



Project number <b>IEC 62325-504 TS Ed.1</b>	
IEC/TC or SC <b>57</b>	Secretariat <b>Germany</b>
Distributed on <b>2014-11-07</b>	Voting terminates on <b>2015-02-13</b>

Also of interest to the following committees	Supersedes document <b>57/1467/CD - 57/1506/CC</b>		
Functions concerned			
<input type="checkbox"/> Safety	<input type="checkbox"/> EMC	<input type="checkbox"/> Environment	<input type="checkbox"/> Quality assurance

THIS DOCUMENT IS STILL UNDER STUDY AND SUBJECT TO CHANGE. IT SHOULD NOT BE USED FOR REFERENCE PURPOSES.

RECIPIENTS OF THIS DOCUMENT ARE INVITED TO SUBMIT, WITH THEIR COMMENTS, NOTIFICATION OF ANY RELEVANT PATENT RIGHTS OF WHICH THEY ARE AWARE AND TO PROVIDE SUPPORTING DOCUMENTATION.

Title

**IEC 62325-504 TS:  
Framework for energy market communications –  
Part 504: Utilization of web services for electronic data interchanges on the  
European energy market for electricity**

**Copyright © 2014 International Electrotechnical Commission, IEC.** All rights reserved. It is permitted to download this electronic file, to make a copy and to print out the content for the sole purpose of preparing National Committee positions. You may not copy or "mirror" the file or printed version of the document, or any part of it, for any other purpose without permission in writing from IEC.

CONTENTS

1

2

3 1 Scope.....5

4 1.1 General.....5

5 1.2 Overview.....5

6 2 Normative References .....5

7 3 Terms, definitions and namespaces.....6

8 3.1 Terms and definitions .....6

9 3.2 Namespaces .....6

10 4 Conformance.....7

11 4.1 General.....7

12 4.2 Client Application Conformance.....7

13 4.3 Server Conformance .....7

14 5 Service Definitions .....8

15 5.1 List Messages .....8

16 5.1.1 General .....8

17 5.1.2 Service Request .....8

18 5.1.3 Service Response .....9

19 5.1.4 Functional Requirements .....10

20 5.2 Get Message.....10

21 5.2.1 General .....10

22 5.2.2 Service Request .....10

23 5.2.3 Service Response .....11

24 5.2.4 Functional Requirements .....11

25 5.3 Put Message .....11

26 5.3.1 General .....11

27 5.3.2 Service Request .....11

28 5.3.3 Service Response .....12

29 5.3.4 Functional Requirements .....12

30 5.4 Query Data.....12

31 5.4.1 General .....12

32 5.4.2 Service Request .....12

33 5.4.3 Service Response .....12

34 5.4.4 Functional Requirements .....12

35 6 Applying IEC 61968-100.....14

36 6.1 Integration Pattern.....14

37 6.1.1 General .....14

38 6.1.2 List Service .....14

39 6.1.3 Get Service .....14

40 6.1.4 Put Service.....15

41 6.2 Service Mapping.....16

42 6.2.1 General .....16

43 6.2.2 Header Values.....16

44 6.2.3 Request Values .....16

45 6.2.4 Response Values .....16

46	6.2.5 Payload Values .....	17
47	7 Schema Definitions.....	17
48	7.1 Common Definitions .....	17
49	7.2 List Message .....	17
50	7.3 QueryData Message.....	18
51	7.4 QueryData List of Data Types.....	19
52	8 Service Provider WSDL Abstract Definitions .....	19
53	9 Service Provider WSDL SOAP Binding .....	20
54	10 Security.....	21
55	Annex A (normative) XML schema for common IEC 62325-504 messages .....	23
56	Annex B (Informative) Message Examples .....	25
57	B.1 List.....	25
58	B.1.1 Basic Example.....	25
59	B.2 Get.....	26
60	B.2.1 Basic Example.....	27
61	B.3 Put .....	29
62	B.3.1 Basic Example.....	29
63	B.3.2 Example with binary data.....	31
64	B.4 Query Data.....	31
65	B.4.1 List of Data Types Example .....	31
66	B.4.2 Server Timestamp Request Example .....	32
67	B.4.3 Server Perimeter Limits Request Example .....	33
68	B.4.4 Generic Query Example.....	34
69	B.5 Fault.....	36
70	B.5.1 SOAP 1.2 .....	36
71	B.5.2 SOAP 1.1 .....	36
72	B.6 Digital Signature.....	36
73	B.6.1 Basic Example.....	36
74	Annex C (Informative) Java code Examples .....	39
75	C.1 Sign .....	39
76	C.2 Verify .....	40
77	Annex D (Informative) Regarding Near Real-time Communications .....	41
78		

79

## INTERNATIONAL ELECTROTECHNICAL COMMISSION

80

81

82

**Technical Specification for the utilization of web services for electronic data interchanges on the European Energy Market for Electricity**

83

84

85

## FOREWORD

86

87

88

89

90

91

92

93

94

95

1) The International Electrotechnical Commission (IEC) is a worldwide organization for standardization comprising all national electrotechnical committees (IEC National Committees). The object of IEC is to promote international co-operation on all questions concerning standardization in the electrical and electronic fields. To this end and in addition to other activities, IEC publishes International Standards, Technical Specifications, Technical Reports, Publicly Available Specifications (PAS) and Guides (hereafter referred to as "IEC Publication(s)"). Their preparation is entrusted to technical committees; any IEC National Committee interested in the subject dealt with may participate in this preparatory work. International, governmental and non-governmental organizations liaising with the IEC also participate in this preparation. IEC collaborates closely with the International Organization for Standardization (ISO) in accordance with conditions determined by agreement between the two organizations.

96

97

98

2) The formal decisions or agreements of IEC on technical matters express, as nearly as possible, an international consensus of opinion on the relevant subjects since each technical committee has representation from all interested IEC National Committees.

99

100

101

102

3) IEC Publications have the form of recommendations for international use and are accepted by IEC National Committees in that sense. While all reasonable efforts are made to ensure that the technical content of IEC Publications is accurate, IEC cannot be held responsible for the way in which they are used or for any misinterpretation by any end user.

103

104

105

106

4) In order to promote international uniformity, IEC National Committees undertake to apply IEC Publications transparently to the maximum extent possible in their national and regional publications. Any divergence between any IEC Publication and the corresponding national or regional publication shall be clearly indicated in the latter.

107

108

109

5) IEC itself does not provide any attestation of conformity. Independent certification bodies provide conformity assessment services and, in some areas, access to IEC marks of conformity. IEC is not responsible for any services carried out by independent certification bodies.

110

111

112

113

114

115

6) All users should ensure that they have the latest edition of this publication.

7) No liability shall attach to IEC or its directors, employees, servants or agents including individual experts and members of its technical committees and IEC National Committees for any personal injury, property damage or other damage of any nature whatsoever, whether direct or indirect, or for costs (including legal fees) and expenses arising out of the publication, use of, or reliance upon, this IEC Publication or any other IEC Publications.

116

117

8) Attention is drawn to the Normative references cited in this publication. Use of the referenced publications is indispensable for the correct application of this publication.

118

119

9) Attention is drawn to the possibility that some of the elements of this IEC Publication may be the subject of patent rights. IEC shall not be held responsible for identifying any or all such patent rights.

120

121

122

The main task of IEC technical committees is to prepare International Standards. In exceptional circumstances, a technical committee may propose the publication of a technical specification when

123

124

- the required support cannot be obtained for the publication of an International Standard, despite repeated efforts, or

125

126

- the subject is still under technical development or where, for any other reason, there is the future but no immediate possibility of an agreement on an International Standard.

127

128

Technical specifications are subject to review within three years of publication to decide whether they can be transformed into International Standards.

129

130

IEC 62325-504, which is a technical specification, has been prepared by IEC technical committee 57: Power systems management and associated information exchange.

131

132

133 The text of this technical specification is based on the following documents:

Enquiry draft	Report on voting
XX/XX/DTS	XX/XX/RVC

134  
135 Full information on the voting for the approval of this technical specification can be found in  
136 the report on voting indicated in the above table.

137 This publication has been drafted in accordance with the ISO/IEC Directives, Part 2.

138 The committee has decided that the contents of this publication will remain unchanged until  
139 the stability date indicated on the IEC web site under "http://webstore.iec.ch" in the data  
140 related to the specific publication. At this date, the publication will be

- 141 • transformed into an International Standard,
- 142 • reconfirmed,
- 143 • withdrawn,
- 144 • replaced by a revised edition, or
- 145 • amended.

146

147 **Document history**

148 Any person intervening in the present document is invited to complete the table below before  
149 sending the document elsewhere. The purpose is to allow all actors to see all changes  
150 introduced and the intervening persons.

151 Any important message to IEC editors should also be included in the table below.

Name of intervening person	Document received		Brief description of the changes introduced	Document sent	
	From	Date		To	Date
Margareta Nöth	James Waight	2014-04-24	CD document	CO	2014-04-28
Margareta Nöth	Agustín Martín Barbero	2014-09-29	English version of DTS document for translation	CO	2014-11-05

152 This table will be removed by IEC editors before FDIS circulation (in case of IS) or before final  
153 publication (in case of TS or TR).

