



# Service Metadata Publishing

---

*Version 0.9.5*



WP8 2009-10-22



## Contents

|       |  |    |
|-------|--|----|
| 1     | Document information .....                                       | 4  |
| 1.1   | Document history .....   | 4  |
| 1.2   | Editor .....   | 4  |
| 2     | Contributors (alphabetically).....                               | 4  |
| 3     | Introduction.....  | 6  |
| 3.1   | Goals and non-goals .....  | 6  |
| 3.2   | Terminology.....   | 6  |
| 3.2.1 | Notational conventions .....                                     | 6  |
| 3.2.2 | Normative references.....  | 6  |
| 3.2.3 | Non-normative references .....                                   | 6  |
| 3.3   | Namespaces.....  | 6  |
| 4     | The Service Discovery Process.....                               | 7  |
| 4.1   | Discovery flow .....   | 7  |
| 4.2   | Discovering services associated with a Business Identifier ..... | 8  |
| 4.3   | Service Metadata Publisher Redirection .....                     | 8  |
| 5     | Interface model .....  | 9  |
| 6     | Data model .....   | 9  |
| 6.1   | On extension points.....   | 9  |
| 6.1.1 | Semantics and use .....  | 9  |
| 6.2   | ServiceGroup .....   | 10 |
| 6.2.1 | Non-normative example.....                                       | 11 |
| 6.3   | ServiceMetadata.....   | 12 |
| 6.3.1 | Redirection .....  | 12 |
| 6.3.2 | Non-normative example.....                                       | 15 |
| 6.4   | SignedServiceMetadata.....                                       | 15 |
| 6.4.1 | Non-normative example.....                                       | 16 |
| 6.4.2 | Redirect, non-normative example.....                             | 17 |
| 6.5   | ProcessList .....  | 18 |
| 7     | Service Metadata Publishing REST binding .....                   | 19 |
| 7.1   | The use of HTTP .....  | 19 |

|       |   |    |
|-------|---|----|
| 7.1.1 | HTTP status codes.....                                      | 19 |
| 7.2   | The use of XML and encoding .....                           | 19 |
| 7.3   | Resources and identifiers .....                             | 20 |
| 7.3.1 | On the use of percent encoding .....                        | 20 |
| 7.3.2 | Using identifiers in the REST Resource URLs .....           | 20 |
| 7.3.3 | Non-normative identifier example .....                      | 21 |
| 7.3.4 | Additional constraints on the Resource Identifier URL ..... | 22 |
| 7.3.5 | Implementation considerations .....                         | 22 |
| 7.4   | Referencing the SMP REST binding .....                      | 22 |
| 7.5   | Security .....  | 22 |
| 7.5.1 | Message signature.....                                      | 22 |
| 7.5.2 | Verifying the signature .....                               | 23 |
| 8     | Appendix A: Schema for the REST interface.....              | 24 |

# 1 Document information

## 1.1 Document history

| Date       | Version | Initials | Changes  |
|------------|---------|----------|--|
| 2009-02-12 | 0.1.0   | GS       | Created document   |
| 2009-02-16 | 0.1.2   | GS       | Added WSDLs & included logical data model  |
| 2009-02-17 | 0.5.0   | GS       | Minor correction   |
| 2009-03-30 | 0.6.0   | GS       | Changed into a REST profile, UDDI and REST profiles collected into one doc, removed hashed   |
| 2009-04-01 | 0.7.0   | GS       | Merge completed, UDDI profile updated with current keys and identifiers, implied data model discussed, encryption certificate placed, non-normative examples up-to-date  |
| 2009-04-29 | 0.8.0   | GS       | Started changes for 0.8 version  |
| 2009-05-04 | 0.8.0   | GS       | Final content for 0.8 version  |
| 2009-06-25 | 0.8.2   | GS       | Updated with structural changes & extension points for 0.9 version   |
| 2009-07-01 | 0.8.3   | GS       | Updates to examples, schema, references. Moved identifier information to 'common definitions' document.  |
| 2009-08-28 | 0.8.4   | GS       | Updated some references & diagrams, changed the redirect scheme, changed 'Certificate' element to 'CertificateCN'  |
| 2009-08-31 | 0.9.0   | GS       | Updated naming, redirect scheme, diagrams, schemas, examples, proof-reading  |
| 2009-10-22 | 0.9.5   | GS       | Added an identifier for referencing the REST binding of the SMP (SMP TNS), updated all diagrams & descriptions to reflect the DNS-based SML, removed requirement for port 80, removed wsu:Id element, editorial updates. |

## 1.2 Editor

Gert Sylvest, Avancode

## 2 Contributors (alphabetically)

Jens Jakob Andersen, NITA

Kenneth Bengtsson, Alfa1lab

Mikkel Hippe Brun, NITA

Mike Edwards, IBM

Paul Fremantle, WSO2

Thomas Gundel, IT Crew

Philip Helger, Bundesrechenzentrum

Joakim Recht, Trifork

Carl-Markus Piswanger, Bundesrechenzentrum

Bergþór Skúlason, NITA

Gert Sylvest, Avanade

## 1 **3 Introduction**

2 This document describes the REST (Representational State Transfer) interface for Service Metadata  
3 Publication within the Business Document Exchange Network (BUSDOX). It describes the request/response  
4 exchanges between a Service Metadata Publisher and a client wishing to discover endpoint information. It  
5 also defines the request processing that must happen at the client.

### 6 **3.1 Goals and non-goals**

7 The goal of this document is to define the REST lookup interface that Service Metadata Publishers ("SMP")  
8 and clients must support. Decisions regarding physical data format and management interfaces are left to  
9 implementers of such a service.

