.....	10
2.3.2.2	Pseudo-schema for Extension.....	10
2.3.2.3	Description of the individual fields (elements and attributes).....	11
2.3.3	ServiceGroup.....	11
2.3.3.1	ServiceGroup.....	11
2.3.3.2	Pseudo-schema for ServiceGroup.....	11
2.3.3.3	Description of the individual fields (elements and attributes).....	12
2.3.3.4	Non-normative example of a ServiceGroup resource.....	12
2.3.4	ServiceMetadata.....	12
2.3.4.1	Redirection.....	12
2.3.4.2	Pseudo-schema for the “ServiceInformation” data type.....	13
2.3.4.3	Pseudo-schema for the “Redirect” data type.....	13
2.3.4.4	Description of the individual fields (elements and attributes).....	14
2.3.5	SignedServiceMetadata.....	16
2.4	Identifiers.....	16
2.4.1	Introduction.....	16
2.4.2	Notational conventions.....	16
2.4.3	On the use of percent encoding in URLs.....	17
2.4.4	On Scheme Identifiers.....	17
2.4.5	Participant Identifiers.....	17
2.4.5.1	Participant Identifiers.....	17
2.4.5.2	XML format for Participant Identifiers.....	17
2.4.5.3	Using participant identifiers in URLs.....	18
2.4.6	Document Identifier.....	18
2.4.6.1	Introduction.....	18
2.4.6.2	XML Representation of Document Identifiers.....	19
2.4.6.3	URL representation of Document Identifiers.....	19
2.4.7	Process Identifiers.....	20
2.4.7.1	XML Representation of Process Identifiers.....	20

3	Service Metadata Publishing REST binding	22
3.1	Introduction	22
3.2	The use of HTTP	22
3.2.1	General use of HTTP	22
3.2.2	Caching of HTTP responses	22
3.3	The use of XML and encoding.....	23
3.4	Resources and identifiers	23
3.4.1	REST interface	23
3.4.2	Using identifiers in the REST Resource URLs.....	23
3.5	Referencing the SMP REST binding	23
3.6	Security	24
3.6.1	General.....	24
3.6.2	Message signature	24
3.6.2.1	Use of XML signatures.....	24
3.6.2.2	Verifying the signature	24
3.6.2.3	Verifying the signature of the destination SMP	24
3.7	Schema for the REST interface.....	24
4	Conformance	25
Appendix A.	Acknowledgments (non-normative)	26
Appendix B.	SMP Schema (non-normative).....	27
Appendix C.	Non-Normative Examples	31
C.1	ServiceGroup resource	31
C.2	SignedServiceMetadata resource	31
C.3	Redirect	33
C.4	Identifier.....	34

1 Introduction

1.1 Introduction

This document describes the protocol and its binding to a REST interface for Service Metadata Publication within a 4-corner network. It defines the messages exchanged between a Service Metadata Publisher and a client wishing to discover endpoint information. A client can either be an end-user business application or an Access Point in a 4-corner network.

It also specifies how these message exchanges should be implemented using a REST transport interface. The SMP protocol itself however is open for binding to other transport protocols like AS4. Such bindings can be specified in future specifications.

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 label “document” used in the data model of this specification may in such networks be interpreted as referring to any resource that is being exchanged in the network.

1.2 Goals and non-goals

The goal of this document is to define the protocol and its binding to a REST interface that Service Metadata Publishers (“SMP”) and clients must support. Decisions regarding physical data format and management interfaces are left to implementers of the SMP and client applications.

Service Metadata Publishers may be subject to additional constraints of agreements and governance frameworks within instances of the 4-corner infrastructure not covered in this specification, which only addresses the technical interface of such a service.

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]** “Key words for use in RFCs to Indicate Requirement Levels”, BCP 14, RFC 2119, March 1997. <http://www.ietf.org/rfc/rfc2119.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/>
- [HTTP 1.1]** “Hypertext Transfer Protocol -- HTTP/1.1”, RFC 2616, June 1999, www.ietf.org/rfc/rfc2616.txt
- [X509v3]** ITU-T Recommendation X.509 version 3 (1997). "Information Technology - Open Systems Interconnection - The Directory Authentication Framework" ISO/IEC 9594-8:1997
- [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-xmlsig-core1-20130411/>. Latest version available at <http://www.w3.org/TR/xmlsig-core1/>.
- [RFC7232]** “Hypertext Transfer Protocol (HTTP/1.1): Conditional Requests”, RFC 7232, June 2014, <http://www.ietf.org/rfc/rfc7232.txt>

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>
- [RFC3986]** Berners-Lee, T., Fielding, R., Masinter, L., “Uniform Resource Identifier (URI): Generic Syntax”, RFC 3968, January 2005, <http://tools.ietf.org/rfc/rfc3986>

2 SMP Protocol

2.1 The Service Discovery Process

2.1.1 Introduction

In a 4-cornered architecture the discovery process is a two-step process that starts with the lookup of the SMP that holds the service meta-data information about a participant identifier in the network. Each Participant Identifier is registered with one and only one Service Metadata Publisher. This lookup is performed by the client using the Business Document Metadata Service Location protocol [BDXL].

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

The diagram below represents the lookup flow for a sender contacting both the Business Document Metadata Service Location and the SMP:

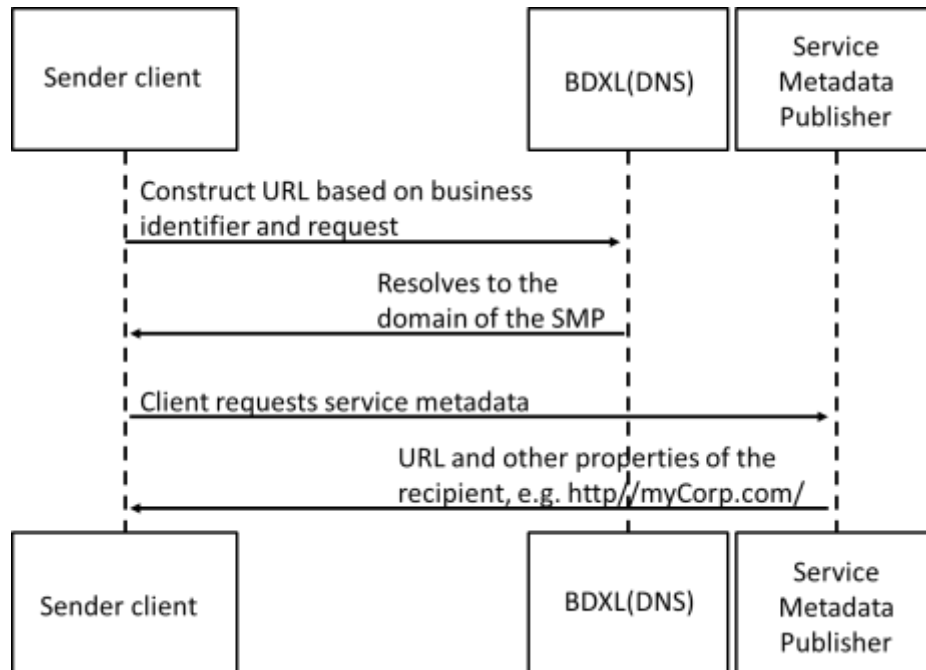


Fig.1. Endpoint lookup with Service Metadata

Note: For optimization reasons, the discovery doesn't have to be performed for every transfer if the necessary information for transfer is already cached from previous sending's. Though necessary exception handling has to be in place i.e. new lookup has to be performed if the sending shows that information is outdated e.g. old endpoint address.