154

155 **Technical Specification for the utilization of web services for electronic data**  
156 **interchanges on the European Energy Market for Electricity**

157 **1 Scope**

158 **1.1 General**

159 This document defines the services needed to support the electronic data interchanges  
160 between different actors on the European Energy Market for Electricity (EME) in a fast (near-  
161 realtime), and secure way. At the same time, this Technical Specification can also be applied  
162 to integration problems outside the scope of IEC 62325-451, such as to the integration of gas  
163 market systems or general enterprise integration

164 Web Services (in WSDL) will be specified for the defined services, applying the Basic Web  
165 Service Pattern implementation profile from IEC 61968-100.

166 **1.2 Overview**

167 The services needed to support the electronic data interchange on the European Energy  
168 Market for Electricity are:

- 169 • List Messages. This service is used by a client application identified with the credentials of  
170 an EME Actor to request a list of messages available on the server for retrieval.
- 171 • Get Message. This service is used by a client application identified with the credentials of  
172 an EME Actor to request a specific message available on the server.
- 173 • Put Message. This service is used by a client application to send a message, usually  
174 providing data related to a Market Participant in the Energy Market for Electricity, to the  
175 server for processing.

176 **2 Normative References**

177 The following referenced documents are indispensable for the application of this Technical  
178 Specification. For dated references, only the edition cited applies. For undated references,  
179 the latest edition of the referenced document (including any amendments) applies.

180 WS-I Basic Profile 1.1, <http://www.ws-i.org/Profiles/BasicProfile-1.1-2006-04-10.html>

181 WSDL, *Web Services Description Language (WSDL) 1.1*

182 IEC-40210, *W3C SOAP Version 1.2 Part I: Messaging Framework (2<sup>nd</sup> Edition)*

183 XML Schema 1.0: XML Schema Language Part 1: Structure, W3C Recommendation 28  
184 October 2004; XML Schema Language Part 2: Data Types, W3C Recommendation 28  
185 October 2004

186 XML Signature Syntax and Processing (Second Edition) <http://www.w3.org/TR/xmlsig-core>

187 IEC-62325 Part 451-1: *Framework for energy market communications - Acknowledgement*  
188 *business process and contextual model for CIM European market.*

189 IEC-61968-100: *Implementation Profiles for IEC 61968*

190 RFC 6176: *Prohibiting SSL 2.0* <http://tools.ietf.org/html/rfc6176>

191 RFC 5280: *Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List*  
192 *(CRL) Profile* <http://tools.ietf.org/rfc/rfc5280>

193 RFC 6818: *Updates to the Internet X.509 Public Key Infrastructure Certificate and Certificate*  
194 *Revocation List (CRL) Profile* <http://tools.ietf.org/rfc/rfc6818>

195 RFC 4346: *The Transport Layer Security (TLS) Protocol V1.1* <http://www.ietf.org/rfc/rfc4346>

## 196 3 Terms, definitions and namespaces

### 197 3.1 Terms and definitions

#### 198 3.1.1

#### 199 **Message Identification**

200 Alphanumeric string that represents the name of a message in the system.

#### 201 3.1.2

#### 202 **Version**

203 Number that represents the message version. The range of values is from 1 to 999.

#### 204 3.1.3

#### 205 **Application Time Interval**

206 Time interval when the message payload applies.

#### 207 3.1.4

#### 208 **Server Timestamp**

209 Date when the message is received by the server (input messages) or when is made available  
210 by the server (output messages)

#### 211 3.1.5

#### 212 **Message Type**

213 Type of the message payload as defined in IEC 62325 Part 451-n  
214 (Schedule\_MarketDocument, Acknowledgement\_MarketDocument, etc.)

#### 215 3.1.6

#### 216 **Message Code**

217 This number identifies a message in the server in a uniquely way. For a given message codes  
218 “x” and “y”, if “y” > “x” then “y” is a newer message. If “y” < “x” then “y” is an older message.  
219 Finally if “y” = “x”, then both messages are the same.

#### 220 3.1.7

#### 221 **Data Owner**

222 Person or entity that is responsible for the information contained in the message (payload).  
223 Usually corresponds with the sender\_MarketParticipant.mRID field in the IEC62325-451-n  
224 series.

#### 225 3.1.8

#### 226 **Data Provider**

227 Person or entity that is responsible for establishing a connection with the server and sending  
228 the message (payload).

#### 229 3.1.9

#### 230 **M/O/C**

231 Mandatory / Optional / Choice (choose one). Cn indicates “Choice n” and if several optional  
232 attributes have the same number “n” in Cn, it means all of them shall be present if this is the  
233 selected choice.

#### 234 3.1.10

#### 235 **Status**

236 Corresponds with the main reason code of the Acknowledgement message associated with a  
237 message as per IEC-62325-451-1. The status of messages without Acknowledgement  
238 (publication or incoming message still being processed) will be “OK”.

### 239 3.2 Namespaces

240 This Technical Specification uses these prefixes and namespaces:

241 a) msg (urn:iec62325.504:messages:1:0): The target namespace of the messages defined  
242 in this Technical Specification.

243 b) wss (urn:iec62325.504:wss:1:0): The WSDL target namespace for this Technical  
244 Specification.

245 This Technical Specification refers to these other prefixes and namespaces:

- 246 a) wsdl (<http://schemas.xmlsoap.org/wsdl/>): This contains the W3C WSDL 1.1 schema.
- 247 b) xs (<http://www.w3.org/2001/XMLSchema>): This contains the W3C XML Schema definition.
- 248 c) soap (<http://schemas.xmlsoap.org/wsdl/soap/>): This contains the W3C SOAP bindings for  
249 WSDL 1.1.
- 250 d) soap12 (<http://schemas.xmlsoap.org/wsdl/soap12/>): This contains the W3C SOAP bindings  
251 for WSDL 1.2
- 252 e) ds (<http://www.w3.org/2000/09/xmldsig#>): This contains the XML Digital Signature Schema  
253 definitions
- 254 f) xmime (<http://www.w3.org/2005/05/xmlmime>): This contains the W3C Media Content  
255 Description of Binary Data in XML
- 256 g) msg (<http://www.iec.ch/TC57/2008/schema/message>): This contains the IEC 61968-100  
257 schema definitions.

## 258 **4 Conformance**

### 259 **4.1 General**

260 This Clause specifies the conformance requirements for an application and a server to  
261 conform to this Technical Specification

### 262 **4.2 Client Application Conformance**

263 In order to conform to this Technical Specification a client application shall:

- 264 a) Support the following services:
  - 265 ○ List Messages and all of the mandatory aspects of this service as specified in  
266 Clause 5
  - 267 ○ Get Message and all of the mandatory aspects of this service as specified in  
268 Clause 5
  - 269 ○ Put Message and all of the mandatory aspects of this service as specified in  
270 Clause 5
- 271 b) Send and receive XML Instance documents according to the XML Schema specified in  
272 Clause 7 in this Technical Specification for the services listed in a).
- 273 c) Use the WSDL definitions, SOAP bindings, and operations specified in Clauses 8 and 9
- 274 d) Be able to access the server via HTTPS, using a client digital certificate recognized by the  
275 server for the purposes of establishing the https communication and creating the digital  
276 signature as specified in Clause 10

### 277 **4.3 Server Conformance**

278 In order to conform to this Technical Specification a server shall:

- 279 a) Support the following services:
  - 280 ○ List Messages and all of the mandatory aspects of this service as specified in  
281 Clause 5
  - 282 ○ Get Message service and all of the mandatory aspects of this service as  
283 specified in Clause 5
  - 284 ○ Put Message service and all of the mandatory aspects of this service as  
285 specified in Clause 5
- 286 b) Send and receive XML Instance documents according to the XML Schema specified in  
287 Clause 7 in this Technical Specification for the services listed in a)
- 288 c) Use the WSDL definitions, SOAP bindings, and operations specified in Clauses 8 and 9



289 d) Provide access to the server via HTTPS, and be able to assess that the client digital  
 290 certificate is valid and that the digital signature as specified in Clause 10 is correct.

291 **5 Service Definitions**

292 **5.1 List Messages**

293 **5.1.1 General**

294 The List Messages service is used to obtain a list of available messages for the client  
 295 according to a given filter (parameters).

296 The main filter shall be one of the following:

- 297 • application date of the returned messages
- 298 • Server Timestamp of the returned messages
- 299 • internal numerical code of the returned messages

300 Additional optional filters include:

- 301 • Message Identification
- 302 • Message Type
- 303 • Data Owner

304 The returned list of messages will comply with the main filter selected and also with all the  
 305 optional filters requested, and will include the following information related to each message:

- 306 • Internal numerical code representing the message in the server
- 307 • Message Identification
- 308 • Message Version
- 309 • Status
- 310 • Application Time Interval
- 311 • Server Timestamp
- 312 • Message Type
- 313 • Data Owner

314 **5.1.2 Service Request**

Parameter Name	Type	M/O/C	Description
StartTime	dateTime	C1	Specifies that the list of messages returned should only include messages whose end of their Application TimeInterval (Document TimeInterval) or Server Timestamp comes after the provided date.
EndTime	dateTime	C1	Specifies that the list of messages returned should only include messages whose start of their Application TimeInterval or ServerTimestamp (when the message was received or published in the server) comes before the provided date.
IntervalType	String	C1	Indicates whether the StartTime and EndTime refer to Application TimeInterval or to Server Timestamp. Permitted values: "Application" (default), "Server"
Code	number	C2	Specifies that the list of messages returned should only include messages with an internal identification number higher than the provided code. This means that the list will contain messages that are newer to the given one.  For optimization purposes, if this filter is used, only messages available since the

			00.00 of D-1 (day before) are guaranteed to be included in the response list
MessageIdentification	string	O	Specifies that the list of messages returned should only include messages whose Message Identification is compliant with the pattern provided in this parameter. (“*” can be used as a wildcard)
MsgType	string	O	Specifies that the list of messages returned should only include messages of the provided type.
Owner	string	O	Specifies that the list of messages returned should only include messages belonging to the provided Owner.

