Service Metadata Publishing (SMP) Version 1.0

Committee Specification Draft 01

06 August 2014

Specification URIs

This version:

http://docs.oasis-open.org/bdxr/bdx-smp/v1.0/csd01/bdx-smp-v1.0-csd01.doc (Authoritative)
http://docs.oasis-open.org/bdxr/bdx-smp/v1.0/csd01/bdx-smp-v1.0-csd01.html
http://docs.oasis-open.org/bdxr/bdx-smp/v1.0/csd01/bdx-smp-v1.0-csd01.pdf

Previous version:

N/A

Latest version:

http://docs.oasis-open.org/bdxr/bdx-smp/v1.0/bdx-smp-v1.0.doc (Authoritative)
http://docs.oasis-open.org/bdxr/bdx-smp/v1.0/bdx-smp-v1.0.html
http://docs.oasis-open.org/bdxr/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
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 schemas: http://docs.oasis-open.org/bdxr/bdx-smp/v1.0/csd01/schemas/

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/bdxr/BDX-Location/v1.0/BDX-Location-v1.0.html.

Declared XML namespace:

- http://docs.oasis-open.org/bdxr/ns/SMP/2014/07

Abstract:

This document describes a protocol for publishing service metadata within a 4-corner network. In a 4-corner network, entities exchange 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) 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/bdxr/.

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 Technical Committee web page (https://www.oasis-open.org/committees/bdxr/ipr.php).

Citation format:

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

[BDX-smp-v1.0]

Service Metadata Publishing (SMP) Version 1.0. Edited by Jens Aabol, Kenneth Bengtsson, Sander Fieten, and Sven Rasmussen. 06 August 2014. OASIS Committee Specification Draft 01. http://docs.oasis-open.org/bdxr/bdx-smp/v1.0/csd01/bdx-smp-v1.0-csd01.html. Latest version: http://docs.oasis-open.org/bdxr/bdx-smp/v1.0/bdx-smp-v1.0.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 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

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.

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] 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.

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

[WSDL-2.0] "Web Services Description Language (WSDL) Version 2.0 Part 1: Core Language ",http://www.w3.org/TR/wsdl20/[WS-I BP] "WS-I Basic Profile Version 1.1", http://www.ws-i.org/Profiles/BasicProfile-1.1.html

[BDXL] Business Document Metadata Service Location (BDXL) http://docs.oasis-open.org/bdxr/BDX-Location/v1.0/cs01/BDX-Location-v1.0-cs01.html

2 SMP Protocol

2.1 The Service Discovery Process

In a 4-cornered architecture the discovery process is a two step process that start 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.

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:

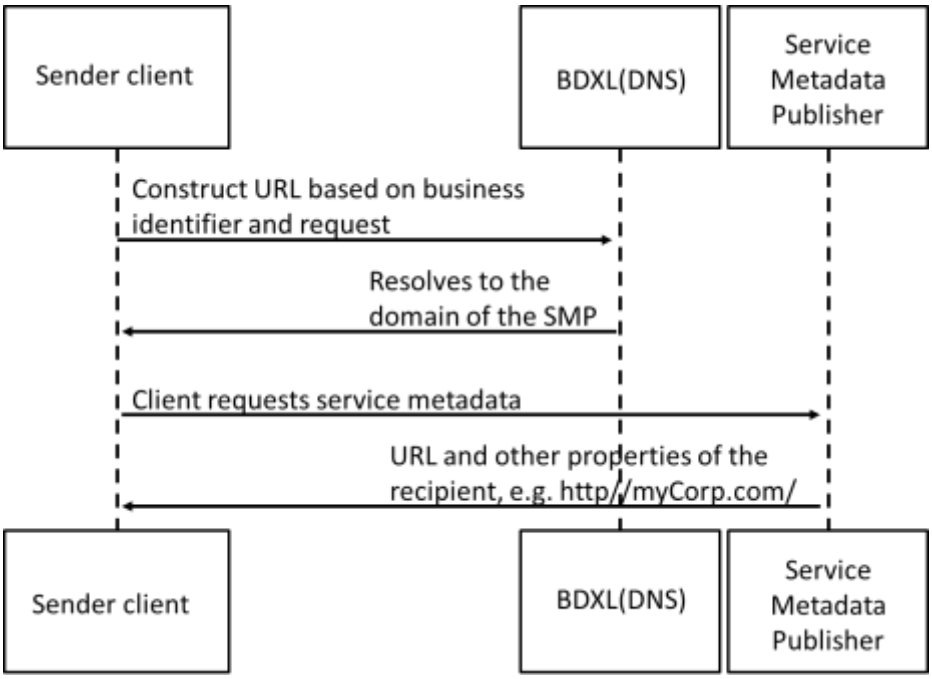

Fig.1. Endpoint lookup with Service Metadata

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.

2.1.1 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.

2.1.2 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:

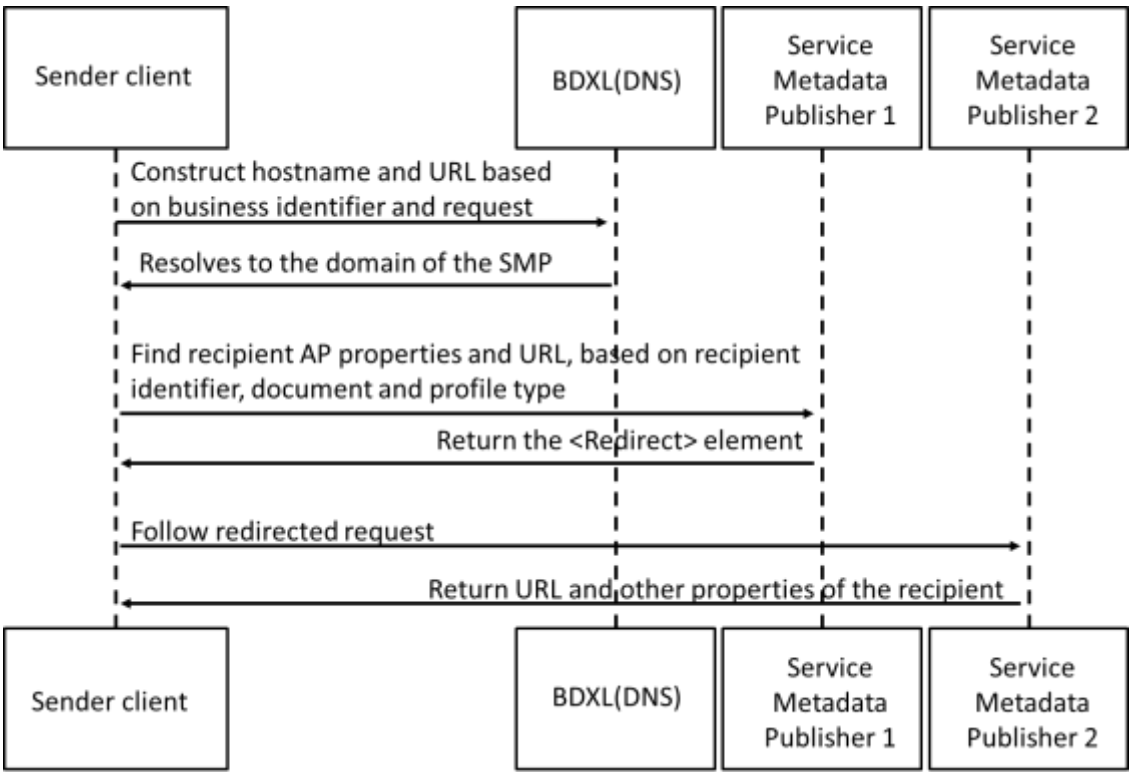

Fig. 2: Service Metadata Redirection

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

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.