2.1.2 Discovering services associated with a Participant Identifier

In addition to the direct lookup of Service Metadata based on Participant Identifier and document type, a sender may want to discover what document types can be handled by a specific Participant Identifier. Such discovery is relevant for applications supporting several equivalent business processes. Knowing the capabilities of the recipient is valuable information to a sender application and ultimately to an end

user. E.g. the end user may be presented with a choice between a “simple” and a “rich” business process.

This is enabled by a pattern where the sender first retrieves the ServiceGroup entity, which holds a list of references to the ServiceMetadata resources associated with it. The SignedServiceMetadata in turn holds the metadata information that describes the capabilities associated with the recipient Participant identifier.

2.1.3 Service Metadata Publisher Redirection

For each Participant identifier, the Business Document Metadata Service Location may only point to a single Service Metadata Publisher. There are cases however where the owner of a Participant Identifier may 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 SignedServiceMetadata can be found. A special element within the SignedServiceMetadata record of the SMP points to the SMP that has the actual Service Metadata and certificate information for that SMP. The diagram below shows this flow:

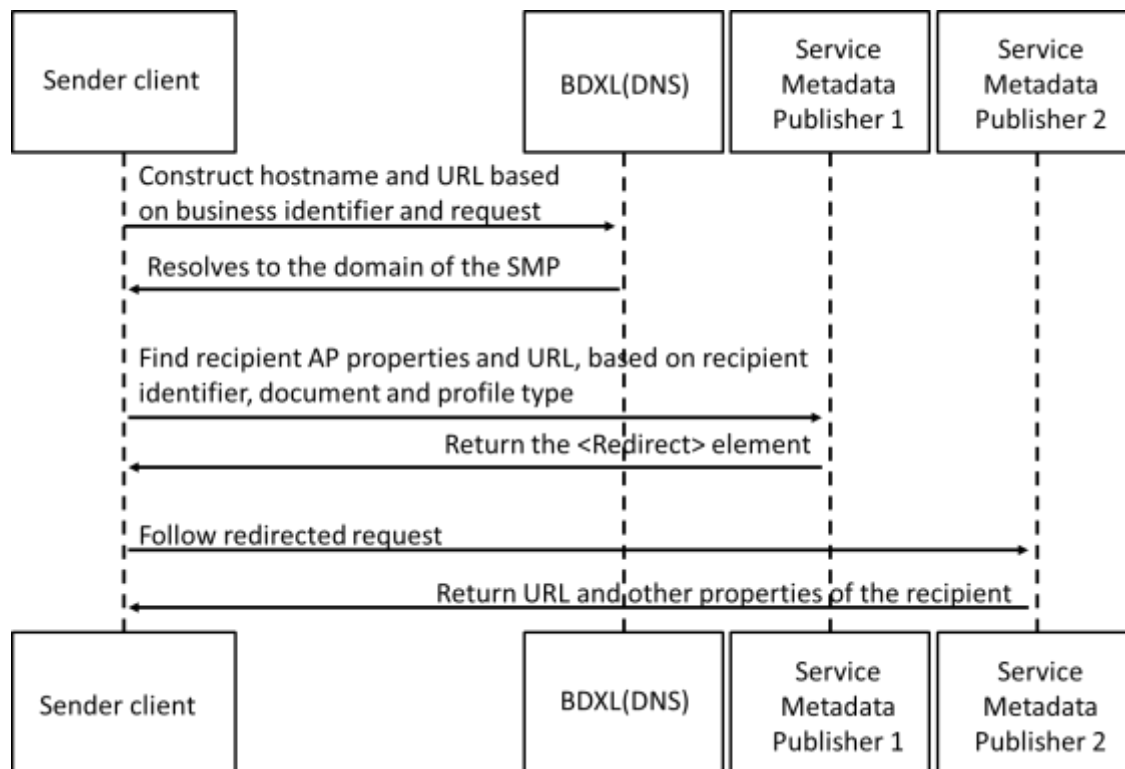


Fig. 2: Service Metadata Redirection

Note that only one degree of redirect is allowed; clients are not required to follow more than one redirect, i.e. a redirect resource cannot point to another redirect resource. Allowing one level of redirect permits the described use case to be realized, while avoiding the possibility of cyclic references and long chains of redirects.

2.2 Interface model

This specification only defines the protocol for retrieving Service Metadata, it does not specify interfaces for creating, updating, deleting and managing Service Metadata, or any internal data storage formats.

The goal is to allow the interface in this specification to expose data from many different Service Metadata back-ends, which may be based on any suitable technology such as for example RDBMS, LDAP or UDDI.

2.3 Data model

2.3.1 Data model

This section outlines the data model of the interface. The data model comprises the following main data types:

- ServiceGroup
- ServiceMetadata / SignedServiceMetadata

Supporting data types for these main types are:

- ServiceInformation
- ServiceEndpointList
- ParticipantIdentifier
- DocumentIdentifier
- Redirect
- Process
- ProcessList
- Endpoint

Each of these data types is described in detail in the following sections.

2.3.2 On extension points

2.3.2.1 Use of extensions

For each major entity, extension points have been added with the optional <Extension> element. Semantics and use child elements of the <Extension> 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 required.
- 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).

2.3.2.2 Pseudo-schema for Extension

```
<Extension>
  <ExtensionID>xs:token</ExtensionID>?
  <ExtensionName>xs:string</ExtensionName>?
```

```

<ExtensionAgencyID>xs:string</ExtensionAgencyID>?
<ExtensionAgencyName>xs:string</ExtensionAgencyName>?
<ExtensionAgencyURI>xs:anyURI</ExtensionAgencyURI>?
<ExtensionVersionID>xs:normalizedString</ExtensionVersionID>?
<ExtensionURI>xs:anyURI</ExtensionURI>?
<ExtensionReasonCode>xs:token</ExtensionReasonCode>?
<ExtensionReason>xs:string</ExtensionReason>?
xs:any
</Extension>

```

2.3.2.3 Description of the individual fields (elements and attributes)

Field	Description
ExtensionID	An identifier for the Extension assigned by the creator of the extension.
ExtensionName	A name for the Extension assigned by the creator of the extension.
ExtensionAgencyID	An agency that maintains one or more Extensions.
ExtensionAgencyName	The name of the agency that maintains the Extension.
ExtensionAgencyURI	A URI for the Agency that maintains the Extension.
ExtensionVersionID	The version of the Extension.
ExtensionURI	A URI for the Extension.
ExtensionReasonCode	A code for reason the Extension is being included.
ExtensionReason	A description of the reason for the Extension.

2.3.3 ServiceGroup

2.3.3.1 ServiceGroup

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 ServiceMetadataReferenceCollection structure.