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

### 13 **3.2 Terminology**

14 For a definition of terms, see [BDEN-CDEF].

#### 15 **3.2.1 Notational conventions**

16 For notational conventions, see [BDEN-CDEF].

#### 17 **3.2.2 Normative references**

18 [XML-DSIG] "XML Signature Syntax and Processing (Second Edition)", <http://www.w3.org/TR/xmlldsig-core/>

19 [RFC3986] "Uniform Resource Identifier (URI): Generic Syntax", <http://tools.ietf.org/html/rfc3986>

20 [WSA-1.0] "Web Services Addressing 1.0 - Core" (<http://www.w3.org/TR/2005/CR-ws-addr-core-20050817/>) and "Web Services Addressing 1.0 - SOAP Binding", <http://www.w3.org/TR/ws-addr-soap/>

22 [RFC-2119] "Key words for use in RFCs to Indicate Requirement Levels",  
23 <http://www.ietf.org/rfc/rfc2119.txt>

24 [BDEN-CDEF] Business Document Exchange Network - Common Definitions, CommonDefinitions.pdf

#### 25 **3.2.3 Non-normative references**

26 [WSDL-2.0] "Web Services Description Language (WSDL) Version 2.0 Part 1: Core Language",  
27 <http://www.w3.org/TR/wsdl20/>

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

30 [BDEN-SML] Service Metadata Locator Profile, ServiceMetadataLocator\_DNS.pdf

## 31 **3.3 Namespaces**

32 For a list of namespaces and prefixes used in this document, see [BDEN-CDEF].

33

34

35 **4 The Service Discovery Process**

36 The interfaces of the Service Metadata Locator service and the Service Metadata Publisher (SMP) service  
37 cover both sender-side lookup and metadata management performed by SMPs. Business Document  
38 Exchange Network (BUSDOX) mandates the following interfaces for these services:

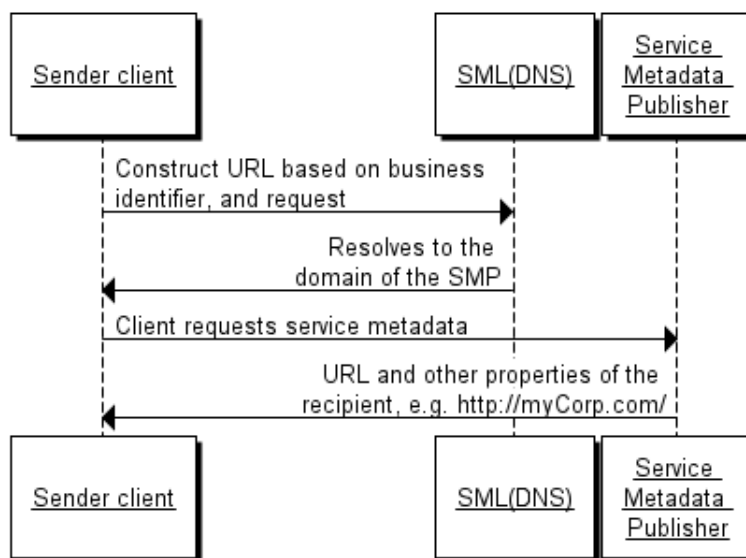
- 39
- 40 • Service Metadata Locator:
    - 41 ○ DNS-based resolve mechanism to locate individual SMPs
    - 42 ○ Management interface towards SMPs
  - 43 • Service Metadata Publishers:
    - 44 ○ Discovery interface towards senders
- 45

45 This specification only covers the discovery interface for Service Metadata Publication services.

46 **4.1 Discovery flow**

47 For a sender, the first step in the Discovery process is to establish the location of the Service Metadata  
48 relating to the particular Business Identifier to which the sender wants to transmit a message. Each  
49 business identifier is registered with one and only one Service Metadata Publisher. The sender looks up the  
50 endpoint for the Service Metadata Publisher using the DNS-based Service Metadata Locator service (this is  
51 a regular DNS resolve). The sender can then retrieve Service Metadata using Service Metadata Publisher  
52 services to obtain the metadata about the Business Identifier, which includes the information necessary to  
53 transmit the message to the recipient endpoint represented by that Business Identifier.

54 The diagram below represents the lookup flow for a sender contacting both the Service Metadata Locator  
55 and the SMP.



56  
57 **Fig. 1. Endpoint lookup with Service Metadata**

58

59 **4.2 Discovering services associated with a Business Identifier**

60 In addition to the direct lookup of Service Metadata based on business identifier and document type, a  
61 sender may want to discover what document types can be handled by a specific business identifier.

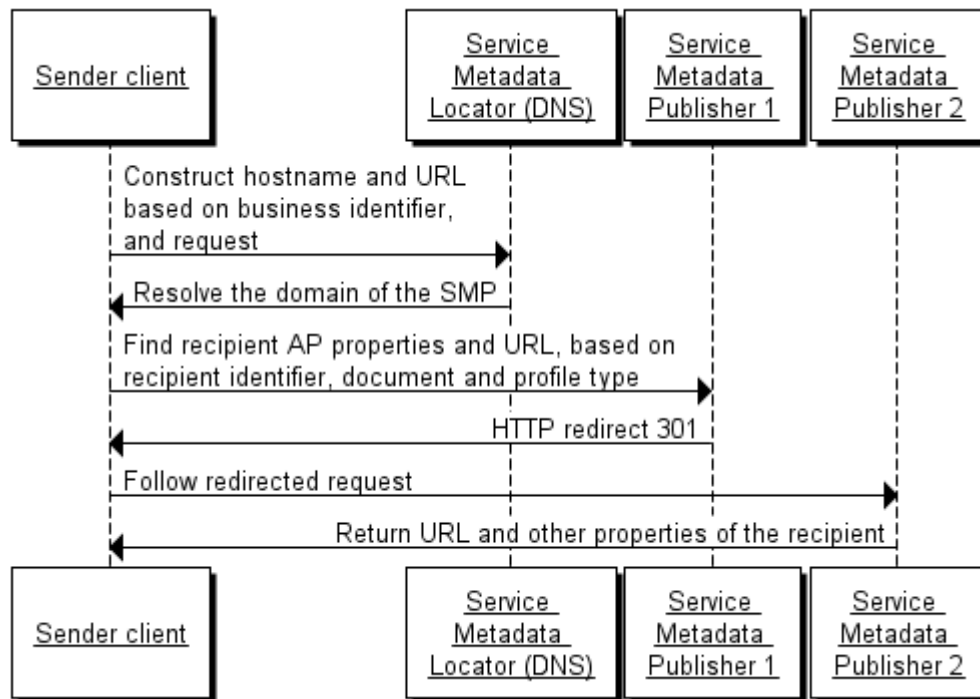
62 This is enabled by a pattern where the sender first retrieves the ServiceGroup entity, which holds a list of  
63 references to the ServiceMetadata resources associated with it.