315

316 **5.1.3 Service Response**

317 If there is no message according to the provided filters, the service will return an empty list.  
 318 Otherwise, a list of message descriptors will be returned. Each message descriptor will  
 319 include the following parameters:

Parameter Name	Type	M/O/C	Description
Code	Position Type (number)	M	Specifies the internal identification number of the message
MessageIdentification	Identification Type (string)	M	Specifies the Message Identification. Messages defined in IEC-62325 Part 451-X series include this information. For additional messages not included in that standard, the server shall have a way of assigning a MessageIdentification to those messages.
MessageVersion	VersionType	O	Specifies the Message Version. Messages defined in IEC-62325 Part 451-X series include this information. For additional messages not included in that standard, the server should have a way of assigning a Message Version to those messages.
Status	String	O	Specifies the status of messages. Corresponds with the main reason code of the Acknowledgement message associated with this message as per IEC-62325-451-1. Possible values are: OK, FAILED. The status value “OK” corresponds with the IEC-62325-451-1 ReasonCode “A01”, and the status value “FAILED” corresponds with the rest of ReasonCodes.
ApplicationTimeInterval.Start	dateTime	M	Specifies the start of the message Application Time Interval. Messages defined in IEC-62325 Part 451-X series include this information. For additional messages not included in that standard, the server shall have a way of assigning an Application TimeInterval to those messages
ApplicationTimeInterval.End	dateTime	O	Specifies the end of the message Application Time Interval. Messages defined in IEC-62325 Part 451-X series include this information. For additional messages not included in that standard, the server shall have a way of assigning an Application TimeInterval to those messages.  When this information is missing, the message Application Time Interval is “from ApplicationTimeIntervalStart on” without an explicit end.

ServerTimestamp	MessageDateTime Type	M	Specifies the server timestamp (when the message was received or published in the server) of the message
Type	LongIdentification Type (string)	M	Specifies the Message Type.
Owner	LongIdentification Type (string)	M	Specifies the Data Owner of the message.

320

321 **5.1.4 Functional Requirements**

322 Confidentiality rules of the European Energy Market for Electricity shall be observed, thus the  
 323 list of messages available to a client will only include those messages to which he/she is  
 324 entitled (either completely or partially).

325 A client will be able to see all his previously submitted messages to the server, their  
 326 responses sent from the server (acknowledgements), and any publications that are available  
 327 to the client.

328 When the service is called with an invalid filter (e.g. malformed application dates) a Fault  
 329 message will be returned.

330 **5.2 Get Message**

331 **5.2.1 General**

332 The Get Messages service is used to obtain the message associated to the given parameter  
 333 (filter).

334 The filter shall be one of the following:

- 335 • Message Identification and Version
- 336 • Message code
- 337 • Queue indication

338 **5.2.2 Service Request**

Parameter Name	Type	M/O/C	Description
MessageIdentification	Identification Type (string)	C1	Specifies the Message Identification of the requested message.
MessageVersion	VersionType	C1	Specifies the Message Version of the requested message. If more than one message in the server have the same MessageIdentification and MessageVersion, the most recent one will be returned.
Code	Position Type (number)	C2	Specifies the internal identification number of the requested message
Queue	String	C3	Indicates that the server will decide which message will be returned. Its value shall be "NEXT".

339

340

341 **5.2.3 Service Response**

Parameter Name	Type	M/O/C	Description
[First child of Payload]	Any <sup>1</sup>	C1	The XML message that is being returned to the client.
BinaryContent	Base64Binary	C2	Optionally binary content may also be returned depending on the type of the requested message.
BinaryName	String	C2	Optionally, the name of the requested binary file.

342

343 **5.2.4 Functional Requirements**

344 Only one message will be retrieved for each Get Message service invocation.

345 A client will be able to retrieve all his previously submitted messages to the server, their  
 346 responses sent from the server (acknowledgements), and any publications that are available  
 347 to the client.

348 Servers are entitled to filter parts of the retrieved xml message for confidentiality reasons,  
 349 when that message, which is available for retrieval, includes information that should not be  
 350 available to the client.

351 If the retrieved message is a binary File, then the content is expressed as base 64 encoded  
 352 wrapped by the optional tag "BinaryContent".

353 When the service is called with an invalid message (e.g. missing or invalid code) a Fault  
 354 message will be returned.

355 If a user requests a message to which he/she is not entitled, the system will return a Fault  
 356 message as if he/she had requested a non-existing message.

357 The Queue parameter can be used when the server keeps an ordered list of messages for  
 358 each client to retrieve. A server not supporting this feature will return a fault message.

359 **5.3 Put Message**360 **5.3.1 General**

361 The Put Message service is used to send a message to the server for further processing  
 362 following the rules of the European Energy Markets for Electricity.

363 A series of standard XML messages related to the European Energy Market for Electricity are  
 364 defined in the IEC-62325 Part 451-X series, but this Technical Specification allows servers to  
 365 process additional XML messages not defined in said series.

366 Optionally, binary files may also be sent, if supported by the server.

367 **5.3.2 Service Request**

Parameter Name	Type	M/O/C	Description
[First child of Payload]	Any <sup>2</sup>	C1	The XML message that is being sent to the server.
BinaryContent	Base64Binary	C2	Optionally binary content may also be sent to the server.
BinaryName	String	C2	Optionally, the name of the binary file sent to the server.

368

369

---

<sup>1</sup> Any: any document with any namespace.

<sup>2</sup> Any: any document with any namespace.

370 **5.3.3 Service Response**

371 The response from the server will be in the form of an XML message indicating the technical  
 372 and/or functional acceptance or rejection of the message. For the messages described in the  
 373 IEC-62325 Part 451-X series, the response from the server should be an acknowledgement  
 374 message as defined in IEC-62325 Part 451-1.

375 **5.3.4 Functional Requirements**

376 For each XML message received the server needs to be able to identify each individual  
 377 message. The IEC-62325 Part 451-X series define such a way via the elements  
 378 DocumentIdentification and optionally DocumentVersion.

379 The server will perform simple validations:

- 380 • A user cannot send two (or more) messages with the same identification (e.g. Document  
 381 Identification and Version). However, another user could send messages with the same  
 382 identification that other user sent before.
- 383 • A user cannot send a message whose Version is lower than another message that has  
 384 been sent previously with the same Identification by the same user.

385 If there is not enough information in the sent message to create a proper response  
 386 (incomplete XML or missing tags), a Fault message will be issued instead.

387 If a user is not authorized to send a specific message, a Fault message shall be returned.

388 **5.4 Query Data**

389 **5.4.1 General**

390 The non-mandatory Query Data Service can be used by clients to request specific data from  
 391 the server using different query parameters.

392 The server does not need to have the response XML message ready before the service  
 393 invocation, and can create a specific message in response to each request as needed.

394 **5.4.2 Service Request**

Parameter Name	Type	M/O	Description
DataType	String	M	Indicates the type of data being requested.
StartTime	dateTime	O	Specifies that the returned message should only include data whose Application Date is after the provided date.
EndTime	dateTime	O	Specifies that the returned message should only include data whose Application Date is before the provided date.
Any <sup>3</sup>	String	O	Specifies additional parameters for the service.

395

396 **5.4.3 Service Response**

Parameter Name	Type	M/O/C	Description
QueryData	gop:QueryData	M	Wraps the XML message that is being returned to the client and the parameters used in the request.

397

398 **5.4.4 Functional Requirements**

399 Only one message will be retrieved for each Query Data service invocation.

---

<sup>3</sup> Any: Any other parameter indicated with the pairs "name" and "value" with the element Option in the message header.

400 When the service is called with invalid parameters) (e.g. malformed application dates) a Fault  
 401 message will be returned.

402 A list of basic datatypes to be supported if this service is implemented is shown below:

Data Type value	M/O	Description
"listOfDataTypes"	M	The server will return a list of valid DataTypes that can be used for this service on this server.
"serverTimestamp"	M	The server will return the Server Timestamp in UTC format
"parameterLimits"	O	<p>The server will return its operational limits for parameters used in the List, Get, Put services, such as maximum message size, maximum number of queried days in the list service, etc. If there is no a specific limit for a given parameter, the response will not include such limit value.</p> <p>If the user breaches one of the specified limits, the server will return a Fault message instead of the Response of the operation used.</p> <p>The following non comprehensive list of parameter names should be used to indicate such limits:</p> <ul style="list-style-type: none"> <li>• MaxNumMessagesInListResponse: Maximum number of messages that will be returned in a list operation response.</li> <li>• NumberOfDaysForLowCodeInListResponse: Number of days that are guarantee to be included in the response list when the request has used a small code value (typically 0). According to this specification, the default value for this parameter limit is 1 (all messages available from 00:00 of D-1).</li> <li>• MaxApplicationTimeIntervalInDaysInListRequest: Maximum number of days that a request for Application time interval type can span.</li> <li>• MaxServerTimeIntervalInDaysInListRequest: Maximum number of days that a request for Server time interval type can span.</li> <li>• MaxPayloadSizeInMBInPutRequest: Maximum size, in Megabytes, that message payload content can have. Messages with bigger size will be rejected.</li> <li>• MaxGetRequestPerMinute: Number of Get operations per minute that a user can execute.</li> <li>• MaxPutRequestPerMinute: Number of Put operations per minute that a user can execute.</li> <li>• MaxListRequestPerMinute: Number of List operations per minute that a user can execute.</li> <li>• MaxQueryRequestPerMinute: Number of Query Data operations per minute that a user can execute.</li> <li>• MaxMessageAgeInDays: Max number of days that a message will be accessible by this Technical Specification operations.</li> <li>• MaxDiffServerTimestampInSeconds: If set, the server will reject messages that do not meet the following criteria:  <math display="block">CT - ST + MD \geq 0</math>           Being CT the current server time, ST the msg:serverTimestamp indicated in the request message and MD this parameter value.</li> </ul> <p>Other parameters could be added.</p>

403

404 In order to ensure non-repudiation, the service response will include the parameters used in the  
 405 invocation along with the response.