2.3.3.2 Pseudo-schema for ServiceGroup

```

<ServiceGroup>
  <ParticipantIdentifier [scheme="xs:string"]>xs:string
</ParticipantIdentifier>
  <ServiceMetadataReferenceCollection>
    <ServiceMetadataReference href="xs:anyURI" />*
  </ServiceMetadataReferenceCollection>
  <Extension />*
</ServiceGroup>

```

2.3.3.3 Description of the individual fields (elements and attributes)

Field	Description
ServiceGroup	Document element
ParticipantIdentifier	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. See the ParticipantIdentifier section 2.4.5 of this specification for information on this data type.
ServiceMetadataReferenceCollection	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. See the SignedServiceMetadata section 2.3.5 of this specification for information on this data type.
ServiceMetadataReference/@href (0..*)	Contains the URL to a specific SignedServiceMetadata instance - see the REST binding section 3.4 for details on the URL format. 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. Resolving URLs referenced by a ServiceMetadataReference MUST be done in accordance with [RFC3986].
Extension	The extension element may contain any XML element. Clients MAY ignore this element (see section 2.3.2). It can be used to add extended metadata to the ServiceGroup structure.

2.3.3.4 Non-normative example of a ServiceGroup resource

See Appendix C.

2.3.4 ServiceMetadata

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.

2.3.4.1 Redirection

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.

Pseudo-schema for this data type:

```
<ServiceMetadata>
  [<ServiceInformation /> | <Redirect />]
</ServiceMetadata>
```

2.3.4.2 Pseudo-schema for the “ServiceInformation” data type

```
<ServiceInformation>
  <ParticipantIdentifier [scheme="xs:string"] />
  <DocumentIdentifier [scheme="xs:string"] />
  <ProcessList>
    <Process>+
      <ProcessIdentifier [scheme="xs:string"] />
      <ServiceEndpointList>
        <Endpoint transportProfile="xs:string">+
          <EndpointURI>xs:anyURI</EndpointURI>
          <RequireBusinessLevelSignature>xs:boolean
          </RequireBusinessLevelSignature?>
          <MinimumAuthenticationLevel>xs:string
          </MinimumAuthenticationLevel?>
          <ServiceActivationDate>xs:dateTime
          </ServiceActivationDate?>
          <ServiceExpirationDate>xs:dateTime
          </ServiceExpirationDate?>
          <Certificate>xs:string</Certificate>
          <ServiceDescription>xs:string
          </ServiceDescription>
          <TechnicalContactUrl>xs:anyURI
          </TechnicalContactUrl>
          <TechnicalInformationUrl>xs:anyURI
          </TechnicalInformationUrl?>
          <Extension />*
        </Endpoint>
      </ServiceEndpointList>
      <Extension />*
    </Process>
  </ProcessList>
  <Extension />*
</ServiceInformation>
```

2.3.4.3 Pseudo-schema for the “Redirect” data type

```
<Redirect href="xs:anyURI">
  <CertificateUID>xs:string</CertificateUID>
  <Extension />*
</Redirect>
```

The REQUIRED «href» attribute of the Redirect element contains the full address of the destination SMP record that the client is redirected to.

For example, assume that an SMP called "SMP1" has the address "http://smp1.eu", and another SMP called "SMP2" has the address "http://smp2.eu", and a client requests a resource with the following URL:

```
http://smp1.eu/urn%3Aoasis%3Anames%3Aatc:ebcore%3Apartyid-
type%3Aiso6523%3A0010%3A%3A5798000000001/services/bdx-docid-
qns%3A%3Aurn%3Aoasis%3Anames%3Aspecification%3Aubl%3Aschema%3Axsd%3AInv
oice-2%3A%3AInvoice%23%23UBL-2.0
```

We now assume that the owner of these metadata has moved them to SMP2. SMP1 would then return a SignedServiceMetadata resource with a Redirect child element that has the “href” attribute set to:

*http://smp2.eu/urn%3Aoasis%3Anames%3Atc:ebcore%3Apartyid-
type%3Aiso6523%3A0010%3A%3A5798000000001/services/bdx-docid-
qns%3A%3Aurn%3Aoasis%3Anames%3Aspecification%3Aubl%3Aschema%3Axsd%3AInvoice-2%3A%3AInvoice%23%23UBL-2.0*

For the list of endpoints under each <Endpoint> element in the ServiceEndpointList, each endpoint MUST have different values of the transportProfile attribute, i.e. represent bindings to different transports.

2.3.4.4 Description of the individual fields (elements and attributes)

Field	Description
ServiceMetadata	Document element
Redirect	The direct child element of ServiceMetadata is either the Redirect element or the ServiceInformation element. The Redirect element indicates that a client must follow the URL of the href attribute of this element.
Redirect/CertificateUID	Holds the Subject Unique Identifier of the certificate of the destination SMP. A client SHOULD validate that the Subject Unique Identifier of the certificate used to sign the resource at the destination SMP matches the Subject Unique Identifier published in the redirecting SMP.
Redirect/Extension	The extension element may contain any XML element. Clients MAY ignore this element (see section 2.3.2). It can be used to add extension metadata to the Redirect.
ServiceInformation	The direct child element of ServiceMetadata is either the Redirect element or the ServiceInformation element. The ServiceInformation element contains service information for an actual service registration, rather than a redirect to another SMP.
ServiceInformation/ ParticipantIdentifier	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. See the ParticipantIdentifier section 2.4.5 of this specification for information on this data type.
ServiceInformation/ DocumentIdentifier	Represents the type of document that the recipient is able to receive. The document is represented by an identifier (identifying the document type) and an identifier scheme. See the DocumentIdentifier section 2.4.6 of this specification for information on this data type.
ServiceInformation/ ProcessList	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.
ProcessList/Process/ ProcessIdentifier	The identifier of the process. A process is identified by a string that is defined outside of this specification. For example, the CEN workshop on business

	<p>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".</p> <p>This document just defines one process identifier, which represents documents that are not sent under any specific process:</p> <p style="text-align: center;"><code>bdx:noprocess</code></p> <p>A process identifier specification or 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.</p> <p>See the Process section 2.4.7 of this specification for information on the identifier format.</p>
ProcessList/Process/ServiceEndpointList	List of one or more endpoints that support this process.
ServiceInformation/ProcessList/./Endpoint	Endpoint represents the technical endpoint and address type of the recipient, as an URL.
ServiceEndpointList/Endpoint/EndpointURI	The address of an endpoint, as a URL
ServiceInformation/ProcessList/./Endpoint/@transportProfile	Indicates the type of transport method that is being used between access points
ServiceInformation/ProcessList/./Endpoint/RequireBusinessLevelSignature	<p>Set to "true" if the recipient requires business-level signatures for the message, meaning a signature applied to the business message before the message is put on the transport. This is independent of the transport-level signatures that a specific transport profile might mandate. This flag does not indicate which type of business-level signature might be required. Setting or consuming business-level signatures would typically be the responsibility of the final senders and receivers of messages, rather than a set of gateways.</p> <p>Including the RequireBusinessLevelSignature element is optional. If not present, its default handling is "false", meaning that the recipient does not require a business-level signature for the message.</p>
ServiceInformation/ProcessList/./Endpoint/MinimumAuthenticationLevel	Indicates the minimum authentication level that recipient requires. The specific semantics of this field is defined in a specific instance of a 4-corner infrastructure.
ServiceInformation/ProcessList/./Endpoint/ServiceActivationDate	Activation date of the service. Senders should ignore services that are not yet activated. Data type of ServiceActivationDate date is <code>xs:dateTime</code> .
ProcessList/./Endpoint/ServiceExpirationDate	Expiration date of the service. Senders should ignore services that are expired. Data type of ServiceExpirationDate date is <code>xs:dateTime</code> .
ProcessList/./Endpoint/Certificate	Holds the complete [X509v3] signing certificate of the recipient gateway, as a base 64 encoded DER formatted value.