2.3.1 On extension points

For each major entity, extension points have been added with the optional <smp:Extension> element. Semantics and use Child elements of the <smp:Extension> element are known as “custom extension elements”. Extension points may be used for optional extensions of service metadata. This implies:

- Extension elements added to a specific Service Metadata resource MUST be ignorable by any client of the transport infrastructure. The ability to parse and adjust client behavior based on an extension element MUST NOT be a prerequisite for a client to locate a service, or to make a successful request at the referenced service.
- A client MAY ignore any extension element added to specific service metadata resource instances.

2.3.2 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 ServiceList structure.

2.3.2.1 Pseudo-schema for ServiceGroup:

```
01     <smp:ServiceGroup>
02       <ids:ParticipantIdentifier [scheme=”xs:string”]>xs:string
03       </ids:ParticipantIdentifier>
04       <smp:ServiceMetadataReferenceCollection>
05         <smp:ServiceMetadataReference href=”xs:anyURI” />*
06       </smp:ServiceMetadataReferenceCollection>
07       <smp:Extension>xs:any</smp:Extension>?
08     </smp:ServiceGroup>
```

2.3.2.2 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 ParticpantIdentifier section 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.
ServiceMetadataReference (0..*)	Contains the URL to a specific SignedServiceMetadata instance - see the REST binding section 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.
Extension	The extension element may contain any XML element. Clients MAY ignore this element. It can be used to add extended metadata to individual references to Service Metadata resources.

2.3.2.3 Non-normative example of a ServiceGroup resource

See Appendix C.

2.3.3 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.

2.3.3.1 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:

```
01 <smp:ServiceMetadata>
02   [<smp:ServiceInformation /> | <smp:Redirect />]
03 </smp:ServiceMetadata>
```

2.3.3.2 Pseudo-schema for the “ServiceInformation” data type

```
01 <smp:ServiceInformation>
02   <ids:ParticipantIdentifier [scheme=”xs:string”]>xs:string
03   </ids:ParticipantIdentifier>
04   <ids:DocumentIdentifier [scheme=”xs:string”] />
05   <smp:ProcessList>
06     <smp:Process>+
07       <ids:ProcessIdentifier [scheme=”xs:string”] />
08       <smp:ServiceEndpointList>
09         <smp:Endpoint transportProfile=”xs:string”>+
10           <smp:EndpointURI>xs:anyURI</smp:EndpointURI>
11           <smp:RequireBusinessLevelSignature>xs:boolean
12           </smp:RequireBusinessLevelSignature>
13           <smp:MinimumAuthenticationLevel>xs:string
14           </smp:MinimumAuthenticationLevel >?
15           <smp:ServiceActivationDate>xs:dateTime
16           </smp:ServiceActivationDate>?
17           <smp:ServiceExpirationDate>xs:dateTime
18           </smp:ServiceExpirationDate>?
19           <smp:Certificate>xs:string</smp:Certificate>
20           <smp:ServiceDescription>xs:string
21           </smp:ServiceDescription>
22           <smp:TechnicalContactUrl>xs:anyURI
23           </smp:TechnicalContactUrl>
24           <smp:TechnicalInformationUrl>xs:anyURI
25           </smp:TechnicalInformationUrl>?
26           <smp:Extension>xs:any</smp:Extension>?
27         </smp:Endpoint>
28       </smp:ServiceEndpointList>
29       <smp:Extension>xs:any</smp:Extension>?
30     </smp:Process>
31   </smp:ProcessList>
32   <smp:Extension>xs:any</smp:Extension>?
33 </smp:ServiceInformation>
```