64 **4.3 Service Metadata Publisher Redirection**

65 For each business identifier, the Service Metadata Locator may only point to a single Service Metadata  
66 Publisher. There are cases however where the owner of a business identifier may want to use different  
67 Service Metadata Publishers for different document types or processes. This is supported by Service  
68 Metadata Publisher Redirection.

69 In this pattern, the sender is redirected by the Service Metadata Publisher to a secondary, remote Service  
70 Metadata Publisher where the actual SignedServiceMetadata can be found. A special element within the  
71 SignedServiceMetadata record of the SMP points to the SMP that has the actual Service Metadata, and  
72 certificate information for that SMP. The diagram below shows this flow:

73



74

75

Fig. 2: Service Metadata Redirection

76

## 77 **5 Interface model**

78 This specification defines a REST-based interface for retrieving Service Metadata, but does not specify  
79 interfaces for creating, updating, deleting and managing Service Metadata, or any internal data storage  
80 formats.

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

## 83 **6 Data model**

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

- 86 • ServiceGroup
- 87 • ServiceMetadata / SignedServiceMetadata

88

89 Supporting data types for these main types are:

- 90 • ServiceList
- 91 • BusinessIdentifier
- 92 • DocumentIdentifier
- 93 • Redirect
- 94 • Process
- 95 • Endpoint

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

### 97 **6.1 On extension points**

98 For each major entity, extension points have been added with the optional `<smp:Extension>` element.

#### 99 **6.1.1 Semantics and use**

100 Child elements of the `<smp:Extension>` element are known as “custom extension elements”. Extension  
101 points may be used for *optional* extensions of service metadata. This implies:

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

107

## 108 **6.2 ServiceGroup**

109 The ServiceGroup structure represents a set of services associated with a specific business identifier that is  
110 handled by a specific Service Metadata Publisher. The ServiceGroup structure holds a list of references to  
111 SignedServiceMetadata resources in the ServiceList structure.

112 Pseudo-schema for ServiceGroup:

```
113 <smp:ServiceGroup>  
114   <ids:BusinessIdentifier scheme="xs:anyURI">  
115     xs:string  
116   </ids:BusinessIdentifier>  
117   <smp:ServiceMetadataReferenceCollection>  
118     <smp:ServiceMetadataReference href="xs:anyURI" /*>  
119   </smp:ServiceMetadataReferenceCollection>  
120   <smp:Extension>xs:any</smp:Extension>?  
121 </smp:ServiceGroup>
```

122

123

124 Description of the individual fields (elements and attributes).

125

| Field                              | Description  |
|------------------------------------|--|
| ServiceGroup                       | Document element   |
| BusinessIdentifier                 | Represents the business level endpoint key and key type, e.g. a DUNS or GLN number that is associated with a group of services. See the BusinessIdentifier section of the 'Common Definitions' document [BDEN-CDEF] 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 <i>SignedServiceMetadata</i> 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, rather than for example. SignedServiceMetadata records hosted 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.  |

126

### 127 6.2.1 Non-normative example

128 Non-normative example of a ServiceGroup resource.

```
129 <?xml version="1.0" encoding="utf-8" ?>
130 <!--
131   This sample assumes that the service metadata publisher resides at
132   "http://serviceMetadata.eu/".
133   It assumes that the business identifier is "0010:5798000000001".
134 -->
135 <ServiceGroup xmlns="http://busdox.org/serviceMetadata/publishing/1.0/"
136 xmlns:ids="http://busdox.org/transport/identifiers/1.0/">
137   <ids:BusinessIdentifier
138     scheme="http://busdox.org/profiles/serviceMetadata/1.0/UniversalBusinessIdentifi
139     er/1.0/">0010:5798000000001</ids:BusinessIdentifier>
140   <ServiceMetadataReferenceCollection>
141     <ServiceMetadataReference
142       href="http://serviceMetadata.eu/smp/http%3a%2f%2fbusdox.org%2fprofiles%2fservice
143       Metadata%2f1.0%2fUniversalBusinessIdentifier%2f1.0%2f%3a%3a0010%3a5798000000001/
144       services/http%3a%2f%2fbusdox.org%2fprofiles%2fserviceMetadata%2f1.0%2fDocumentTy
145       peIdentifierType%2fQNameSubtype%2f%3a%3aurn%3aoasis%3aname%3aspecification%3au
146       bl%3aschema%3axsd%3aInvoice-2%3a%3aInvoice%23%23UBL-2.0" />
147     </ServiceMetadataReferenceCollection>
148     <Extension>
149       <ex:Test xmlns:ex="http://test.eu">Test</ex:Test>
150     </Extension>
151   </ServiceGroup>
152
```

153

## 154 **6.3 ServiceMetadata**

155 This data structure represents Metadata about a specific electronic service. The role of the  
156 ServiceMetadata structure is to associate a business identifier with the ability to receive a specific  
157 document type over a specific transport. It also describes which business processes a document can  
158 participate in, and various operational data such as service activation and expiration times.

159 The *ServiceMetadata* resource contains all the metadata about a service that a sender Access Point needs  
160 to know in order to send the message.

### 161 **6.3.1 Redirection**

162 For recipients that want to associate more than one SMP with their participant identifier, they may redirect  
163 senders to an alternative SMP for specific document types. To achieve this, the ServiceMetadata element  
164 defines the optional element 'Redirect'. This element holds the URL of the alternative SMP, as well as the  
165 Subject Unique Identifier of the destination SMPs certificate used to sign its resources.

166 In the case that a client encounters such a redirection element, the client MUST follow the first redirect  
167 reference to the alternative SMP. If the SignedServiceMetadata resource at the alternative SMP also  
168 contains a redirection element, the client SHOULD NOT follow that redirect.