ProcessList/./Endpoint/ ServiceDescription	A human readable description of the service
ProcessList/./Endpoint/ TechnicalContactUrl	Represents a link to human readable contact information. This might also be an email address.
ProcessList/./Endpoint/ TechnicalInformationUrl	A URL to human readable documentation of the service format. This could for example be a web site containing links to XML Schemas, WSDLs, Schematrons and other relevant resources.
ProcessList/./Endpoint/Extension	The extension element may contain any XML element. Clients MAY ignore this element (see section 2.3.2). It can be used to add extension metadata to the endpoint metadata.
Process/Extension	The extension element may contain any XML element. Clients MAY ignore this element (see section 2.3.2). It can be used to add extension metadata to the process metadata block as a whole.
ServiceInformation/ Extension	The extension element may contain any XML element. Clients MAY ignore this element (see section 2.3.2). It can be used to add extension metadata to the service metadata.

For a **non-normative example** of a ServiceMetadata resource, see the SignedServiceMetadata non-normative example in Appendix C.

2.3.5 SignedServiceMetadata

The SignedServiceMetadata structure is a ServiceMetadata structure that has been signed by the ServiceMetadataPublisher, according to governance policies that are not covered by this document. Pseudo-schema for this data type:

```
<SignedServiceMetadata>
  <ServiceMetadata />
  <ds:Signature />
</SignedServiceMetadata>
```

- **ServiceMetadata:** The ServiceMetadata element covered by the signature.
- **Signature** represents an enveloped XML signature over the SignedServiceMetadata element.

Non-normative example of a SignedServiceMetadata resource – see Appendix C.

2.4 Identifiers

2.4.1 Introduction

This section defines what participant business-, document- and process-identifiers are, and how they are represented in XML elements and URLs.

2.4.2 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 «0010:5798000000001», the format definition

```
/{identifier}/service[/docType]
```


Can be instantiated to either of the strings

```
/0010:5798000000001/service
```

And

```
/0010:5798000000001/service/docType
```

2.4.3 On the use of percent encoding in URLs

When any type of identifiers are used in URLs, each section between slashes MUST be percent encoded individually, i.e. section by section.

For example, this implies that for an URL in the form of `«/{identifier scheme}::{id}/services/{docType}»`, the slash literals MUST NOT be URL encoded.

2.4.4 On Scheme Identifiers

Identifier schemes for all schemed identifier types (participants, documents, profiles) 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.

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 the European PEPPOL network, `«iso6523-actorid-upis»`.

2.4.5 Participant Identifiers

2.4.5.1 Participant Identifiers

A “participant identifier” is a business level endpoint key that uniquely identifies an end-user entity (“participant”) in the 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. Here the *scheme identifier* indicates the specification of the participant identifier format, i.e. its representation and meaning.

2.4.5.2 XML format for Participant Identifiers

The `<ParticipantIdentifier>` element is used to represent participant identifiers and scheme information.

Pseudo-scheme for ParticipantIdentifier:

```
<ParticipantIdentifier [scheme="xs:string"]>xs:string
</ParticipantIdentifier>
```

Where the «scheme» attribute indicates the scheme of the participant identifier.

Field	Description
ParticipantIdentifier	A participant identifier which may be associated with a group of services.

ParticipantIdentifier/ @scheme	<p>The scheme of the participant identifier. This scheme indicates the format of the participant identifier (i.e. the textual format) – not its semantic type (e.g. DUNS or GLN).</p> <p>This scheme type MUST be in the form of a URI.</p> <p>A participant identifier scheme MAY define its own requirements for case sensitivity handling. Unless defined differently by the participant identifier scheme, a participant identifier in XML format MUST be treated as case insensitive.</p>
-----------------------------------	--

2.4.5.3 Using participant identifiers in URLs

The following format is used:

```
{identifier scheme}::{id}
```

Where «identifier scheme» is the format of the identifier, and «id» is the participant identifier itself, following the format indicated by the «identifier type» part.

In a URL, the string represented by «{identifier scheme}::{id}» MUST be percent encoded following and the guidelines given above.

Non-normative example using the [ebCorePartyId] URN format, assuming an ISO 6523 International Code Designator 0010 with the participant identifier «5798000000001»:

```
urn:oasis:names:tc:ebcore:partyid-type:iso6523:0010::5798000000001
```

In percent encoded form:

```
urn%3Aoasis%3Anames%3Atc%3Aebcore%3Apartyid-type%3Aiso6523%3A0010%3A%3A5798000000001
```

And the same non-normative example using the PEPPOL Universal Participant Identifier Format:

```
iso6523-actorid-upis::0010:5798000000001
```

In percent encoded form:

```
iso6523-actorid-upis%3A%3A0010%3A5798000000001
```

A participant identifier scheme MAY define its own requirements for case sensitivity handling. Unless defined differently by the participant identifier scheme, a participant identifier in a URL MUST be treated as case insensitive. Note that any surrounding slashes which belong to the URL rather than the various identifiers (which may take the forms of URLs) are *not* percent encoded.

2.4.6 Document Identifier

2.4.6.1 Introduction

Documents are represented by an identifier (identifying the document type) and a scheme type 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.

This specification defines a single identifier scheme, the “QName/Subtype Identifier Scheme”, which is identified by the following URI:

```
bdx-docid-qns
```

This scheme is based on a concatenation of the document namespace, root element, and optional (and document-dependent) subtype:

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

Where “[]” denotes an optional part of the identifier, and everything outside “{ }” are string literals.

For example, in the case of a UBL 2.0 order, this document can then be identified by

- 01 **Root namespace:** *urn:oasis:names:specification:ubl:schema:xsd:Order-2*
- 02 **Document element local name:** *Order*
- 03 **Subtype identifier:** *UBL-2.0 (since several versions of the Order schema may use the same namespace + document element name)*

The document type identifier will then be:

urn:oasis:names:specification:ubl:schema:xsd:Order-2::Order##UBL-2.0

2.4.6.2 XML Representation of Document Identifiers

The <DocumentIdentifier> element is used to represent document identifiers and scheme information.

Pseudo-scheme for DocumentIdentifier:

```
<DocumentIdentifier [scheme="xs:string"]>xs:string</DocumentIdentifier>
```

Where the «scheme» attribute indicates the scheme of the document identifier.

Field	Description
DocumentIdentifier	A document identifier representing a specific document type. In the case of a UBL 2.0 order document, this would be «urn:oasis:names:specification:ubl:schema:xsd:Order-2::Order##UBL-2.0».
DocumentIdentifier/ @scheme	The scheme of the document identifier. This scheme identifier MUST be in the form of a URI. This document defines the «QName/Subtype Identifier Scheme», which is identified by the following URI: <i>bdx-docid-qns</i> A document identifier scheme MAY define its own requirements for case sensitivity handling. Unless defined differently by the document identifier scheme, a document identifier in XML format MUST be treated as case insensitive.