406

407 **6 Applying IEC 61968-100**

408 **6.1 Integration Pattern**

409 **6.1.1 General**

410 The interactions between Client and Server described in this Technical Specification  
411 correspond with the IEC 61968-100 use case “Simple Request/Reply” supported by the “Basic  
412 Web Service Integration Pattern”.

413 This does not preclude the implementation of other Integration Patterns to support additional  
414 use cases such as “Request/Reply with ESB”, “Events” etc.

415 The server will expose in a WSDL a single operation “request” that will serve as the single  
416 entry point for the services described in this Technical Specification.

417 In the following diagrams the request/reply exchange between client and server is shown,  
418 including the type of message sent, the IEC 61968-100 verb and noun used, and the payload  
419 when applicable.

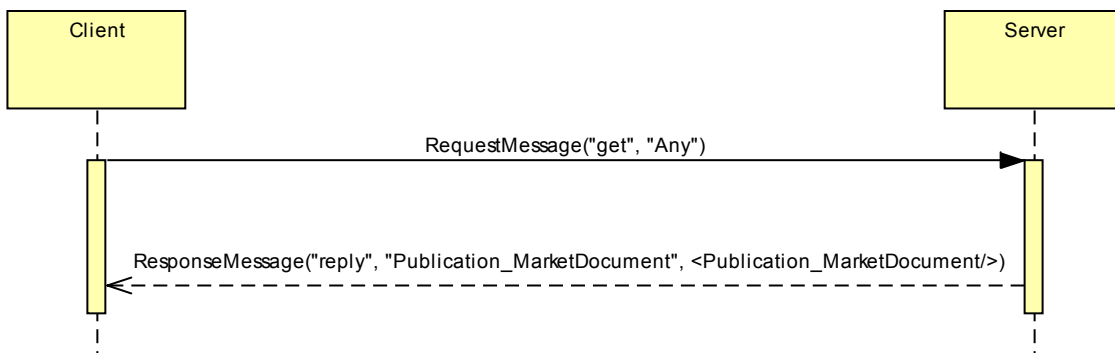
420 **6.1.2 List Service**



421

422 **6.1.3 Get Service**

423 In this example, a message of type “Publication\_MarketDocument” is requested:



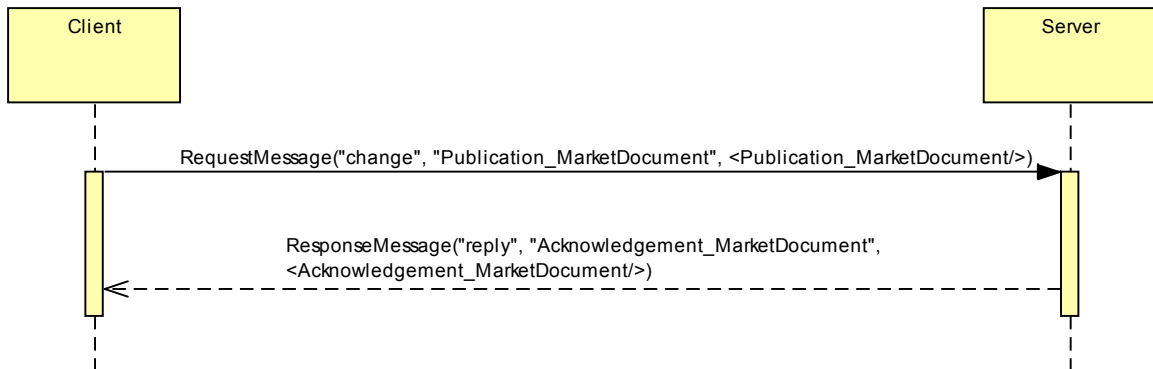
424

425

426

427 **6.1.4 Put Service**

428 In this example, a message of type “Publication\_MarketDocument” is sent to the server:



429

430



431 **6.2 Service Mapping**

432 **6.2.1 General**

433 The mapping between the different services described in this Technical Specification and the  
 434 IEC 61968-100 messages, including some relevant parameters, is shown in the following  
 435 table:

Service	61968-100 Stereotype Msg.	61968-100 Verb	61968-100 Noun	61968-100 Payload
List (Req.)	RequestMessage	get	MessageList	n/a
List (Resp)	ResponseMessage	reply	MessageList	gop:MessageList
Get (Req)	RequestMessage	get	Any <sup>4</sup>	n/a
Get (Resp)	ResponseMessage	reply	ChildOfPayload	xsd:any <sup>5</sup>
Put (Req.)	RequestMessage	create	ChildOfPayload	xsd:any <sup>6</sup>
Put (Resp)	ResponseMessage	reply	ChildOfPayload	xsd:any <sup>7</sup>
QueryData (Req.)	RequestMessage	get	QueryData	n/a
QueryData (Resp.)	ResponseMessage	reply	QueryData	gop:QueryData

436  
 437 Unless otherwise stated, all elements in the Header, Request or Response defined in  
 438 IEC61968-100 can be used.  
 439

440 **6.2.2 Header Values**

441 The “Source” element, if used, will be the EIC code of the Sender Entity for messages  
 442 following IEC 62325-451-n series.

443 It is strongly recommended to use the element “Timestamp” for all the interchanges. This  
 444 gives an added value to the signature allowing the recipient to reject messages with  
 445 timestamp set in the future. The QueryData “serverTimestamp” response uses this element to  
 446 indicate the required value.

447 **6.2.3 Request Values**

448 The parameters in the List Service Request, Get Service Request and Query Service Request  
 449 that do not correspond directly with a parameter in the IEC 61968-100 msg:RequestMessage  
 450 will be provided as Name/Value pairs under the msg:Option element of the msg:Request.

451 In the case of the Put service Request, when sending a binary file, the name of the file will be  
 452 included in the msg:Request.ID, with the attribute idType equal to “name”.

453 **6.2.4 Response Values**

454 In the case of the Put service Response, the msg:Reply.Result values “OK” and “FAILED”  
 455 correspond with the “Completely Accepted” and “Completely Rejected” responses described  
 456 in IEC-62325 Part 451-1.

457 In the case of the Get service Response, when requesting a binary file, the name of the file  
 458 will be included in the msg:Response.ID, with the attribute idType equal to “name”.

459 In the case of the Query service Response, all the attributes (filters) that were included in the  
 460 msg:Option element in the request will be included in the QueryData response.

---

<sup>4</sup> Any: Literal text “Any”.

<sup>5</sup> xsd:any: any document with any namespace.

<sup>6</sup> xsd:any: any document with any namespace.

<sup>7</sup> xsd:any: any document with any namespace.

## 461 6.2.5 Payload Values

462 The generic message payload container will be used (it is not mandatory to have strongly  
463 typed WSDLs)

464 The "BinaryContent" described in this Technical Specification will be mapped to the  
465 "msg:Compressed" element of the EIC 61968-100.

466 The "msg:Format" element will be absent or have the value "XML" if the payload is an XML,  
467 and "BINARY" if the payload is a binary document.

468

## 469 7 Schema Definitions

### 470 7.1 Common Definitions

```
471 <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
472 xmlns="urn:iec62325.504:messages:1:0"
473 targetNamespace="urn:iec62325.504:messages:1:0"
474 elementFormDefault="qualified" attributeFormDefault="unqualified">
475   <xsd:annotation>
476     <xsd:documentation>TF EDI EME - Common Types</xsd:documentation>
477   </xsd:annotation>
478   <xsd:complexType name="TimeIntervalType">
479     <xsd:sequence>
480       <xsd:element name="start" type="xsd:dateTime"/>
481       <xsd:element name="end" type="xsd:dateTime" minOccurs="0"/>
482     </xsd:sequence>
483   </xsd:complexType>
484   <xsd:simpleType name="StatusType">
485     <xsd:restriction base="xsd:string">
486       <xsd:enumeration value="OK"/>
487       <xsd:enumeration value="FAILED"/>
488     </xsd:restriction>
489   </xsd:simpleType>
490   <xsd:element name="Parameter">
491     <xsd:complexType>
492       <xsd:sequence>
493         <xsd:element name="name" type="xsd:string"/>
494         <xsd:element name="value" type="xsd:string" minOccurs="0"/>
495       </xsd:sequence>
496     </xsd:complexType>
497   </xsd:element>
498 </xsd:schema>
```