2.3.3.3 Pseudo-schema for the “Redirect” data type

```
01 <smp:Redirect href=”xs:anyURI”>
02   <smp:CertificateUID>xs:string</smp:CertificateUID>
03   <smp:Extension>xs:any</smp:Extension>?
04 <smp: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/busdox-actorid-
upis%3A%3A0010%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/busdox-actorid-
upis%3A%3A0010%3A5798000000001/services/bdx-docid-
qns%3A%3Aurn%3Aoasis%3Anames%3Aspecification%3Aubl%3Aschema%3Axsd%3AInv
oice- 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.

2.3.3.4 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. 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 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, which the format of the identifier itself. See the DocumentIdentifier section 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 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 just defines one process identifier, which represents documents that are not sent under any specific process: `bdx:noprocess`

	The process identifier MUST be treated as case insensitive. See the Process section of this specification for information on the identifier format.
/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	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.
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. Format of ServiceActivationDate date is `xs:dateTime`.
/ProcessList/../Endpoint/ ServiceExpirationDate	Expiration date of the service. Senders should ignore services that are expired. Format of ServiceExpirationDate date is `xs:dateTime`.
/ProcessList/../Endpoint/ Certificate	Holds the complete signing certificate of the recipient gateway, as a PEM base 64 encoded X509 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.
/Process/Extension	The extension element may contain any XML element. Clients MAY ignore this element. It can be used to add extension metadata to the process metadata block as a whole.

/ServiceInformation/ Extension	The extension element may contain any XML element. Clients MAY ignore this element. It can be used to add extension metadata to the service metadata.

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

2.3.4 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:

```
01 <smp:SignedServiceMetadata>
02   <smp:ServiceMetadata />
03   <ds:Signature />
04 </smp: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

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

2.4.1 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[/endpointName]
```

Can be instantiated to either of the strings

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

And

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

2.4.2 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.

2.4.3 On Scheme Identifiers

Identifier schemes for all schemed identifier types (participants, documents, profiles, transports) may be defined outside of this specification. Any instance of a 4-cornered infrastructure may choose to define identifier schemes that match the type of documents, participants or profiles that are relevant to support in that instance.

An example is the Scheme Identifier being used in the European PEPPOL network, «busdox-actorid-upis».

2.4.4 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.

2.4.4.1 XML format for Participant Identifiers

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

Pseudo-scheme for ParticipantIdentifier:

```
01 <smp:ParticipantIdentifier [scheme=”xs:string”]>xs:string
02 </smp: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	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). This scheme type MUST be in the form of a URI. When processing a participant identifier in XML format, it MUST be treated as case insensitive.

2.4.4.2 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 that uses the PEPPOL Universal Participant Identifier Format, assuming the participant identifier «0010:5798000000001»:

```
busdox-actorid-upis::0010:5798000000001
```

In percent encoded form:

```
busdox-actorid-upis%3a%3a0010%3a5798000000001
```

When processing a participant identifier in an URL, it 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.

2.4.5 DocumentIdentifier

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 order, this document can then be identified by

- **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:

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

2.4.5.1 XML Representation of Document Identifiers

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

Pseudo-scheme for DocumentIdentifier:

```
01 <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 order document, this would be «urn:oasis:names:specification:ubl:schema:xsd:Order-2::Order##UBL-2.0».
ParticipantIdentifier/ @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: *bdx-docid-qns* When processing a document identifier in XML format, it MUST be treated as case insensitive.

2.4.5.2 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 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

Note that although namespaces and element names are case sensitive, the document identifier MUST be treated as case insensitive.

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.

2.4.6 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.

2.4.6.1 XML Representation of Process Identifiers

Pseudo-schema for the ProcessIdentifier XML element:

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

Description of the individual fields (elements and attributes):

Field	Description

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 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 just defines one process identifier, which represents documents that are not sent under any specific process: `bdx:noprocess` The process identifier MUST be treated as case insensitive.
ProcessIdentifier/@scheme	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. This document just defines one process scheme identifier, which represents transport-specific process identifiers: *bdx-procid-transport* 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.

3 Service Metadata Publishing REST binding

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.

3.1.1 The use of HTTP

A service implementing the REST binding MUST set the HTTP “content-type” header, and give it a value of “text/xml”. A service implementing the REST profile MUST NOT use TLS (Transport Layer Security) or SSL (Secure Sockets Layer). 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.

3.1.2 The use of XML and encoding

XML document returned by HTTP GET MUST be UTF-8 encoded. They MUST contain a document type declaration starting with “<?xml” which includes the «encoding’ attribute set to “UTF-8”. Please observe that the content of the encoding attribute is case sensitive. Version 1.0 of XML is used.

3.1.3 Resources and identifiers

The REST interface comprises 2 types of resources:

Resource	URI	Me-thod	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 below 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

3.1.3.1 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.

3.1.4 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/bdxr/ns/SMP/2014/07

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

3.2 Security

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.

3.2.1 Message signature

The message returned by the service is signed by the Service Metadata Publisher with XML-Signature according to the standard http://www.w3.org/TR/2002/REC-xmldsig-core-20020212/.

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/xmldsig#envelopedsignature
- The <ds:KeyInfo> element MUST contain an <ds:X509Data> element with an <ds:X509Certificate> sub-element containing the signer's X.509 certificate as PEM 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/2000/09/xmldsig#rsa-sha1

- The DigestMethod MUST be http://www.w3.org/2000/09/xmldsig#sha1

3.2.1.1 Verifying the signature

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

3.2.1.2 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.

3.3 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:

- xsd schemas
- Use of signatures for signing
- Process execution
- The syntax and semantics defined in the normative portions of this specification

This specification allows extensions. Each implementation SHALL fully support all required functionality of the specification exactly as specified. The use of extensions SHALL NOT contradict nor cause the non-conformance of functionality defined in the specification.

Appendix A. Acknowledgments

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
Mikkel Brun	Tradeshift Network Ltd.
Andrea Caccia	AITI-Associazione Italiana Tesorieri de Impresa
Juan Cruellas	Departamento de Arquitectura de Computadores, Univ Politecnica de Cataluna
Pim van der Eijk	Sonnenglanz Consulting
Sander Fieten	
Martin Forsberg	Swedish Association of Local Authorities & Regions
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
Consip, 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

bdx-smp-201407.xsd:

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified"
  targetNamespace="http://docs.oasis-open.org/bdxr/ns/SMP/2014/07"
  xmlns="http://docs.oasis-open.org/bdxr/ns/SMP/2014/07"
id="ServiceMetadataPublishing"
  xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
xmlns:wsa="http://www.w3.org/2005/08/addressing">
  <xs:import namespace="http://www.w3.org/2000/09/xmldsig#"
    schemaLocation="http://www.w3.org/TR/xmldsig-core/xmldsig-core-