2.4.6.3 URL representation of Document Identifiers

When representing document identifiers in URLs, the document identifier itself will be prefixed with the scheme identifier. For example, the «QName/Subtype Identifier Scheme» is indicated by this identifier:

«bdx-docid-qns»

The format of this is:

{identifier scheme}::{id}

In the case that the «QName/Subtype Identifier Scheme» is used, the complete format is:

{identifier scheme}::{rootNamespace}::{documentElementLocalName}{##{Subtype identifier}}

As a non-normative example, in the case of a UBL 2.0 order, this document can then be identified by:

- **Identifier scheme:** *bdx-docid-qns*
- **Root namespace:** *urn:oasis:names:specification:ubl:schema:xsd:Order-2*
- **Document element local name:** *Order*
- **Subtype identifier:** *UBL-2.0 (since several versions of the Order schema may use the same namespace + document element name)*

The document type identifier will then be:

`bdx-docid-qns::urn:oasis:names:specification:ubl:schema:xsd:Order-2::Order##UBL-2.0`

Rules for parsing this identifier:

- The text up until the first «::» is the identifier scheme identifier
- The text before the next to last «::» and last «::» is the root namespace
- The text between the last occurrence of «::» and last 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

A document identifier scheme MAY define its own requirements for case sensitivity handling. Note that although namespaces and element names are case sensitive, the document identifier MUST be treated as case insensitive unless defined differently by the document identifier scheme.

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%3AOrd  
er-2%3A%3AOrder%23%23UBL-2.0
```

Note the limitation that XML document types with the following characteristics MUST NOT be referenced using Service Metadata Publishing:

- Documents with only local names (i.e. without namespaces)
- Documents 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.
- Documents where the namespace URI contains «::».

2.4.7 Process Identifiers

A process identifier represents a process that a specific document type can participate in. Process identifiers consist of the process identifier itself, and a scheme or identifier format type. As for the other schemed identifier types, additional process identifier schemes may be defined outside of this specification.

2.4.7.1 XML Representation of Process Identifiers

Pseudo-schema for the ProcessIdentifier XML element:

```
<ProcessIdentifier [scheme="xs:string"]>xs:string</ProcessIdentifier>
```

Description of the individual fields (elements and attributes):

Field	Description
ProcessIdentifier	<p>The identifier of the process.</p> <p>A process is identified by a string that 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".</p> <p>This document just defines one process identifier, which represents documents that are not sent under any specific process:</p>

	<p><code>bdx:noprocess</code></p> <p>A process identifier specification a 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.</p>
<p>ProcessIdentifier/@scheme</p>	<p>Indicates the format of the process identifier. The format of this identifier may be different from domain to domain – for example, in a domain where BPEL definitions of processes exist, a technical identifier derived from the BPEL definitions may be used, whereas a taxonomy may be created in another domain. Processes (or profiles) defined by the CEN BII workshop could for example choose to use the identifier «cenbii-procid-ubl» to indicate the format of process identifiers.</p> <p>This document just defines one process scheme identifier, which represents transport-specific process identifiers:</p> <p><i>bdx-procid-transport</i></p> <p>Currently the only valid process identifier under this scheme is the identifier «bdx:noprocess», which indicates that a message is not sent under any named process.</p>

3 Service Metadata Publishing REST binding

3.1 Introduction

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

3.2 The use of HTTP

3.2.1 General use of HTTP

A service implementing the REST binding **MUST** support **[HTTP 1.1]**, and **MUST** set the HTTP “content-type” header and give it a value of “text/xml”. An instance of a 4-cornered infrastructure **MAY** set restrictions on what ports are allowed.

An implementation of the SMP might 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 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.
500	Code 500 must be returned if the service experiences an internal processing error.

The service **MAY** support other HTTP status codes as well.

The service **SHOULD NOT** use redirection in the manner indicated by the HTTP 3xx codes. Clients are not required to support active redirection.

An SMP service **SHOULD** respond in accordance with **[HTTP 1.1]** to a request using the HTTP HEAD method.

3.2.2 Caching of HTTP responses

When using HTTP(S) 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, but **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(S) 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 2015 19:14:44 GMT
```

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

```
If-Modified-Since: Tue, 01 Dec 2015 19:14:44 GMT
```

3.3 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” which includes the «encoding» attribute set to “UTF-8”.

3.4 Resources and identifiers

3.4.1 REST interface

The REST interface comprises 2 types of resources:

Resource	URI	Method	XML resource root element	HTTP Status	Description of returned content
ServiceGroup	/ {identifier scheme} :: {id}	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.
SignedServiceMetadata	/ {identifier scheme} :: {id} / services / {docType} See section 2.4.6 for {docType} format	GET	<SignedServiceMetadata>	200; 500; 404	Holds all of the metadata about a Service, or a redirection URL to another Service Metadata Publisher holding this information.

Fig. 2: Table of resources and identifiers

3.4.2 Using identifiers in the REST Resource URLs

This section describes specifically how participant and document identifiers are used to reference <ServiceGroup> and <SignedServiceMetadata> REST resources. For a general definition on how to represent participant and document identifiers in URLs, see the Participant Identifier section of this document.

For the URL referencing a <ServiceGroup> resource, the “{identifier scheme} :: {id}” part follows the format described in the Participant Identifier section of this document.

In the reference to the SignedServiceMetadata or Redirect elements (/ {id} / services / {docType}), the {docType} part consists of {document identifier scheme} :: {document identifier}. For information on the format of {document identifier}, see the Document Identifier section of this document.

3.5 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 1.0 of the SMP REST binding:

<http://docs.oasis-open.org/bdxml/ns/SMP/2016/05>

This is identical to the target namespace of the SMP schema.

3.6 Security

3.6.1 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.

3.6.2 Message signature

3.6.2.1 Use of XML signatures

The message returned by the service is signed by the Service Metadata Publisher with XML-Signature [XML-DSIG1].

The signature MUST be an enveloped XML signature represented via an <ds:Signature> element embedded in the <SignedServiceMetadata> 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/xmlsig#enveloped-signature>
- The <ds:KeyInfo> element MUST contain an <ds:X509Data> element with an <ds:X509Certificate> sub-element containing the signer's X.509 certificate as base 64 encoded X509 DER value
- The canonicalization algorithm MUST be <http://www.w3.org/TR/2001/REC-xml-c14n-20010315>
- The SignatureMethod MUST be <http://www.w3.org/2001/04/xmlsig-more#rsa-sha256>
- The DigestMethod MUST be <http://www.w3.org/2001/04/xmlenc#sha256>

3.6.2.2 Verifying the signature

When verifying the signature, the consumer has access to the full certificate as a base 64 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).

3.6.2.3 Verifying the signature of the destination SMP

For the redirect scheme, the unique identifier of the destination SMP signing certificate is 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.

3.7 Schema for the REST interface

See appendix B for the XML Schema for all the resources of the REST interface.