169 Pseudo-schema for this data type:

```
170 <smp:ServiceMetadata>  
171   [<smp:ServiceInformation /> | <smp:Redirect />]  
172 </smp:ServiceMetadata>  
173
```

174 Pseudo-schema for the 'ServiceInformation' data type:

```
175 <smp:ServiceInformation>  
176   <smp:ServiceActivationDate>xs:dateTime</smp:ServiceActivationDate?>  
177   <smp:ServiceExpirationDate>xs:dateTime</smp:ServiceExpirationDate?>  
178   <smp:CertificateUID>xs:string</smp:CertificateUID>  
179   <smp:ServiceDescription>xs:string</smp:ServiceDescription>  
180   <smp:TechnicalContactUrl>xs:anyURI</smp:TechnicalContactUrl>  
181   <smp:TechnicalInformationUrl>xs:anyURI</smp:TechnicalInformationUrl?>  
182   <ids:DocumentIdentifier scheme="xs:anyURI" />  
183   <smp:ProcessList>  
184     <smp:Process>+  
185       <ids:ProcessIdentifier scheme="xs:anyURI" />  
186       <smp:ServiceEndpointList>  
187         <smp:Endpoint transportProfile="xs:anyURI">+  
188           <wsa:EndpointReference />  
189           <smp:Extension>xs:any</smp:Extension?>  
190         </smp:Endpoint>  
191       </smp:ServiceEndpointList>  
192       <smp:Extension>xs:any</smp:Extension?>  
193     </smp:Process>  
194   </smp:ProcessList>
```

```
195 <ids:BusinessIdentifier scheme="xs:anyURI">xs:string</ids:BusinessIdentifier>
196 <smp:ServiceGroupReference href="xs:anyURI" />
197 <smp:Extension>xs:any</smp:Extension>?
198 </smp:ServiceInformation>
199
```

200 Pseudo-schema for the 'Redirect' data type:

```
201
202 <smp:Redirect href="xs:anyURI">
203   <smp:CertificateUID>xs:string</smp:CertificateUID>
204   <smp:Extension />? <!-- may contain any type -->
205 </smp:Redirect>
206
```

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

209 Note that the 'href' attribute of the Redirect element only contains the address of the Service Metadata  
210 Publisher – not the identifiers and other identifiers and schemes.

211 For example, assume that an SMP, "SMP1" has an address "http://smp1.eu", and another SMP "SMP2" has  
212 an address "http://smp2.eu", and a client requests a resource with the following URL (note that these  
213 examples have been percent encoded using the rules from [BDEN-CDEF]):

```
214 http://smp1.eu/smp/http%3a%2f%2fbusdox.org%2fprofiles%2fserviceMetadata%2f1.0%2f
215 UniversalBusinessIdentifier%2f1.0%2f%3a%3a0010%3a5798000000001/services/http%3a%
216 2f%2fbusdox.org%2fprofiles%2fserviceMetadata%2f1.0%2fDocumentIdentifierType%2fQN
217 ameeSubtype%2f%3a%3aurn%3aoasis%3anames%3aspecification%3aubl%3aschema%3axsd%3aI
218 nvoice-2%3a%3aInvoice%23%23UBL-2.0
```

219 We now assume that the owner of these metadata has moved them to SMP2. SMP1 would then return a  
220 *SignedServiceMetadata* resource with a *Redirect* child element that has the "href" attribute set to  
221 "http://smp2.eu", rather than the full URL to the resource.

222 The client would then have to add the service-specific part of the URL to that SMP address in order to fetch  
223 the resource from it. The resulting client request would then be:

```
224 http://smp2.eu/smp/http%3a%2f%2fbusdox.org%2fprofiles%2fserviceMetadata%2f1.0%2f
225 UniversalBusinessIdentifier%2f1.0%2f%3a%3a0010%3a5798000000001/services/http%3a%
226 2f%2fbusdox.org%2fprofiles%2fserviceMetadata%2f1.0%2fDocumentIdentifierType%2fQN
227 ameeSubtype%2f%3a%3aurn%3aoasis%3anames%3aspecification%3aubl%3aschema%3axsd%3aI
228 nvoice-2%3a%3aInvoice%23%23UBL-2.0
229
230
```

231 Description of the individual fields (elements and attributes).

232

| Field  | Description  |
|--|--|
| ServiceMetadata                                | Document element   |
| /Redirect                                      | The direct child element of <i>ServiceMetadata</i> is either the <i>Redirect</i> element or the <i>ServiceInformation</i> element. The <i>Redirect</i> element indicates that a client must follow the URL of the <i>href</i> attribute of this element. Note that the 'href' attribute of the <i>Redirect</i> element only contains the address of the Service Metadata Publisher – not the identifiers and other identifiers and schemes – see description in the section above. |
| /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 service metadata.  |
| /ServiceInformation                            | The direct child element of <i>ServiceMetadata</i> is either the <i>Redirect</i> element or the <i>ServiceInformation</i> element. The <i>ServiceInformation</i> element contains service information for an actual service registration, rather than a redirect to another SMP.   |
| ServiceInformation/<br>ServiceActivationDate   | Activation date of the service. Senders should ignore services that are not yet activated, or expired. Format of <i>ServiceActivationDate</i> date is xs:dateTime  |
| ServiceInformation/<br>ServiceExpirationDate   | Expiration date of the service. Senders should ignore services that are not yet activated, or expired. Format of <i>ServiceExpirationDate</i> date is xs:dateTime.   |
| ServiceInformation/<br>CertificateUID          | Holds a unique identifier of the signing certificate of the recipient endpoint. This could for example be the CN, or Certificate Subject Identifier, or other fields that uniquely identifies the certificate according to its policy.   |
| ServiceInformation/<br>ServiceDescription      | A human readable description of the service  |
| ServiceInformation/<br>TechnicalContactUrl     | Represents a link to human readable contact information. This might also be an email address.  |
| ServiceInformation/<br>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.  |
| ServiceInformation/<br>DocumentIdentifier      | Represents the type of document that the recipient is able to handle. The document is represented by an identifier (identifying the document type) and an identifier scheme, which the format of the identifier itself. See the <i>DocumentIdentifier</i> section of the 'Common Definitions'  |