499

### 500 7.2 List Message

```
501 <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
502 xmlns="urn:iec62325.504:messages:1:0"
503 targetNamespace="urn:iec62325.504:messages:1:0"
504 elementFormDefault="qualified" attributeFormDefault="unqualified">
505   <xsd:annotation>
506     <xsd:documentation>TF EDI EME - List</xsd:documentation>
507   </xsd:annotation>
508   <xsd:element name="MessageList">
509     <xsd:annotation>
510       <xsd:documentation>List of messages</xsd:documentation>
511     </xsd:annotation>
512     <xsd:complexType>
513       <xsd:sequence>
```

```

514         <xsd:element name="Message" minOccurs="0"
515 maxOccurs="unbounded">
516             <xsd:complexType>
517                 <xsd:sequence>
518                     <xsd:element name="Code" type="xsd:positiveInteger"/>
519                     <xsd:element name="MessageIdentification"
520 type="xsd:string"/>
521                     <xsd:element name="MessageVersion"
522 type="xsd:positiveInteger" minOccurs="0"/>
523                     <xsd:element name="Status" type="StatusType"
524 minOccurs="0"/>
525                     <xsd:element name="ApplicationTimeInterval"
526 type="TimeIntervalType"/>
527                     <xsd:element name="ServerTimestamp"
528 type="xsd:dateTime"/>
529                     <xsd:element name="Type" type="xsd:string"/>
530                     <xsd:element name="Owner" type="xsd:string"/>
531                 </xsd:sequence>
532             </xsd:complexType>
533         </xsd:element>
534     </xsd:sequence>
535 </xsd:complexType>
536 </xsd:element>
537 </xsd:schema>
    
```

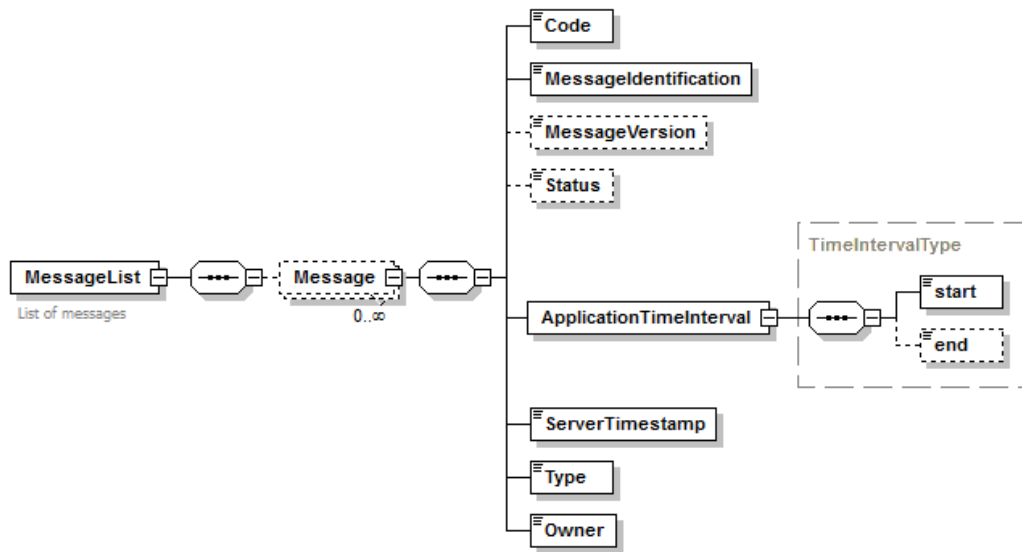


Figure 1 - MessageList schema structure

538  
539

540

541 **7.3 QueryData Message**

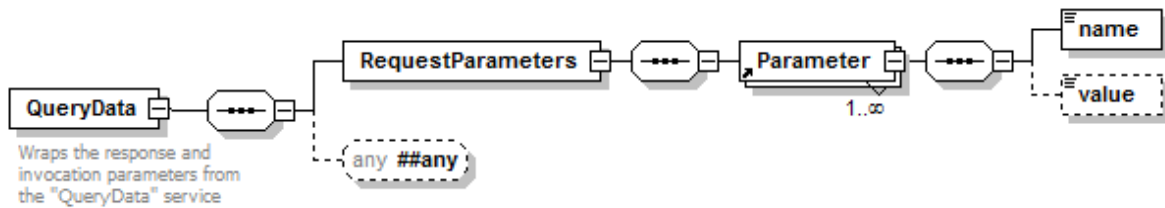
```

542 <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
543 xmlns="urn:iec62325.504:messages:1:0"
544 targetNamespace="urn:iec62325.504:messages:1:0"
545 elementFormDefault="qualified" attributeFormDefault="unqualified">
546 <xsd:annotation>
547     <xsd:documentation>TF EDI EME - QueryData</xsd:documentation>
548 </xsd:annotation>
549 <xsd:element name="QueryData">
550     <xsd:annotation>
551         <xsd:documentation>Wraps the response and request parameters from
552 the "QueryData" service</xsd:documentation>
553     </xsd:annotation>
554 <xsd:complexType>
    
```

```

555     <xsd:sequence>
556     <xsd:element name="RequestParameters">
557     <xsd:complexType>
558     <xsd:sequence>
559     <xsd:element ref="Parameter" maxOccurs="unbounded"/>
560     </xsd:sequence>
561     </xsd:complexType>
562     </xsd:element>
563     <xsd:any namespace="##any" processContents="lax" minOccurs="0"/>
564     </xsd:sequence>
565     </xsd:complexType>
566     </xsd:element>
567 </xsd:schema>
568

```



569  
570

Figure 2 - QueryData schema structure

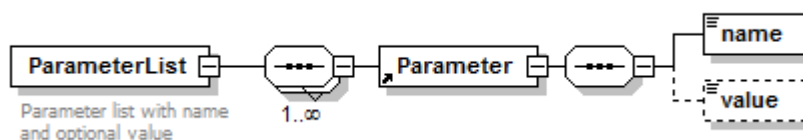
571

#### 572 7.4 QueryData List of Data Types

```

573 <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
574 xmlns="urn:iec62325.504:messages:1:0"
575 targetNamespace="urn:iec62325.504:messages:1:0"
576 elementFormDefault="qualified" attributeFormDefault="unqualified">
577   <xsd:annotation>
578     <xsd:documentation>TF EDI EME - QueryData List of
579 Parameters</xsd:documentation>
580   </xsd:annotation>
581   <xsd:element name="ParameterList">
582     <xsd:annotation>
583       <xsd:documentation>Parameter list with name and optional
584 value</xsd:documentation>
585     </xsd:annotation>
586     <xsd:complexType>
587       <xsd:sequence maxOccurs="unbounded">
588         <xsd:element ref="Parameter"/>
589       </xsd:sequence>
590     </xsd:complexType>
591   </xsd:element>
592 </xsd:schema>
593

```



594  
595

Figure 3 - ParameterList schema structure

## 596 8 Service Provider WSDL Abstract Definitions

597 By using synchronous web services, a client is immediately aware of the result of the service  
598 invocation.

599 This clause specifies the abstract WSDL definitions to support the services specified in this  
600 Technical Specification.

601

```
602 <definitions xmlns="http://schemas.xmlsoap.org/wsdl/"
603 xmlns:xs="http://www.w3.org/2001/XMLSchema"
604 xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
605 xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/"
606 xmlns:cmmsg="urn:iec62325.504:messages:1:0"
607 xmlns:wss="urn:iec62325.504:wss:1:0"
608 xmlns:msg="http://iec.ch/TC57/2011/schema/message"
609     targetNamespace="urn:iec62325.504:wss:1:0">
610     <types>
611         <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
612             <xs:import namespace="urn:iec62325.504:messages:1:0"
613 schemaLocation="./urn-iec62325-504-messages-1-0.xsd"/>
614             <xs:import
615 namespace="http://iec.ch/TC57/2011/schema/message" schemaLocation="./http-
616 iec-ch-TC57-2011-schema-message.xsd"/>
617         </xs:schema>
618     </types>
619     <message name="msgRequestMessage">
620         <part name="parameter" element="msg:RequestMessage"/>
621     </message>
622     <message name="msgResponseMessage">
623         <part name="parameter" element="msg:ResponseMessage"/>
624     </message>
625     <message name="msgFaultMsg">
626         <part name="msgFaultMessage" element="msg:FaultMessage"/>
627     </message>
628     <portType name="port_TFEDI_type">
629         <operation name="request">
630             <input message="wss:msgRequestMessage"/>
631             <output message="wss:msgResponseMessage"/>
632             <fault name="msgFaultMessage" message="wss:msgFaultMsg"/>
633         </operation>
634     </portType>
635 </definitions>
636
```

## 637 9 Service Provider WSDL SOAP Binding

638 This clause specifies the binding template of the abstract WSDL definitions in Clause 8 with  
639 SOAP Messages with the http(s) transport protocol. Although SOAP 1.2 is shown here, SOAP  
640 1.1 may also be used.

```
641 <definitions xmlns="http://schemas.xmlsoap.org/wsdl/"
642 xmlns:xs="http://www.w3.org/2001/XMLSchema"
643 xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
644 xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/"
645 xmlns:cmmsg="urn:iec62325.504:messages:1:0"
646 xmlns:wss="urn:iec62325.504:wss:1:0"
647 xmlns:msg="http://iec.ch/TC57/2011/schema/message"
648 targetNamespace="urn:iec62325.504:wss:1:0">
649     <binding name="binding_TFEDI" type="wss:port_TFEDI_type">
650         <soap12:binding style="document"
651 transport="http://schemas.xmlsoap.org/soap/http"/>
652         <operation name="request">
653             <soap12:operation soapActionRequired="false" style="document"/>
654             <input>
655                 <soap12:body use="literal"/>
656             </input>