4 Conformance

An implementation exhibits core conformance when Service implementations and Sender client lookup implementations are subject to conformance by complying with this specification including:

- 01 XML schemas distributed with this specification in the file bdx-smp-201605.xsd
- 02 Use of signatures for signing and verifications as defined in section 3.6.2
- 03 Process execution as defined in section 2.1
- 04 The syntax and semantics defined in the normative portions of chapter 3 of this specification
- 05 The Service Metadata Publisher REST binding as defined in sections 3.2, 3.3, 3.4 and 3.5 of this specification

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

Appendix A. 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
Oriol Bausa Peris	
Kenneth Bengtsson	Alfa1lab
Erlend Klakegg Bergheim	Difi-Agency for Public Management and eGovernment
Mikkel Brun	Tradeshift Network Ltd.
Andrea Caccia	AITI-Associazione Italiana Tesorieri de Impresa
Juan Cruellas	Departamento de Arquitectura de Computadores, Univ Politecnica de Cataluna
Kees Duvekot	RFS Holland Holding B.V.
Pim van der Eijk	Sonnenglanz Consulting
Sander Fieten	
Martin Forsberg	Swedish Association of Local Authorities & Regions
G. Ken Holman	Crane Softwrights Ltd.
Tim McGrath	Document Engineering Services Limited
Dale Moberg	Axway Software
Klaus Pedersen	Difi-Agency for Public Management and eGovernment
Sven Rasmussen	Danish Agency for Digitisation, Ministry of Finance
Susanne Wigard	Land Nordrhein-Westfalen

The BDXR TC wishes to thank everybody who participated in creating the original PEPPOL Transport Infrastructure Service Metadata Publisher specification which was submitted as input to the BDXR TC:

Organizations:

DIFI (Direktoratet for forvaltning og IKT), Norway
NITA (IT- og Telestyrelsen), Denmark
BRZ (Bundesrechenzentrum), Austria
Consp, Italy

Persons:

Bergthór Skúlason	NITA
Carl-Markus Piswanger	BRZ
Gert Sylvest	NITA/Avanade
Jens Jakob Andersen	NITA
Joakim Recht	NITA/Trifork
Kenneth Bengtsson	NITA/Alfa1lab
Klaus Vilstrup Pedersen	DIFI
Mike Edwards	NITA/IBM
Mikkel Hippe Brun	NITA
Paul Fremantle	NITA/WSO2
Philip Helger	BRZ
Thomas Gundel	NITA/IT Crew

Appendix B. SMP Schema (non-normative)

This appendix contains the SMP schema and should be identical to the schema file bdx-smp-201605.xsd distributed with this document. In case of discrepancies, the schema file bdx-smp-201605.xsd contains the normative information.

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns="http://docs.oasis-open.org/bdxr/ns/SMP/2016/05"
xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
elementFormDefault="qualified" targetNamespace="http://docs.oasis-
open.org/bdxr/ns/SMP/2016/05" id="ServiceMetadataPublishing">
  <xs:import namespace="http://www.w3.org/2000/09/xmldsig#"
schemaLocation="http://www.w3.org/TR/xmldsig-core/xmldsig-core-
schema.xsd"/>
  <xs:element name="ServiceGroup" type="ServiceGroupType"/>
  <xs:element name="ServiceMetadata" type="ServiceMetadataType"/>
  <xs:element name="SignedServiceMetadata"
type="SignedServiceMetadataType"/>
  <xs:complexType name="SignedServiceMetadataType">
    <xs:sequence>
      <xs:element ref="ServiceMetadata"/>
      <xs:element ref="ds:Signature"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="ServiceMetadataType">
    <xs:choice>
      <xs:element name="ServiceInformation"
type="ServiceInformationType"/>
      <xs:element name="Redirect" type="RedirectType"/>
    </xs:choice>
  </xs:complexType>
  <xs:complexType name="ServiceInformationType">
    <xs:sequence>
      <xs:element ref="ParticipantIdentifier"/>
      <xs:element ref="DocumentIdentifier"/>
      <xs:element name="ProcessList" type="ProcessListType"/>
      <xs:element name="Extension" type="ExtensionType" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="ProcessListType">
    <xs:sequence>
      <xs:element name="Process" type="ProcessType"
maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="ProcessType">
    <xs:sequence>
      <xs:element ref="ProcessIdentifier"/>
      <xs:element name="ServiceEndpointList"
type="ServiceEndpointList"/>
      <xs:element name="Extension" type="ExtensionType" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```

```

</xs:complexType>
<xs:complexType name="ServiceEndpointList">
  <xs:sequence>
    <xs:element name="Endpoint" type="EndpointType"
maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="EndpointType">
  <xs:sequence>
    <xs:element name="EndpointURI" type="xs:anyURI"/>
    <xs:element name="RequireBusinessLevelSignature"
type="xs:boolean" minOccurs="0" default="false"/>
    <xs:element name="MinimumAuthenticationLevel" type="xs:string"
minOccurs="0"/>
    <xs:element name="ServiceActivationDate" type="xs:dateTime"
minOccurs="0"/>
    <xs:element name="ServiceExpirationDate" type="xs:dateTime"
minOccurs="0"/>
    <xs:element name="Certificate" type="xs:base64Binary"/>
    <xs:element name="ServiceDescription" type="xs:string"/>
    <xs:element name="TechnicalContactUrl" type="xs:anyURI"/>
    <xs:element name="TechnicalInformationUrl" type="xs:anyURI"
minOccurs="0"/>
    <xs:element name="Extension" type="ExtensionType" minOccurs="0"
maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="transportProfile" type="xs:string"
use="required"/>
</xs:complexType>
<xs:complexType name="ServiceGroupType">
  <xs:sequence>
    <xs:element ref="ParticipantIdentifier"/>
    <xs:element name="ServiceMetadataReferenceCollection"
type="ServiceMetadataReferenceCollectionType"/>
    <xs:element name="Extension" type="ExtensionType" minOccurs="0"
maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="ServiceMetadataReferenceCollectionType">
  <xs:sequence>
    <xs:element name="ServiceMetadataReference"
type="ServiceMetadataReferenceType" minOccurs="0"
maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="ServiceMetadataReferenceType">
  <xs:attribute name="href" type="xs:anyURI"/>
</xs:complexType>
<xs:complexType name="RedirectType">
  <xs:sequence>
    <xs:element name="CertificateUID" type="xs:string"/>
    <xs:element name="Extension" type="ExtensionType" minOccurs="0"
maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="href" type="xs:anyURI" use="required"/>
</xs:complexType>
<xs:element name="ParticipantIdentifier"
type="ParticipantIdentifierType"/>