|   |   |
|---|---|
|   | document [BDEN-CDEF] for information on this data type.   |
| ServiceInformation/<br>ProcessList                                  | Represents the processes that a specific document type can participate in, and endpoint address and binding information. See section 6.5 for a description of this type. Also see the <i>Process</i> section of the 'Common Definitions' document [BDEN-CDEF] for information on the identifier format.             |
| ServiceInformation/<br>ProcessList/./Endpoint                       | Endpoint represents the technical endpoint and address type of the recipient, as an URL. The 'transportProfile' attribute   |
| ServiceInformation/<br>ProcessList/./Endpoint/<br>@transportProfile | Indicates the type of BUSDOX transport that is being used between access points, e.g. the BUSDOX START profile. This specification defines the following identifier URI which denotes the BUSDOX START transport: "http://busdox.org/profiles/serviceMetadata/1.0/start/1.0/"                                       |
| ServiceInformation/<br>BusinessIdentifier                           | The business 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 BusinessIdentifier section of the 'Common Definitions' document [BDEN-CDEF] for information on this data type. |
| ServiceInformation/<br>ServiceGroupReference                        | The 'href' attribute holds the URL to the 'ServiceGroup' resource from which it was referenced.   |
| ServiceInformation/<br>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.   |

233

### 234 6.3.2 Non-normative example

235 For a non-normative example of a ServiceMetadata resource, see the SignedServiceMetadata non-  
236 normative example below.

## 237 6.4 SignedServiceMetadata

238 The SignedServiceMetadata structure is a ServiceMetadata structure that has been signed by the  
239 ServiceMetadataPublisher, according to governance policies that are not covered by this document.

240 Pseudo-schema for this data type:

```
241 <smp:SignedServiceMetadata>
242   <smp:ServiceMetadata />
243   <ds:Signature />
244 </smp:SignedServiceMetadata>
```

245

- 246 • **ServiceMetadata** (1..1): The ServiceMetadata element covered by the signature.
- 247 • **Signature** (1..1) represents an enveloped XML signature over the ServiceMetadata element.

248

## 249 6.4.1 Non-normative example

250 Non-normative example of a SignedServiceMetadata resource.

```
251 <?xml version="1.0" encoding="utf-8" ?>
252 <!--
253   This sample assumes that the service metadata publisher resides at
254   "http://serviceMetadata.eu/".
255   It assumes that the business identifier is "0010:5798000000001".
256   It assumes that the documenttype identifier is
257   "urn:oasis:names:specification:ubl:schema:xsd:Invoice-2::Invoice##UBL-2.02"
258   -->
259 <SignedServiceMetadata xmlns="http://busdox.org/serviceMetadata/publishing/1.0/"
260 xmlns:ids="http://busdox.org/transport/identifiers/1.0/"
261   <ServiceMetadata
262     wsu:Id="_1"
263     xmlns="http://busdox.org/serviceMetadata/publishing/1.0/"
264     xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
265 wssecurity-utility-1.0.xsd">
266     <ServiceInformation>
267       <ServiceActivationDate>2009-05-01T09:00:00</ServiceActivationDate>
268       <ServiceExpirationDate>2016-05-01T09:00:00</ServiceExpirationDate>
269       <CertificateUID>PID:9208-2001-3-279815395</CertificateUID>
270       <ServiceDescription>invoice service</ServiceDescription>
271       <TechnicalContactUrl>http://example.com</TechnicalContactUrl>
272       <TechnicalInformationUrl>http://example.com/info</TechnicalInformationUrl>
273       <ids:DocumentIdentifier
274 scheme="http://busdox.org/profiles/serviceMetadata/1.0/DocumentTypeIdentifierTyp
275 e/QNameSubtype/">
276         urn:oasis:names:specification:ubl:schema:xsd:Invoice-2::Invoice##UBL-
277 2.02
278       </ids:DocumentIdentifier>
279       <ProcessList>
280         <Process>
281           <ids:ProcessIdentifier
282 scheme="http://busdox.org/processIdentifiers/1.0/UBL/CENBII/1.0/">BII04</ids:Pro
283 cessIdentifier>
284           <ServiceEndpointList>
285             <Endpoint
286 transportProfile="http://busdox.org/profiles/serviceMetadata/1.0/start/1.0/"
287               <EndpointReference xmlns="http://www.w3.org/2005/08/addressing">
288                 <Address>http://busdox.org/sampleService/</Address>
289               </EndpointReference>
290             </Endpoint>
291           </ServiceEndpointList>
292         </Process>
293         <Process>
294           <ids:ProcessIdentifier
295 scheme="http://oio.dk/profiles/OWSA/modelT/1.0/UDDI/Identifiers/UBL/CENBII/1.0/"
296 >BII07</ids:ProcessIdentifier>
297           <ServiceEndpointList>
298             <Endpoint
299 transportProfile="http://busdox.org/profiles/serviceMetadata/1.0/start/1.0/"
300               <EndpointReference xmlns="http://www.w3.org/2005/08/addressing">
301                 <Address>http://busdox.org/sampleService/</Address>
302               </EndpointReference>
303             <Extension>
304               <ex:Test xmlns:ex="http://test.eu">Test</ex:Test>
```

```

305         </Extension>
306     </Endpoint>
307 </ServiceEndpointList>
308 <Extension>
309     <ex:Test xmlns:ex="http://test.eu">Test</ex:Test>
310 </Extension>
311 </Process>
312 </ProcessList>
313 <ids:BusinessIdentifier
314 scheme="http://busdox.org/profiles/serviceMetadata/1.0/UniversalBusinessIdentifi
315 er/1.0/">
316     0010:5798000000001
317 </ids:BusinessIdentifier>
318 <ServiceGroupReference
319 href="http://serviceMetadata.eu/smp/http%3a%2f%2fbusdox.org%2fprofiles%2fservice
320 Metadata%2f1.0%2fUniversalBusinessIdentifier%2f1.0%2f%3a%3a0010%3a5798000000001"
321 />
322     <Extension>
323         <ex:Test xmlns:ex="http://test.eu">Test</ex:Test>
324     </Extension>
325 </ServiceInformation>
326 </ServiceMetadata>
327 <!-- Message signature, details omitted for brevity -->
328 <Signature xmlns="http://www.w3.org/2000/09/xmldsig#" />
329
330 </SignedServiceMetadata>
331
332
333