```

```

657     <output>
658         <soap12:body use="literal"/>
659     </output>
660     <fault name="msgFaultMessage">
661         <soap12:fault name="msgFaultMessage" use="literal"/>
662     </fault>
663 </operation>
664 </binding>
665 <service name="ServiceEME">
666     <port name="Service_EME_Port" binding="wss:binding_TFEDI">
667         <soap12:address
668 location="https://example.com/WebService_EME/Service_EME"/>
669     </port>
670 </service>
671 </definitions>
672
673

```

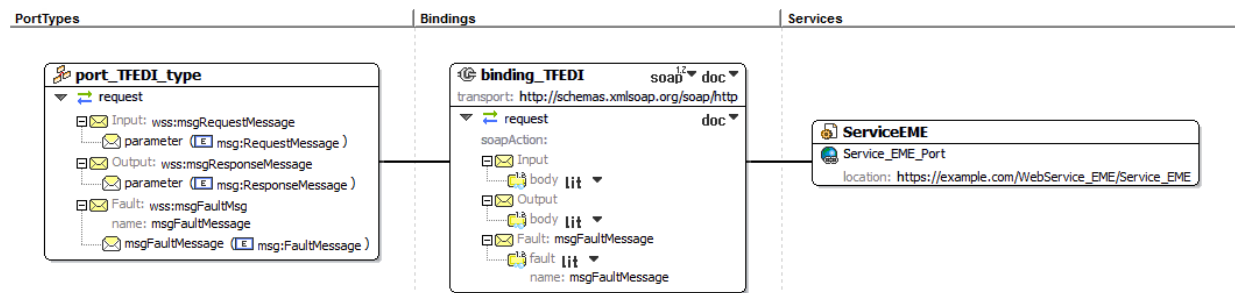


Figure 4 - WSDL structure

674  
675  
676  
677

678 Fault messages are used when the application produces an error condition that cannot be  
679 expressed with the operation response. Those errors are not usually related to business rules.

680 The fault message consists of two parts, a code and reason (also known as a fault string)  
681 and, optionally a detailed block that can be used to include more information about the error.

## 682 10 Security

683 The confidentiality of communications, and therefore the confidentiality of the information  
684 transmitted are achieved in the transport layer using https with client and server certificates  
685 (2-way SSL).

686 All the involved certificates must conform with the X.509 v3 specification (RFC 5280).

687  
688 Since this Technical Specification relies on HTTPS in order to cipher and authenticate  
689 communications, and as such, also relies on Transport Layer Security (TLS) for security  
690 services, special care must be taken when securing the transport layer. Thus, in order to  
691 conform to this Technical Specification, TLS clients and servers must use TLS1.1 or higher.  
692 Negotiating the use of SSL3 and/or TLS1.0 is not recommended, and its use should be  
693 explicitly agreed upon by the involved parties.

694 To ensure the authentication of the signer, the document integrity and non-repudiation, all IEC  
695 61968-100 messages with XML documents in Payloads for the "Get", "Put" and "QueryData"  
696 services are digitally signed either by the Data Owner or the Data Provider.

697 This digital signature, made with the private key associated with the user's digital certificate,  
698 follows the "XML Digital Signature" standard, and is transmitted in the signed message  
699 (enveloped signature). The signature is included in the IEC 61968-100 messages Header.

700 The signature applies to the complete IEC 61968-100 message that is being sent or retrieved,  
701 not to the XML Soap, nor exclusively to the single XML document in the Payload.

702 Thus the Reference URI shall always be "" in order to point to the whole IEC 61968-100  
703 document.

704

705 The CanonicalizationMethod will be one of:

- 706 • <http://www.w3.org/TR/2001/REC-xml-c14n-20010315>
- 707 • <http://www.w3.org/TR/2001/REC-xml-c14n-20010315#WithComments>
- 708 • <http://www.w3.org/2001/10/xml-exc-c14n>
- 709 • <http://www.w3.org/2001/10/xml-exc-c14n#WithComments>

710 The set of supported transforms (in this order) are:

- 711 • <http://www.w3.org/2000/09/xmlsig#enveloped-signature>
- 712 • (optional) any of the permitted canonicalization algorithms

713 In particular, no XPath or XSLT transforms are allowed.

714 The element "KeyInfo" includes the public key of the certificate used to create the signature.

715 In the case of messages with binary content, document signing is similar; since the  
716 "Compressed" element falls inside the IEC 61968-100 message that is being signed.

717

718

719  
720  
721  
722

## Annex A (normative)

### XML schema for common IEC 62325-504 messages

```

723 <?xml version="1.0" encoding="utf-8"?>
724 <!-- Common messages for IEC 62325-504 operations -->
725 <!-- Change Log -->
726 <!-- 2014/09/12 Update after CD review -->
727 <!-- 2013/06/12 First version -->
728 <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
729 xmlns="urn:iec62325.504:messages:1:0"
730 targetNamespace="urn:iec62325.504:messages:1:0" elementFormDefault="qualified"
731 attributeFormDefault="unqualified">
732 <!-- ----- -->
733 <xsd:annotation>
734 <xsd:documentation>TF EDI EME - Common Types</xsd:documentation>
735 </xsd:annotation>
736 <xsd:complexType name="TimeIntervalType">
737 <xsd:sequence>
738 <xsd:element name="start" type="xsd:dateTime"/>
739 <xsd:element name="end" type="xsd:dateTime" minOccurs="0"/>
740 </xsd:sequence>
741 </xsd:complexType>
742 <xsd:simpleType name="StatusType">
743 <xsd:restriction base="xsd:string">
744 <xsd:enumeration value="OK"/>
745 <xsd:enumeration value="FAILED"/>
746 </xsd:restriction>
747 </xsd:simpleType>
748 <xsd:element name="Parameter">
749 <xsd:complexType>
750 <xsd:sequence>
751 <xsd:element name="name" type="xsd:string"/>
752 <xsd:element name="value" type="xsd:string" minOccurs="0"/>
753 </xsd:sequence>
754 </xsd:complexType>
755 </xsd:element>
756 <!-- ----- -->
757 <xsd:annotation>
758 <xsd:documentation>TF EDI EME - List</xsd:documentation>
759 </xsd:annotation>
760 <xsd:element name="MessageList">
761 <xsd:annotation>
762 <xsd:documentation>List of messages</xsd:documentation>
763 </xsd:annotation>
764 <xsd:complexType>
765 <xsd:sequence>
766 <xsd:element name="Message" minOccurs="0" maxOccurs="unbounded">
767 <xsd:complexType>
768 <xsd:sequence>
769 <xsd:element name="Code" type="xsd:positiveInteger"/>
770 <xsd:element name="MessageIdentification" type="xsd:string"/>
771 <xsd:element name="MessageVersion" type="xsd:positiveInteger"
772 minOccurs="0"/>
773 <xsd:element name="Status" type="StatusType" minOccurs="0"/>
774 <xsd:element name="ApplicationTimeInterval"
775 type="TimeIntervalType"/>
776 <xsd:element name="ServerTimestamp" type="xsd:dateTime"/>
777 <xsd:element name="Type" type="xsd:string"/>
778 <xsd:element name="Owner" type="xsd:string"/>
779 </xsd:sequence>
780 </xsd:complexType>
781 </xsd:element>
782 </xsd:sequence>
783 </xsd:complexType>
784 </xsd:element>

```



```

785 <!-- _____ -->
786 <xsd:annotation>
787   <xsd:documentation>TF EDI EME - QueryData</xsd:documentation>
788 </xsd:annotation>
789 <xsd:element name="QueryData">
790   <xsd:annotation>
791     <xsd:documentation>Wraps the response and request parameters from the
792 "QueryData" service</xsd:documentation>
793   </xsd:annotation>
794   <xsd:complexType>
795     <xsd:sequence>
796       <xsd:element name="RequestParameters">
797         <xsd:complexType>
798           <xsd:sequence>
799             <xsd:element ref="Parameter" maxOccurs="unbounded"/>
800           </xsd:sequence>
801         </xsd:complexType>
802       </xsd:element>
803       <xsd:any namespace="##any" processContents="lax" minOccurs="0"/>
804     </xsd:sequence>
805   </xsd:complexType>
806 </xsd:element>
807 <!-- _____ -->
808 <xsd:annotation>
809   <xsd:documentation>TF EDI EME - QueryData List of
810 Parameters</xsd:documentation>
811 </xsd:annotation>
812 <xsd:element name="ParameterList">
813   <xsd:annotation>
814     <xsd:documentation>Parameter list with name and optional
815 value</xsd:documentation>
816   </xsd:annotation>
817   <xsd:complexType>
818     <xsd:sequence maxOccurs="unbounded">
819       <xsd:element ref="Parameter"/>
820     </xsd:sequence>
821   </xsd:complexType>
822 </xsd:element>
823 </xsd:schema>