```

```

    <xs:element name="DocumentIdentifier" type="DocumentIdentifierType"/>
    <xs:element name="ProcessIdentifier" type="ProcessIdentifierType"/>
    <xs:element name="RecipientIdentifier"
type="ParticipantIdentifierType"/>
    <xs:element name="SenderIdentifier"
type="ParticipantIdentifierType"/>
    <xs:complexType name="ParticipantIdentifierType">
      <xs:simpleContent>
        <xs:extension base="xs:string">
          <xs:attribute name="scheme" type="xs:string"/>
        </xs:extension>
      </xs:simpleContent>
    </xs:complexType>
    <xs:complexType name="DocumentIdentifierType">
      <xs:simpleContent>
        <xs:extension base="xs:string">
          <xs:attribute name="scheme" type="xs:string"/>
        </xs:extension>
      </xs:simpleContent>
    </xs:complexType>
    <xs:complexType name="ProcessIdentifierType">
      <xs:simpleContent>
        <xs:extension base="xs:string">
          <xs:attribute name="scheme" type="xs:string"/>
        </xs:extension>
      </xs:simpleContent>
    </xs:complexType>
    <xs:complexType name="ExtensionType">
      <xs:annotation>
        <xs:documentation>
          A single extension for private use.
        </xs:documentation>
      </xs:annotation>
      <xs:sequence>
        <xs:element maxOccurs="1" minOccurs="0" name="ExtensionID"
type="xs:token">
          <xs:annotation>
            <xs:documentation>
              An identifier for the Extension assigned by the creator of the
              extension.
            </xs:documentation>
          </xs:annotation>
        </xs:element>
        <xs:element maxOccurs="1" minOccurs="0" name="ExtensionName"
type="xs:string">
          <xs:annotation>
            <xs:documentation>
              A name for the Extension assigned by the creator of the
              extension.
            </xs:documentation>
          </xs:annotation>
        </xs:element>
        <xs:element maxOccurs="1" minOccurs="0" name="ExtensionAgencyID"
type="xs:string">
          <xs:annotation>
            <xs:documentation>
              An agency that maintains one or more Extensions.
            </xs:documentation>
          </xs:annotation>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:sequence>
</xs:element>

```

```

        </xs:annotation>
    </xs:element>
    <xs:element maxOccurs="1" minOccurs="0"
name="ExtensionAgencyName" type="xs:string">
        <xs:annotation>
            <xs:documentation>
                The name of the agency that maintains the Extension.
            </xs:documentation>
        </xs:annotation>
    </xs:element>
    <xs:element maxOccurs="1" minOccurs="0" name="ExtensionAgencyURI"
type="xs:anyURI">
        <xs:annotation>
            <xs:documentation>
                A URI for the Agency that maintains the Extension.
            </xs:documentation>
        </xs:annotation>
    </xs:element>
    <xs:element maxOccurs="1" minOccurs="0" name="ExtensionVersionID"
type="xs:normalizedString">
        <xs:annotation>
            <xs:documentation>
                The version of the Extension.
            </xs:documentation>
        </xs:annotation>
    </xs:element>
    <xs:element maxOccurs="1" minOccurs="0" name="ExtensionURI"
type="xs:anyURI">
        <xs:annotation>
            <xs:documentation>
                A URI for the Extension.
            </xs:documentation>
        </xs:annotation>
    </xs:element>
    <xs:element maxOccurs="1" minOccurs="0"
name="ExtensionReasonCode" type="xs:token">
        <xs:annotation>
            <xs:documentation>
                A code for reason the Extension is being included.
            </xs:documentation>
        </xs:annotation>
    </xs:element>
    <xs:element maxOccurs="1" minOccurs="0" name="ExtensionReason"
type="xs:string">
        <xs:annotation>
            <xs:documentation>
                A description of the reason for the Extension.
            </xs:documentation>
        </xs:annotation>
    </xs:element>
    <xs:any namespace="##other" processContents="lax"/>
</xs:sequence>
</xs:complexType>
</xs:schema>

```

Appendix C. Non-Normative Examples

This appendix contains non-normative examples.

C.1 ServiceGroup resource

Non-normative example of the ServiceGroup resource:

```
01 <?xml version="1.0" encoding="utf-8"?>
02 <!--
03   This sample assumes that the service metadata publisher resides at
04   "http://example.org/".
05   It assumes that the that the identifier scheme is ISO 6523 ICD 0010
06   expressed using the ebCore Party ID URN format, and that the
07   business identifier is "5798000000001".
08 -->
09 <ServiceGroup xmlns="http://docs.oasis-open.org/bdxr/ns/SMP/2016/05">
10   <ParticipantIdentifier scheme="urn:oasis:names:tc:ebcore:partyid-
11     type:iso6523:0010">
12     5798000000001
13   </ParticipantIdentifier>
14   <ServiceMetadataReferenceCollection>
15     <ServiceMetadataReference
16       href="http://example.org/urn%3Aaasis%3Anames%3Atc:ebcore%3Apartyid-
17       type%3Aiso6523%3A0010%3A%3A5798000000001/services/bdx-docid-
18       qns%3A%3Aurn%3Aaasis%3Anames%3Aspecification%3Aubl%3Aschema%3Axsd%3AInvoic
19       e-2%3A%3AInvoice%23%23UBL-2.0"/>
20   </ServiceMetadataReferenceCollection>
21   <Extension>
22     <ExtensionID>Example</ExtensionID>
23     <ExtensionName>Example</ExtensionName>
24     <ExtensionAgencyID>BDXR</ExtensionAgencyID>
25     <ExtensionAgencyName>OASIS BDXR TC</ExtensionAgencyName>
26     <ExtensionAgencyURI>urn:example:org</ExtensionAgencyURI>
27     <ExtensionVersionID>0.1</ExtensionVersionID>
28     <ExtensionURI>urn:example:org</ExtensionURI>
29     <ExtensionReasonCode>Test</ExtensionReasonCode>
30     <ExtensionReason>Test</ExtensionReason>
31     <ex:Test xmlns:ex="http://example.org">Test</ex:Test>
32   </Extension>
33 </ServiceGroup>
```

Pseudo-schema for this data type:

```
<ServiceMetadata>
  [<ServiceInformation /> | <Redirect />]
</ServiceMetadata>
```

C.2 SignedServiceMetadata resource

Non-normative example of the SignedServiceMetadata resource:

```
01 <?xml version="1.0" encoding="utf-8"?>
02 <SignedServiceMetadata
03   xmlns="http://docs.oasis-open.org/bdxr/ns/SMP/2016/05"
04   xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
05   <ServiceMetadata>
06     <ServiceInformation>
```

```

07     <ParticipantIdentifier scheme="urn:oasis:names:tc:ebcore:partyid-
type:iso6523:0010">
08         5798000000001
09     </ParticipantIdentifier>
10     <DocumentIdentifier scheme="bdx-docid-qns">
11         urn:oasis:names:specification:ubl:schema:xsd:Invoice-2::Invoice
12     </DocumentIdentifier>
13     <ProcessList>
14         <Process>
15             <ProcessIdentifier scheme="cenbii-procid-ubl">
16                 urn:www.cenbii.eu:profile:bii04:ver2.0
17             </ProcessIdentifier>
18             <ServiceEndpointList>
19                 <Endpoint transportProfile="bdx-example-ebms-as4">
20                     <EndpointURI>http://example.org/sampleService/</EndpointURI>
21                     <RequireBusinessLevelSignature>
22                         false
23                     </RequireBusinessLevelSignature>
24                     <MinimumAuthenticationLevel>2</MinimumAuthenticationLevel>
25                     <ServiceActivationDate>
26                         2016-03-16T01:00:00.000+01:00
27                     </ServiceActivationDate>
28                     <ServiceExpirationDate>
29                         2026-03-15T01:00:00.000+01:00
30                     </ServiceExpirationDate>
31                     <Certificate>TlRMTVNTUAABAAAAA7IY4gk....</Certificate>
32                     <ServiceDescription>
33                         Sample invoice service
34                     </ServiceDescription>
35                     <TechnicalContactUrl>
36                         service@example.org
37                     </TechnicalContactUrl>
38                     <TechnicalInformationUrl>
39                         http://example.org/info
40                     </TechnicalInformationUrl>
41                     <Extension>
42                         <ex:Test xmlns:ex="http://example.org">Test</ex:Test>
43                     </Extension>
44                     <Extension>
45                         <ex:Test xmlns:ex="http://example.org/2">Test2</ex:Test>
46                     </Extension>
47                 </Endpoint>
48                 <Endpoint transportProfile="bdx-example-as2">
49                     <EndpointURI>http://example.org/sampleService/</EndpointURI>
50                     <RequireBusinessLevelSignature>
51                         false
52                     </RequireBusinessLevelSignature>
53                     <MinimumAuthenticationLevel>2</MinimumAuthenticationLevel>
54                     <ServiceActivationDate>
55                         2016-03-16T01:00:00.000+01:00
56                     </ServiceActivationDate>
57                     <ServiceExpirationDate>
58                         2026-03-15T01:00:00.000+01:00
59                     </ServiceExpirationDate>
60                     <Certificate>IjANBgkqhkiG9w0BAQEFAAO....</Certificate>
61                     <ServiceDescription>
62                         Sample invoice service 2
63                     </ServiceDescription>