```

#### 334 6.4.2 Redirect, non-normative example

```

335 <?xml version="1.0" encoding="utf-8" ?>
336 <!--
337     This sample assumes that the user contacts a service metadata publisher that
338     resides at "http://serviceMetadata.eu/",
339     but is redirected to a service metadata publisher that resides at
340     "http://serviceMetadata2.eu".
341     -->
342
343 <SignedServiceMetadata
344 xmlns="http://busdox.org/serviceMetadata/publishing/1.0/">
345     <ServiceMetadata
346         xmlns="http://busdox.org/serviceMetadata/publishing/1.0/">
347         <Redirect xmlns="http://busdox.org/serviceMetadata/publishing/1.0/"
348 href="http://serviceMetadata2.eu">
349             <CertificateUID>PID:9208-2001-3-279815395</CertificateUID>
350             <Extension>
351                 <ex:Test xmlns:ex="http://test.eu">Test</ex:Test>
352             </Extension>
353         </Redirect>
354     </ServiceMetadata>
355     <!-- Message signature, details omitted for brevity -->
356     <Signature xmlns="http://www.w3.org/2000/09/xmldsig#" />
357 </SignedServiceMetadata>
358
359
360
361

```

362 **6.5 ProcessList**

363 For a definition of Process Identifiers, see the ‘Common Definitions’ document [BDEN-CDEF].The  
 364 ProcessList element represents a process that a specific document type can participate in. Each process  
 365 element describes a specific business process that accepts this type of document as input and holds the  
 366 endpoint addresses of services that implement the business process, plus information about the transport  
 367 used for each endpoint.

368 This element also holds the endpoint address of a specific service that will accept a document in this  
 369 specific process, as well as information on the transport that the endpoint uses.

370 Pseudo-schema for ProcessList:

```

371 <sm:ProcessList>
372   <sm:Process>+
373     <sm:ProcessIdentifier scheme="xs:anyURI" />
374     <sm:ServiceEndpointList>
375       <sm:Endpoint transportProfile="xs:anyURI">+
376         <wsa:EndpointReference />
377         <sm:Extension>xs:any</sm:Extension?>
378       </sm:Endpoint>
379     </sm:ServiceEndpointList>
380     <sm:Extension>xs:any</sm:Extension?>
381   </sm:Process>
382 </sm:ProcessList>
  
```

383 Description of the individual fields (elements and attributes).

384

| Field                      | Description   |
|----------------------------|---|
| ProcessIdentifier          | The identifier of the process. See the ‘Common Definitions’ document for a definition of process identifiers [BDEN-CDEF]  |
| ProcessIdentifier/@scheme  | Indicates the scheme of the process identifier. See the ‘Common Definitions’ document for a definition of process identifiers [BDEN-CDEF]   |
| ServiceEndpointList        | List of one or more endpoints that support this process.  |
| Endpoint/EndpointReference | The address of an endpoint, as an WS-Addressing Endpoint Reference (EPR)  |
| Endpoint/Extension         | The extension element may contain any XML element. Clients MAY ignore this element. It can be used to add extension metadata to individual endpoints within a process.  |
| Endpoint/@transportProfile | Indicates the type of BUSDOX transport that is being used between access points, e.g. the BUSDOX START profile. This specification defines the following identifier URI which denotes the BUSDOX START transport: "http://busdox.org/profiles/serviceMetadata/1.0/start/1.0/" |
| 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.  |

385

386

387 **7 Service Metadata Publishing REST binding**

388 This section describes the REST binding of the Service Metadata Publishing interface.

389 **7.1 The use of HTTP**

390 A service implementing the REST binding MUST set the HTTP “content-type” header, and give it a value of  
391 “text/xml”. A service implementing the REST profile MUST NOT use TLS (Transport Layer Security) or SSL  
392 (Secure Sockets Layer). An instance of the BUSDOX infrastructure MAY set restrictions on what ports are  
393 allowed.

394 This specification does not mandate support for PUT or DELETE.

395 **7.1.1 HTTP status codes**

396 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 business identifier that does not exist. |
| 500              | Code 500 must be returned if the service experiences an internal processing error.   |

397

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

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

401 **7.2 The use of XML and encoding**

402 XML document returned by HTTP GET MUST be UTF-8 encoded. They MUST contain a document type  
403 declaration starting with “<?xml” which includes the ‘encoding’ attribute set to “UTF-8”.

404

405 **7.3 Resources and identifiers**

406 The REST interface comprises 2 types of resources.

| Resource                | URI  | Method | XML resource root element | HTTP Status         | Description   |
|-------------------------|--|--------|---------------------------|---------------------|---|
| ServiceGroup            | /smp/{identifier scheme}::{id}   | GET    | <ServiceGroup>            | 200;<br>500;<br>404 | Holds the business identifier of the recipient, and a list of references to individual ServiceMetadata resources that are associated with that business identifier. |
| Signed Service Metadata | /smp/{identifier scheme}::{id}/services/ {docType}<br><br>See section below for {docType} format | GET    | <SignedServiceMetadata>   | 200;<br>500;<br>404 | Holds all of the metadata about a Service, or a redirection URL to another Service Metadata Publisher holding this information.                                     |

407 **Fig. 3: Table of resources and identifiers**

408 A service implementing the REST binding MUST support these resource types. It MUST provide access to  
409 these using the URI scheme of table in Fig. 3.