schema.xsd"/>
  <xs:import
    namespace="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-utility-1.0.xsd"/>
  <xs:element name="ServiceGroup" type="ServiceGroupType"/>
  <xs:element name="ServiceMetadata" type="ServiceMetadataType"/>
  <xs:element name="SignedServiceMetadata"
type="SignedServiceMetadataType"/>
  <xs:complexType name="SignedServiceMetadataType">
    <xs:sequence>
      <xs:element ref="ServiceMetadata"/>
      <xs:element ref="ds:Signature"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="ServiceMetadataType">
    <xs:choice>
      <xs:element name="ServiceInformation"
type="ServiceInformationType"/>
      <xs:element name="Redirect" type="RedirectType"/>
    </xs:choice>
  </xs:complexType>
  <xs:complexType name="ServiceInformationType">
    <xs:sequence>
      <xs:element ref="ParticipantIdentifier"/>
      <xs:element ref="DocumentIdentifier"/>
      <xs:element name="ProcessList" type="ProcessListType"/>
      <xs:element name="Extension" type="ExtensionType" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="ProcessListType">
    <xs:sequence>
      <xs:element name="Process" type="ProcessType"
maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="ProcessType">
    <xs:sequence>
      <xs:element ref="ProcessIdentifier"/>
      <xs:element name="ServiceEndpointList" type="ServiceEndpointList"/>
      <xs:element name="Extension" type="ExtensionType" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
```

```
45    <xs:complexType name="ServiceEndpointList">
46      <xs:sequence>
47        <xs:element name="Endpoint" type="EndpointType"
    maxOccurs="unbounded"/>
48      </xs:sequence>
49    </xs:complexType>
50    <xs:complexType name="EndpointType">
51      <xs:sequence>
52        <xs:element name="EndpointURI" type="xs:anyURI"/>
53        <xs:element name="RequireBusinessLevelSignature"
    type="xs:boolean"/>
54        <xs:element name="MinimumAuthenticationLevel" type="xs:string"
    minOccurs="0"/>
55        <xs:element name="ServiceActivationDate" type="xs:dateTime"
    minOccurs="0"/>
56        <xs:element name="ServiceExpirationDate" type="xs:dateTime"
    minOccurs="0"/>
57        <xs:element name="Certificate" type="xs:base64Binary"/>
58        <xs:element name="ServiceDescription" type="xs:string"/>
59        <xs:element name="TechnicalContactUrl" type="xs:anyURI"/>
60        <xs:element name="TechnicalInformationUrl" type="xs:anyURI"
    minOccurs="0"/>
61        <xs:element name="Extension" type="ExtensionType" minOccurs="0"/>
62      </xs:sequence>
63      <xs:attribute name="transportProfile" type="xs:string"
    use="required"/>
64    </xs:complexType>
65    <xs:complexType name="ServiceGroupType">
66      <xs:sequence>
67        <xs:element ref="ParticipantIdentifier"/>
68        <xs:element name="ServiceMetadataReferenceCollection"
69          type="ServiceMetadataReferenceCollectionType"/>
70        <xs:element name="Extension" type="ExtensionType" minOccurs="0"/>
71      </xs:sequence>
72    </xs:complexType>
73    <xs:complexType name="ServiceMetadataReferenceCollectionType">
74      <xs:sequence>
75        <xs:element name="ServiceMetadataReference"
    type="ServiceMetadataReferenceType"
76          minOccurs="0" maxOccurs="unbounded"/>
77      </xs:sequence>
78    </xs:complexType>
79    <xs:complexType name="ServiceMetadataReferenceType">
80      <xs:attribute name="href" type="xs:anyURI"/>
81    </xs:complexType>
82    <xs:complexType name="RedirectType">
83      <xs:sequence>
84        <xs:element name="CertificateUID" type="xs:string"/>
85        <xs:element name="Extension" type="ExtensionType" minOccurs="0"/>
86      </xs:sequence>
87      <xs:attribute name="href" type="xs:anyURI" use="required"/>
88    </xs:complexType>
89    <xs:complexType name="ExtensionType">
90      <xs:sequence>
91        <xs:any/>
92      </xs:sequence>
93    </xs:complexType>
```



```
94      <xs:element name="ParticipantIdentifier"
     type="ParticipantIdentifierType"/>
95      <xs:element name="DocumentIdentifier" type="DocumentIdentifierType"/>
96      <xs:element name="ProcessIdentifier" type="ProcessIdentifierType"/>
97      <xs:element name="RecipientIdentifier"
     type="ParticipantIdentifierType"/>
98      <xs:element name="SenderIdentifier" type="ParticipantIdentifierType"/>
99      <xs:complexType name="ParticipantIdentifierType">
100       <xs:simpleContent>
101         <xs:extension base="xs:string">
102           <xs:attribute name="scheme" type="xs:string"/>
103         </xs:extension>
104       </xs:simpleContent>
105     </xs:complexType>
106     <xs:complexType name="DocumentIdentifierType">
107       <xs:simpleContent>
108         <xs:extension base="xs:string">
109           <xs:attribute name="scheme" type="xs:string"/>
110         </xs:extension>
111       </xs:simpleContent>
112     </xs:complexType>
113     <xs:complexType name="ProcessIdentifierType">
114       <xs:simpleContent>
115         <xs:extension base="xs:string">
116           <xs:attribute name="scheme" type="xs:string"/>
117         </xs:extension>
118       </xs:simpleContent>
119     </xs:complexType>
120   </xs:schema>
```