```

824  
825  
826  
827

## Annex B (Informative)

### Message Examples

#### 828 B.1 List

##### 829 B.1.1 Basic Example

##### 830 B.1.1.1 Request:

```

831 <soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
832   <soap:Header/>
833   <soap:Body>
834     <msg:RequestMessage xmlns:msg="http://iec.ch/TC57/2011/schema/message">
835       <msg:Header>
836         <msg:Verb>get</msg:Verb>
837         <msg:Noun>MessageList</msg:Noun>
838         <msg:Context>PRODUCTION</msg:Context>
839         <msg:Timestamp>2012-11-30T09:30:47.581Z</msg:Timestamp>
840         <msg:AckRequired>true</msg:AckRequired>
841       </msg:Header>
842       <msg:Request>
843         <msg:StartTime>2012-11-26T23:00:00Z</msg:StartTime>
844         <msg:EndTime>2012-11-27T23:00:00Z</msg:EndTime>
845         <msg:Option>
846           <msg:name>IntervalType</msg:name>
847           <msg:value>Server</msg:value>
848         </msg:Option>
849         <msg:Option>
850           <msg:name>Owner</msg:name>
851           <msg:value>10EXAMPLE-EIC-P</msg:value>
852         </msg:Option>
853       </msg:Request>
854     </msg:RequestMessage>
855   </soap:Body>
856 </soap:Envelope>

```

857

##### 858 B.1.1.2 Response:

```

859 <soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
860   <soap:Header/>
861   <soap:Body>
862     <msg:ResponseMessage xmlns:msg="http://iec.ch/TC57/2011/schema/message">
863       <msg:Header>
864         <msg:Verb>reply</msg:Verb>
865         <msg:Noun>MessageList</msg:Noun>
866         <msg:Context>PRODUCTION</msg:Context>
867         <msg:Timestamp>2012-11-30T09:30:48.321Z</msg:Timestamp>
868       </msg:Header>
869       <msg:Reply>
870         <msg:Result>OK</msg:Result>
871       </msg:Reply>
872       <msg:Payload>
873         <urn:MessageList xmlns:urn="urn:iec62325.504:messages:1:0">
874           <urn:Message>
875             <urn:Code>6465</urn:Code>
876             <urn:MessageIdentification>NTCForecast_20140917</urn:MessageIdentification>
877             <urn:MessageVersion>1</urn:MessageVersion>
878             <urn:Status>FAILED</urn:Status>
879             <urn:ApplicationTimeInterval>
880               <urn:start>2014-09-16T22:00:00Z</urn:start>
881             </urn:ApplicationTimeInterval>
882             <urn:ServerTimestamp>2014-09-16T09:00:06Z</urn:ServerTimestamp>
883             <urn:Type>Publication_MarketDocument</urn:Type>
884             <urn:Owner>10EXAMPLE-EIC-P</urn:Owner>

```

```

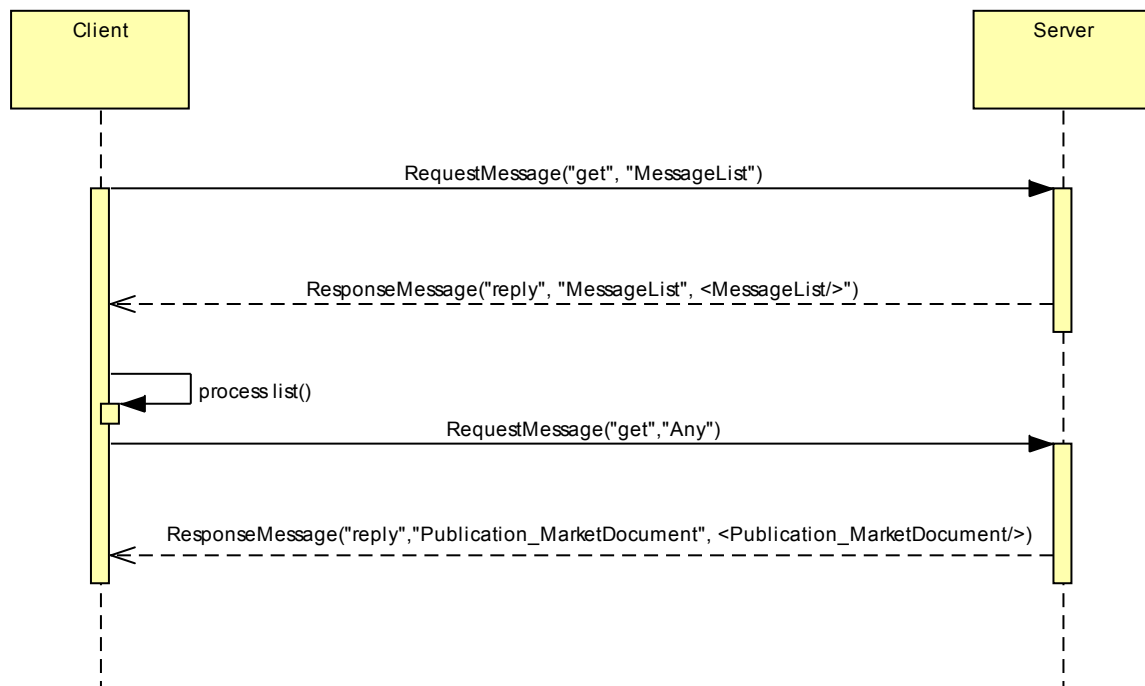
885     </urn:Message>
886     <urn:Message>
887         <urn:Code>6466</urn:Code>
888     <urn:MessageIdentification>AckNTCForecast_20140917</urn:MessageIdentificati
889 on>
890         <urn:MessageVersion>1</urn:MessageVersion>
891         <urn:Status>FAILED</urn:Status>
892         <urn:ApplicationTimeInterval>
893             <urn:start>2014-09-16T22:00:00Z</urn:start>
894         </urn:ApplicationTimeInterval>
895         <urn:ServerTimestamp>2014-09-16T09:00:06Z</urn:ServerTimestamp>
896         <urn:Type>Acknowledgement_MarketDocument</urn:Type>
897         <urn:Owner>10XEXAMPLE-EIC-P</urn:Owner>
898     </urn:Message>
899     <urn:Message>
900         <urn:Code>6467</urn:Code>
901     <urn:MessageIdentification>PhysicalFlows_2014091611</urn:MessageIdentificat
902 ion>
903         <urn:MessageVersion>1</urn:MessageVersion>
904         <urn:Status>OK</urn:Status>
905         <urn:ApplicationTimeInterval>
906             <urn:start>2014-09-16T08:00:00Z</urn:start>
907             <urn:end>2014-09-16T09:00:00Z</urn:end>
908         </urn:ApplicationTimeInterval>
909         <urn:ServerTimestamp>2014-09-16T09:13:04Z</urn:ServerTimestamp>
910         <urn:Type>Publication_MarketDocument</urn:Type>
911         <urn:Owner>10XEXAMPLE-EIC-P</urn:Owner>
912     </urn:Message>
913     <urn:Message>
914         <urn:Code>6468</urn:Code>
915     <urn:MessageIdentification>AckPhysicalFlows_2014091611</urn:MessageIdentifi
916 cation>
917         <urn:MessageVersion>1</urn:MessageVersion>
918         <urn:Status>OK</urn:Status>
919         <urn:ApplicationTimeInterval>
920             <urn:start>2014-09-16T08:00:00Z</urn:start>
921             <urn:end>2014-09-16T09:00:00Z</urn:end>
922         </urn:ApplicationTimeInterval>
923         <urn:ServerTimestamp>2014-09-16T09:13:04Z</urn:ServerTimestamp>
924         <urn:Type>Acknowledgement_MarketDocument</urn:Type>
925         <urn:Owner>10XEXAMPLE-EIC-P</urn:Owner>
926     </urn:Message>
927 </urn:MessageList>
928 </msg:Payload>
929 </msg:ResponseMessage>
930 </soap:Body>
931 </soap:Envelope>
932

```

933 **B.2 Get**

934 As a general rule, before retrieving a Message, the client should find out if the message  
935 exists.

936 Thus, the recommended sequence diagram is the following figure, where the actor requests  
937 the "List" service in first place, recovering the list of available messages. Once the actor has  
938 such information, he will invoke the "Get" service for a message that appears in the retrieved  
939 list.



940

941

## 942 B.2.1 Basic Example

### 943 B.2.1.1 Request by code:

```

944 <soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
945   <soap:Header/>
946   <soap:Body>
947     <msg:RequestMessage xmlns:msg="http://iec.ch/TC57/2011/schema/message">
948       <msg:Header>
949         <msg:Verb>get</msg:Verb>
950         <msg:Noun>Any</msg:Noun>
951         <msg:Context>PRODUCTION</msg:Context>
952         <msg:Timestamp>2012-11-30T09:30:47.581Z</msg:Timestamp>
953         <msg:AckRequired>true</msg:AckRequired>
954       </msg:Header>
955       <msg:Request>
956         <msg:Option>
957           <msg:name>Code</msg:name>
958           <msg:value>879021</msg:value>
959         </msg:Option>
960       </msg:Request>
961     </msg:RequestMessage>
962   </soap:Body>
963 </soap:Envelope>
  
```

964

### 965 B.2.1.2 Request by queue:

```

966 <soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
967   <soap:Header/>
968   <soap:Body>
969     <msg:RequestMessage xmlns:msg="http://iec.ch/TC57/2011/schema/message">
970       <msg:Header>
971         <msg:Verb>get</msg:Verb>
972         <msg:Noun>Any</msg:Noun>
973         <msg:Context>PRODUCTION</msg:Context>
974         <msg:Timestamp>2012-11-30T09:30:47.581Z</msg:Timestamp>
975         <msg:AckRequired>true</msg:AckRequired>
976       </msg:Header>
  
```

```

977         <msg:Request>
978             <msg:Option>
979                 <msg:name>Queue</msg:name>
980                 <msg:value>NEXT</msg:value>
981             </msg:Option>
982         </msg:Request>
983     </msg:RequestMessage>
984 </soap:Body>
985 </soap:Envelope>