410 Note the path “/smp/” which it is mandatory to include in the URL.

411 **7.3.1 On the use of percent encoding**

412 See the ‘Common Definitions’ document for requirements for percent encoding [BDEN-CDEF].

413 **7.3.2 Using identifiers in the REST Resource URLs**

414 This section describes specifically how business and document identifiers are used to reference  
415 <ServiceGroup> and <SignedServiceMetadata> REST resources. For a general definition on how to  
416 represent business and document identifiers in URLs, see the ‘Common Definitions’ document [BDEN-  
417 CDEF].

418 For the URL referencing a <ServiceGroup> resource, the “{identifier scheme}::{id}” part follows the business  
419 identifier format described in the “BusinessIdentifier” section of the ‘Common Definitions’ document  
420 [BDEN-CDEF].

421

422 The following URL format is used:

423 /smp/{identifier scheme}::{id}

424 In the reference to the SignedServiceMetadata or Redirect elements (/smp/{id}/services/ {docType}), the  
425 {docType} part consists of {document identifier scheme}::{document identifier}. For information on the  
426 format of {document identifier}, see the DocumentIdentifier section of the 'Common Definitions' document  
427 [BDEN-CDEF].

### 428 7.3.3 Non-normative identifier example

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

430 A business with the business identifier "0010:5798000000001" would have the following identifier for the  
431 ServiceGroup resource:

432 http://serviceMetadata.eu/smp/http://busdox.org/profiles/serviceMetadata/1.0/UniversalBusinessIdentifi  
433 er/1.0/::0010:5798000000001

434 After percent encoding:

435 **http://serviceMetadata.eu/smp/http%3a%2f%2fbusdox.org%2fprofiles%2fserviceMetadata%2f1.0%2fUni**  
436 **versalBusinessIdentifier%2f1.0%2f%3a%3a0010%3a5798000000001**

437 Note that the "/smp" part and forward slash preceding the {identifier scheme}::{id} part is *not* percent  
438 encoded, since it belongs to the URL.

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

- 440 • **Identifier format type:**

441 http://busdox.org/profiles/serviceMetadata/1.0/DocumentIdentifierType/QNameSubtype/

- 442 • **Root namespace:** urn:oasis:names:specification:ubl:schema:xsd:Order-2

- 443 • **Document element local name:** Order

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

446 The document type identifier will then be:

447 http://busdox.org/profiles/serviceMetadata/1.0/DocumentIdentifierType/QNameSubtype/::urn:oasis:nam  
448 es:specification:ubl:schema:xsd:Order-2::Order##UBL-2.0

449 The document type identifier MUST be percent encoded as described in [RFC3986]. The above, non-  
450 normative example is thus encoded to

451 **http%3a%2f%2fbusdox.org%2fprofiles%2fserviceMetadata%2f1.0%2fDocumentIdentifierType%2fQNameSubtype%2f%3a%3aurn%3aoasis%3anames%3aspecification%3aubl%3aschema%3axsd%3aOrder-**  
452 **2%3a%3aOrder%23%23UBL-2.0**

454

455 The entire URL reference to a SignedServiceMetadata or Redirect element thus has the form

456 {URL to server}/smp/{identifier scheme}::{id}/services/{document identifier  
457 type}::{rootNamespace}::{documentElementLocalName}[##{Subtype identifier}]

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

459 ***http://serviceMetadata.eu/smp/http%3a%2f%2fbusdox.org%2fprofiles%2fServiceMetadata%2f1.0%2fUni***  
460 ***versalBusinessIdentifier%2f1.0%2f%3a%3a0010%3a579800000001/services/http%3a%2f%2fbusdox.org%***  
461 ***2fprofiles%2fServiceMetadata%2f1.0%2fDocumentIdentifierType%2fQNameSubtype%2f%3a%3aurn%3aoo***  
462 ***sis%3anames%3aspecification%3aubl%3aschema%3axsd%3aInvoice-2%3a%3aInvoice%23%23UBL-2.0***

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

#### 465 **7.3.4 Additional constraints on the Resource Identifier URL**

466 The “/{identifier scheme}::{id}” part of the ServiceGroupReference ‘href’ attribute MUST correspond to the  
467 /{identifier scheme}::{id} part of the URL of the corresponding ServiceGroup resource.

#### 468 **7.3.5 Implementation considerations**

469 When a client is redirected to an SMP using the DNS-based SML scheme described in [BDEN-SML], the HTTP  
470 host header will be set to a value originating from the CNAME alias set in the SML. Implementations should  
471 be prepared to accept requests with this host header value.

### 472 **7.4 Referencing the SMP REST binding**

473 For referencing the SMP REST binding, for example from Service Metadata Locator records, the following  
474 identifier should be used for the version 1.0 of the SMP REST binding:

475 `http://busdox.org/serviceMetadata/publishing/1.0/`

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

## 477 **7.5 Security**

478 At the transport level, the service is not secured.

### 479 **7.5.1 Message signature**

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

482 The signature MUST be an enveloped XML signature represented via an <ds:Signature> element embedded  
483 in the <SignedServiceMetadata> element. The <ds:Signature> element MUST be constructed according to  
484 the following rules:

- 485 • The <Reference> MUST use exactly one Transform being:  
486 “http://www.w3.org/2000/09/xmlsig#envelopedsignature”
- 487 • The <ds:KeyInfo> element MUST contain an <ds:X509Data> element with an <ds:X509Certificate>  
488 sub-element containing the signer’s X.509 certificate as a base64 encoded value.
- 489 • The canonicalization algorithm MUST be `http://www.w3.org/2001/10/xml-exc-c14n#`

- 490 • The SignatureMethod MUST be <http://www.w3.org/2000/09/xmldsig#rsa-sha1>
- 491 • The DigestMethod MUST be <http://www.w3.org/2000/09/xmldsig#sha1>

## 492 **7.5.2 Verifying the signature**

493 When verifying the signature, the consumer may not know the specific certificate, but only have trust in a  
494 known root certificate. In this case, the consumer could verify the signature by a) extracting the certificate  
495 from the <ds:X509Data> element, b) verify that it has been issued by the trusted root, c) perform a  
496 validation of the signature, and c) perform the required certificate validation steps (which might include  
497 checking expiration/activation dates and revocation lists).