Appendix C. Non-Normative Examples

This appendix contains non-normative examples.

C.1 ServiceGroup resource

Non-normative example of the ServiceGroup resource:

```
<?xml version="1.0" encoding="utf-8" ?>
<!--
This sample assumes that the service metadata publisher resides at
"http://serviceMetadata.eu/".
It assumes that the business identifier is "0010:5798000000001".
-->
<ServiceGroup xmlns="http://busdox.org/serviceMetadata/publishing/1.0/"
  xmlns:ids="http://busdox.org/transport/identifiers/1.0/">
  <ids:ParticipantIdentifier scheme="busdox-actorid-upis">
    0010:5798000000001
  </ids:ParticipantIdentifier>
  <ServiceMetadataReferenceCollection>
    <ServiceMetadataReference href="http://serviceMetadata.eu/busdox-
actorid-upis%3A%3A0010%3A5798000000001/services/bdx-docid-
qns%3A%3Aurn%3Aoasis%3Anames%3Aspecification%3Aubl%3Aschema%3Axsd%3AInvoic
e- 2%3A%3AInvoice%23%23UBL-2.0" />
  </ServiceMetadataReferenceCollection>
  <Extension>
    <ex:Test xmlns:ex="http://test.eu">Test</ex:Test>
  </Extension>
</ServiceGroup>
```

Pseudo-schema for this data type:

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

C.2 SignedServiceMetadata resource

Non-normative example of the SignedServiceMetadata resource:

```
<?xml version="1.0" encoding="utf-8" ?>
<!--
This sample assumes that the service metadata publisher resides at
"http://serviceMetadata.eu/".
It assumes that the business identifier is "0010:5798000000001".
-->
<SignedServiceMetadata
  xmlns="http://busdox.org/serviceMetadata/publishing/1.0/"
  xmlns:ids="http://busdox.org/transport/identifiers/1.0/">
  <ServiceMetadata
    xmlns="http://busdox.org/serviceMetadata/publishing/1.0/"
    xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-utility-1.0.xsd">
    <ServiceInformation>
      <ids:ParticipantIdentifier scheme="busdox-actorid-upis">
        0010:5798000000001
      </ids:ParticipantIdentifier>
      <ids:DocumentIdentifier scheme="bdx-docid-qns">
```

```
17          urn:oasis:names:specification:ubl:schema:xsd:Invoice-
   2::Invoice##UBL-2.02
18        </ids:DocumentIdentifier>
19        <ProcessList>
20          <Process>
21            <ids:ProcessIdentifier scheme="cenbii-procid-ubl">BII04
22            </ids:ProcessIdentifier>
23            <ServiceEndpointList>
24              <Endpoint transportProfile="busdox-transport-start">
25                <EndpointURI>http://busdox.org/sampleService/</EndpointURI>
26                <RequireBusinessLevelSignature>false
27                </RequireBusinessLevelSignature>
28                <MinimumAuthenticationLevel>2</MinimumAuthenticationLevel>
29                <ServiceActivationDate>2009-05-01T09:00:00
30                </ServiceActivationDate>
31                <ServiceExpirationDate>2016-05-01T09:00:00
32                </ServiceExpirationDate>
33                <Certificate>TlRMTVNTUAABAAAAt7IY4gk....</Certificate>
34                <ServiceDescription>invoice service</ServiceDescription>
35                <TechnicalContactUrl>https://example.com
36                </TechnicalContactUrl>
37                <TechnicalInformationUrl>http://example.com/info
38                </TechnicalInformationUrl>
39              </Endpoint>
40            </ServiceEndpointList>
41          </Process>
42          <Process>
43            <ids:ProcessIdentifier scheme="cenbii-procid-ubl">BII07
44            </ids:ProcessIdentifier>
45            <ServiceEndpointList>
46              <Endpoint transportProfile="busdox-transport-start">
47                <EndpointURI>http://busdox.org/sampleService/</EndpointURI>
48                <RequireBusinessLevelSignature>true
49                </RequireBusinessLevelSignature>
50                <MinimumAuthenticationLevel>1</MinimumAuthenticationLevel>
51                <ServiceActivationDate>2009-05-01T09:00:00
52                </ServiceActivationDate>
53                <ServiceExpirationDate>2016-05-01T09:00:00
54                </ServiceExpirationDate>
55                <Certificate>TlRMTVNTUAABAAAAt7IY4gk....</Certificate>
56                <ServiceDescription>invoice service</ServiceDescription>
57                <TechnicalContactUrl>https://example.com
58                </TechnicalContactUrl>
59                <TechnicalInformationUrl>http://example.com/info
60                </TechnicalInformationUrl>
61                <Extension>
62                  <ex:Test xmlns:ex="http://test.eu">Test</ex:Test>
63                </Extension>
64              </Endpoint>
65            </ServiceEndpointList>
66            <Extension>
67              <ex:Test xmlns:ex="http://test.eu">Test</ex:Test>
68            </Extension>
69          </Process>
70        </ProcessList>
71        <Extension>
72          <ex:Test xmlns:ex="http://test.eu">Test</ex:Test>
73        </Extension>
```



```
74      </ServiceInformation>
75    </ServiceMetadata>
76    <!-- Message signature, details omitted for brevity -->
77    <Signature xmlns="http://www.w3.org/2000/09/xmldsig#"/>
78  </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://serviceMetadata.eu/",
05  but is redirected to a service metadata publisher that resides at
06  "http://serviceMetadata2.eu/".
07  -->
08  <SignedServiceMetadata
09    xmlns="http://busdox.org/serviceMetadata/publishing/1.0/">
10    <ServiceMetadata
11      xmlns="http://busdox.org/serviceMetadata/publishing/1.0/">
12      <Redirect xmlns="http://busdox.org/serviceMetadata/publishing/1.0/"
13        href="http://serviceMetadata2.eu/busdox-actorid-
    upis%3A%3A0010%3A5798000000001/services/bdx-docid-
    qns%3A%3Aurn%3Aoasis%3Anames%3Aspecification%3Aubl%3Aschema%3Axsd%3AInvoic
    e- 2%3A%3AInvoice%23%23UBL-2.0">
14        <CertificateUID>PID:9208-2001-3-279815395</CertificateUID>
15        <Extension>
16          <ex:Test xmlns:ex="http://test.eu">Test</ex:Test>
17        </Extension>
18      </Redirect>
19    </ServiceMetadata>
20    <!-- Message signature, details omitted for brevity -->
21    <Signature xmlns="http://www.w3.org/2000/09/xmldsig#"/>
22  </SignedServiceMetadata>
```