```



```

64         <TechnicalContactUrl>
65             service@example.org
66         </TechnicalContactUrl>
67         <TechnicalInformationUrl>
68             http://example.org/info
69         </TechnicalInformationUrl>
70         <Extension>
71             <ex:Test xmlns:ex="http://example.org">Test</ex:Test>
72         </Extension>
73         <Extension>
74             <ex:Test xmlns:ex="http://example.org">Test</ex:Test>
75         </Extension>
76     </Endpoint>
77 </ServiceEndpointList>
78 </Process>
79 </ProcessList>
80 <Extension>
81     <ExtensionID>Example</ExtensionID>
82     <ExtensionName>Example</ExtensionName>
83     <ExtensionAgencyID>BDXR</ExtensionAgencyID>
84     <ExtensionAgencyName>OASIS BDXR TC</ExtensionAgencyName>
85     <ExtensionAgencyURI>urn:example:org</ExtensionAgencyURI>
86     <ExtensionVersionID>0.1</ExtensionVersionID>
87     <ExtensionURI>urn:example:org</ExtensionURI>
88     <ExtensionReasonCode>Test</ExtensionReasonCode>
89     <ExtensionReason>Test</ExtensionReason>
90     <ex:Test xmlns:ex="http://example.org">Test</ex:Test>
91 </Extension>
92 </ServiceInformation>
93 </ServiceMetadata>
94 <!-- Message signature, details omitted for brevity -->
95 <ds:Signature/>
96 </SignedServiceMetadata>

```

C.3 Redirect

Non-normative example of a Redirect response:

```

01 <?xml version="1.0" encoding="utf-8"?>
02 <!--
03     This sample assumes that the user contacts a service metadata publisher
04     that resides at "http://test.org/",
05     but is redirected to a service metadata publisher that resides at
06     "http://test2.org/".
07     -->
08 <SignedServiceMetadata
09     xmlns="http://docs.oasis-open.org/bdxr/ns/SMP/2016/05"
10     xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
11     <ServiceMetadata>
12         <Redirect
13             href="http://test2.org/urn%3Aaosis%3Anames%3Aatc:ebcore%3Apartyid-
14             type%3Aiso6523%3A0010%3A%3A5798000000001/services/bdx-docid-
15             qns%3A%3Aurn%3Aaosis%3Anames%3Aspecification%3Aubl%3Aschema%3Axsd%3AInvoice-
16             2%3A%3AInvoice%23%23UBL-2.0">
17             <CertificateUID>PID:9208-2001-3-279815395</CertificateUID>
18             <Extension>
19                 <ExtensionID>Example</ExtensionID>
20                 <ExtensionName>Example</ExtensionName>
21                 <ExtensionAgencyID>BDXR</ExtensionAgencyID>

```

```

18     <ExtensionAgencyName>OASIS BDXR TC</ExtensionAgencyName>
19     <ExtensionAgencyURI>urn:example:org</ExtensionAgencyURI>
20     <ExtensionVersionID>0.1</ExtensionVersionID>
21     <ExtensionURI>urn:example:org</ExtensionURI>
22     <ExtensionReasonCode>Test</ExtensionReasonCode>
23     <ExtensionReason>Test</ExtensionReason>
24     <ex:Test xmlns:ex="http://example.org">Test</ex:Test>
25   </Extension>
26 </Redirect>
27 </ServiceMetadata>
28 <!-- Message signature, details omitted for brevity -->
29 <ds:Signature/>
30 </SignedServiceMetadata>

```

C.4 Identifier

We assume a Service Metadata Publisher can be accessed at the URL “http://example.org”.

A business with the Participant Identifier «5798000000001» of ISO 6523 ICD 0010 (using the **[ebCorePartyId]** URN format) would have the following identifier for the ServiceGroup resource:

```

http://example.org/urn:oasis:names:tc:ebcore:partyid-
type:iso6523:0010::5798000000001

```

After percent encoding:

```

http://example.org/urn%3Aoasis%3Anames%3Atc:ebcore%3Apartyid-
type%3Aiso6523%3A0010%3A%3A5798000000001

```

In the case of a UBL 2.0 order, a SignedServiceMetadata or Redirect resource can then be identified by:

06 Identifier format type: bdx-docid-qns

07 Root namespace: urn:oasis:names:specification:ubl:schema:xsd:Order-2

08 Document element local name: Order

09 Subtype identifier: UBL-2.0 (since several versions of the Order schema may use the same namespace + document element name)

The document type identifier will then be:

```

bdx-docid-qns::urn:oasis:names:specification:ubl:schema:xsd:Order-
2::Order##UBL-2.0

```

The document type identifier **MUST** be percent encoded. The above, non-normative example is thus encoded to:

```

bdx-docid-
qns%3A%3Aurn%3Aoasis%3Anames%3Aspecification%3Aubl%3Aschema%3Axsd%3AOrd
er-2%3A%3AOrder%23%23UBL-2.0

```

The entire URL reference to a SignedServiceMetadata or Redirect element thus has the form {URL to server}/{identifier scheme}::{id}/services/{document identifier type}::{rootNamespace}::{documentElementLocalName}[##{Subtype identifier}]

The percent-encoded form of the identifier using the above example will then be:

```

http://example.org/urn%3Aoasis%3Anames%3Atc:ebcore%3Apartyid-
type%3Aiso6523%3A0010%3A%3A5798000000001/services/bdx-docid-
qns%3A%3Aurn%3Aoasis%3Anames%3Aspecification%3Aubl%3Aschema%3Axsd%3AOrd
er-2%3A%3AOrder%23%23UBL-2.0

```

Note that the forward slashes delimiting the individual parts of the REST resource identifier URL are not percent encoded, since they are part of the URL.