498

499

500

## 501 **8 Appendix A: Schema for the REST interface**

502 This section defines the XML Schema for all the resources of the REST interface.

```
503 <?xml version="1.0" encoding="utf-8"?>
504 <xs:schema id="ServiceMetadataPublishing"
505           targetNamespace="http://busdox.org/serviceMetadata/publishing/1.0/"
506           elementFormDefault="qualified"
507           xmlns="http://busdox.org/serviceMetadata/publishing/1.0/"
508           xmlns:ids="http://busdox.org/transport/identifiers/1.0/"
509           xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
510           xmlns:xs="http://www.w3.org/2001/XMLSchema"
511           xmlns:wsa="http://www.w3.org/2005/08/addressing">
512   <xs:import schemaLocation="xmldsig-core-schema.xsd"
513   namespace="http://www.w3.org/2000/09/xmldsig#" />
514   <xs:import schemaLocation="oasis-200401-wss-wssecurity-utility-1.0.xsd"
515   namespace="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-
516   utility-1.0.xsd" />
517   <xs:import schemaLocation="ws-addr.xsd"
518   namespace="http://www.w3.org/2005/08/addressing" />
519   <xs:import schemaLocation="Identifiers-0.9.xsd"
520   namespace="http://busdox.org/transport/identifiers/1.0/" />
521
522   <xs:element name="ServiceGroup" type="ServiceGroupType"/>
523   <xs:element name="ServiceMetadata" type="ServiceMetadataType"/>
524   <xs:element name="SignedServiceMetadata" type="SignedServiceMetadataType"/>
525
526   <xs:complexType name="SignedServiceMetadataType">
527     <xs:sequence>
528       <xs:element ref="ServiceMetadata"/>
529       <xs:element ref="ds:Signature" />
530     </xs:sequence>
531   </xs:complexType>
532
533   <xs:complexType name="ServiceMetadataType">
534     <xs:sequence>
535       <xs:choice>
536         <xs:element name="ServiceInformation" type="ServiceInformationType"/>
537         <xs:element name="Redirect" type="RedirectType"/>
538       </xs:choice>
539     </xs:sequence>
540   </xs:complexType>
541
542   <xs:complexType name="ServiceInformationType">
543     <xs:sequence>
544       <xs:element name="ServiceActivationDate" type="xs:dateTime" minOccurs="0"
545   />
546       <xs:element name="ServiceExpirationDate" type="xs:dateTime" minOccurs="0"
547   />
548       <xs:element name="CertificateUID" type="xs:string" />
549       <xs:element name="ServiceDescription" type="xs:string" />
550       <xs:element name="TechnicalContactUrl" type="xs:anyURI" />
551       <xs:element name="TechnicalInformationUrl" type="xs:anyURI" minOccurs="0"
552   />
553       <xs:element ref="ids:DocumentIdentifier" />
```

```

554     <xs:element name="ProcessList" type="ProcessListType" />
555     <xs:element ref="ids:BusinessIdentifier" />
556     <xs:element name="ServiceGroupReference" type="ServiceGroupReferenceType"
557 />
558     <xs:element name="Extension" type="ExtensionType" minOccurs="0" />
559 </xs:sequence>
560 </xs:complexType>
561
562 <xs:complexType name="ProcessListType">
563     <xs:sequence>
564         <xs:element name="Process" type="ProcessType" maxOccurs="unbounded" />
565     </xs:sequence>
566 </xs:complexType>
567
568 <xs:complexType name="ProcessType">
569     <xs:sequence>
570         <xs:element ref="ids:ProcessIdentifier" />
571         <xs:element name="ServiceEndpointList" type="ServiceEndpointList"/>
572         <xs:element name="Extension" type="ExtensionType" minOccurs="0" />
573     </xs:sequence>
574 </xs:complexType>
575
576 <xs:complexType name="ServiceEndpointList">
577     <xs:sequence>
578         <xs:element name="Endpoint" type="EndpointType" maxOccurs="unbounded" />
579     </xs:sequence>
580 </xs:complexType>
581
582 <xs:complexType name="EndpointType">
583     <xs:sequence>
584         <xs:element ref="wsa:EndpointReference"/>
585         <xs:element name="Extension" type="ExtensionType" minOccurs="0"/>
586     </xs:sequence>
587     <xs:attribute name="transportProfile" type="xs:anyURI" />
588 </xs:complexType>
589
590 <xs:complexType name="ServiceGroupType">
591     <xs:sequence>
592         <xs:element ref="ids:BusinessIdentifier" />
593         <xs:element name="ServiceMetadataReferenceCollection"
594 type="ServiceMetadataReferenceCollectionType" />
595         <xs:element name="Extension" type="ExtensionType" minOccurs="0"/>
596     </xs:sequence>
597 </xs:complexType>
598 <xs:complexType name="ServiceMetadataReferenceCollectionType">
599     <xs:sequence>
600         <xs:element name="ServiceMetadataReference"
601 type="ServiceMetadataReferenceType" minOccurs="0" maxOccurs="unbounded" />
602     </xs:sequence>
603 </xs:complexType>
604 <xs:complexType name="ServiceMetadataReferenceType">
605     <xs:attribute name="href" type="xs:anyURI" />
606 </xs:complexType>
607 <xs:complexType name="ServiceGroupReferenceType">
608     <xs:simpleContent>
609         <xs:extension base="xs:string">
610             <xs:attribute name="href" type="xs:anyURI" />
611         </xs:extension>
612     </xs:simpleContent>

```

```
613 </xs:complexType>
614 <xs:complexType name="RedirectType">
615   <xs:sequence>
616     <xs:element name="CertificateUID" type="xs:string" />
617     <xs:element name="Extension" type="ExtensionType" minOccurs="0"/>
618   </xs:sequence>
619   <xs:attribute name="href" type="xs:anyURI" />
620 </xs:complexType>
621 <xs:complexType name="ExtensionType">
622   <xs:sequence>
623     <xs:any />
624   </xs:sequence>
625 </xs:complexType>
626 </xs:schema>
627
628
629
630
631
```