C.4 Identifier

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

A business with the Participant Identifier “0010:5798000000001” would have the following identifier for the ServiceGroup resource:

```
http://serviceMetadata.eu/busdox-actorid-upis::0010:5798000000001
```

After percent encoding:

```
http://serviceMetadata.eu/busdox-actorid-upis%3a%3a0010%3a5798000000001
```

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

- **Identifier format type:** 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
```

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://serviceMetadata.eu/busdox-actorid-
upis%3a%3a0010%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.

Appendix D. Revision History

Revision	Date	Editor	Changes Made
wd-01	2014-02-27	Jens Aabol	Initial draft
wd-02	2014-05-14	Jens Aabol and Sander Fieten	Non-normative examples moved to Appendix B. Definitions moved to Appendix C. Conformance Definitions placed under Conformance. Namespace changed to Oasis specification. Changed EndpointReference to EndpointURI etc. Updated Appendix A. Merged definitions of common identifiers into document and rearranged sections. Generalized explications of REST and transport protocol bindings.
wd-03	2014-07-23	Sander Fieten and Kenneth Bengtsson	Moved XML schema to XSD file and updated declared namespace accordingly. Consolidated XML schemas. Moved the pseudo schema for the REST interface from chapter 3 to Appendix B. Moved non-normative examples to Appendix C. Moved revision history to Appendix D. Updated conformance clause (chapter 4). Updated Acknowledgements in Appendix A with members of the TC and inserted credits for organizations and individuals authoring the original PEPPOL SMP specification. Reformatted document using OASIS starter document.