```

987 **B.2.1.3 Request by identification and version:**

```

988 <soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
989     <soap:Header/>
990     <soap:Body>
991         <msg:RequestMessage xmlns:msg="http://iec.ch/TC57/2011/schema/message">
992             <msg:Header>
993                 <msg:Verb>get</msg:Verb>
994                 <msg:Noun>Any</msg:Noun>
995                 <msg:Context>PRODUCTION</msg:Context>
996                 <msg:Timestamp>2012-11-30T09:30:47.581Z</msg:Timestamp>
997                 <msg:AckRequired>true</msg:AckRequired>
998             </msg:Header>
999             <msg:Request>
1000                 <msg:Option>
1001                     <msg:name>MessageIdentification</msg:name>
1002                     <msg:value>Schedule_D_20140416</msg:value>
1003                 </msg:Option>
1004                 <msg:Option>
1005                     <msg:name>MessageVersion</msg:name>
1006                     <msg:value>1</msg:value>
1007                 </msg:Option>
1008             </msg:Request>
1009         </msg:RequestMessage>
1010     </soap:Body>
1011 </soap:Envelope>

```

1012

1013 **B.2.1.4 Response:**

```

1014 <soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
1015     <soap:Header>
1016     </soap:Header>
1017     <soap:Body>
1018         <msg:ResponseMessage
1019     xmlns:msg="http://iec.ch/TC57/2011/schema/message">
1020             <msg:Header>
1021                 <msg:Verb>reply</msg:Verb>
1022                 <msg:Noun>Schedule_MarketDocument</msg:Noun>
1023                 <msg:Context>PRODUCTION</msg:Context>
1024                 <msg:Timestamp>2012-11-30T09:30:48.397Z</msg:Timestamp>
1025                 <Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
1026                     <SignedInfo>
1027                         <CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-
1028     xml-c14n-20010315"/>
1029                         <SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-
1030     sha1"/>
1031                         <Reference URI="">
1032                             <Transforms>
1033                                 <Transform
1034     Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"/>
1035                                 <Transform Algorithm="http://www.w3.org/TR/2001/REC-xml-
1036     c14n-20010315"/>
1037                             </Transforms>
1038                             <DigestMethod
1039     Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
1040                                 <DigestValue>xSnE7qGXJkWGG11rq3ze5DYGxHw=</DigestValue>
1041                             </Reference>

```

```

1042     </SignedInfo>
1043 <SignatureValue>QLk/r49g4uoR8pXT9WX3fg6QYEh/r0.....==</SignatureValue>
1044     <KeyInfo>
1045         <X509Data>
1046             <X509IssuerSerial>
1047                 <X509IssuerName>CN=testCA, O=test Domain</X509IssuerName>
1048                 <X509SerialNumber>238</X509SerialNumber>
1049             </X509IssuerSerial>
1050             <X509SubjectName>UID=myname, OU=intern, OU=myou, OU=users,
1051 O=test, C=test</X509SubjectName>
1052             <X509Certificate>MIID2TCCAsGgAwIBA .... =</X509Certificate>
1053         </X509Data>
1054     </KeyInfo>
1055 </Signature>
1056 </msg:Header>
1057 <msg:Reply>
1058     <msg:Result>OK</msg:Result>
1059 </msg:Reply>
1060 <msg:Payload>
1061 <Schedule_MarketDocument xmlns="urn:iec62325.351:tc57wg16:451-
1062 2:scheduledocument:5:0">
1063     <mRID>Schedule_D_20140416</mRID>
1064     <revisionNumber>1</revisionNumber>
1065     <type>A04</type>
1066     <process.processType>A01</process.processType>
1067     <process.classificationType>A01</process.classificationType>
1068     <sender_MarketParticipant.mRID codingScheme="A01">10EXAMPLE-EIC-P
1069 </sender_MarketParticipant.mRID>
1070     <sender_MarketParticipant.marketRole.type>A04
1071 </sender_MarketParticipant.marketRole.type>
1072     <receiver_MarketParticipant.mRID codingScheme="A01">10EXAMPLE-EIC-P
1073 </receiver_MarketParticipant.mRID>
1074 <receiver_MarketParticipant.marketRole.type>A04</receiver_MarketParticipant.ma
1075 rketRole.type>
1076     <createdDateTime>2014-04-15T13:06:29Z</createdDateTime>
1077     <schedule_Time_Period.timeInterval>
1078         <start>2014-04-15T22:00Z</start>
1079         <end>2014-04-16T22:00Z</end>
1080     </schedule_Time_Period.timeInterval>
1081     ...
1082 </Schedule_MarketDocument>
1083 </msg:Payload>
1084 </msg:ResponseMessage>
1085 </soap:Body>
1086 </soap:Envelope>
1087

```

## 1088 B.3 Put

### 1089 B.3.1 Basic Example

#### 1090 B.3.1.1 Request:

```

1091 <soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
1092     <soap:Header/>
1093     <soap:Body>
1094         <msg:RequestMessage xmlns:msg="http://iec.ch/TC57/2011/schema/message">
1095             <msg:Header>
1096                 <msg:Verb>create</msg:Verb>
1097                 <msg:Noun>Schedule_MarketDocument</msg:Noun>
1098                 <msg:Context>PRODUCTION</msg:Context>
1099                 <msg:Timestamp>2014-04-15T13:06:29.885Z</msg:Timestamp>
1100                 <msg:AckRequired>true</msg:AckRequired>
1101             <Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
1102                 ...
1103             </Signature>
1104         </msg:Header>
1105     </soap:Body>

```

```

1106 <Schedule_MarketDocument xmlns="urn:iec62325.351:tc57wg16:451-
1107 2:scheduledocument:5:0">
1108   <mRID>Schedule_D_20140416</mRID>
1109   <revisionNumber>1</revisionNumber>
1110   <type>A04</type>
1111   <process.processType>A01</process.processType>
1112   <process.classificationType>A01</process.classificationType>
1113   <sender_MarketParticipant.mRID codingScheme="A01">10XEXAMPLE-EIC-P
1114 </sender_MarketParticipant.mRID>
1115   <sender_MarketParticipant.marketRole.type>A04
1116 </sender_MarketParticipant.marketRole.type>
1117   <receiver_MarketParticipant.mRID codingScheme="A01">10XEXAMPLE-EIC-P
1118 </receiver_MarketParticipant.mRID>
1119   <receiver_MarketParticipant.marketRole.type>A04
1120 </receiver_MarketParticipant.marketRole.type>
1121   <createdDateTime>2014-04-15T13:06:29Z</createdDateTime>
1122   ...
1123 </Schedule_MarketDocument>
1124 </msg:Payload>
1125 </msg:RequestMessage>
1126 </soap:Body>
1127 </soap:Envelope>

```

1128

### 1129 B.3.1.2 Response:

```

1130 <soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
1131   <soap:Header/>
1132   <soap:Body>
1133     <msg:ResponseMessage xmlns:msg="http://iec.ch/TC57/2011/schema/message">
1134       <msg:Header>
1135         <msg:Verb>reply</msg:Verb>
1136         <msg:Noun>Acknowledgement_MarketDocument</msg:Noun>
1137         <msg:Context>PRODUCTION</msg:Context>
1138         <msg:Timestamp>2014-04-15T13:07:31.380Z</msg:Timestamp>
1139         <Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
1140           ...
1141         </Signature>
1142       </msg:Header>
1143       <msg:Reply>
1144         <msg:Result>OK</msg:Result>
1145       </msg:Reply>
1146       <msg:Payload>
1147         <Acknowledgement_MarketDocument xmlns="urn:iec62325.351:tc57wg16:451-
1148 1:acknowledgementdocument:6:0">
1149           <mRID>ACK_Schedule_D_20140416</mRID>
1150           <createdDateTime>2014-04-15T13:07:30Z</createdDateTime>
1151           <sender_MarketParticipant.mRID codingScheme="A01">10XEXAMPLE-EIC-P
1152 </sender_MarketParticipant.mRID>
1153           <sender_MarketParticipant.marketRole.type>A04
1154 </sender_MarketParticipant.marketRole.type>
1155           <receiver_MarketParticipant.mRID codingScheme="A01">10XEXAMPLE-EIC-P
1156 </receiver_MarketParticipant.mRID>
1157           <receiver_MarketParticipant.marketRole.type>A04
1158 </receiver_MarketParticipant.marketRole.type>
1159           <received_MarketDocument.mRID>Schedule_D_20140416
1160 </received_MarketDocument.mRID>
1161           <received_MarketDocument.revisionNumber>1
1162 </received_MarketDocument.revisionNumber>
1163           <received_MarketDocument.type>A51</received_MarketDocument.type>
1164           <received_MarketDocument.createdDateTime>2014-04-15T13:06:29Z
1165 </received_MarketDocument.createdDateTime>
1166           <Reason>
1167             <code>A01</code>
1168           </Reason>
1169         </Acknowledgement_MarketDocument>
1170       </msg:Payload>
1171     </msg:ResponseMessage>
1172   </soap:Body>
1173 </soap:Envelope>

```

1174

1175 **B.3.2 Example with binary data**

1176 If the message has a binary part, this can be transmitted as 64b encoded form:

```

1177 <soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
1178   <soap:Header/>
1179   <soap:Body>
1180     <msg:RequestMessage xmlns:msg="http://iec.ch/TC57/2011/schema/message">
1181       <msg:Header>
1182         <msg:Verb>create</msg:Verb>
1183         <msg:Noun>Schedule_MarketDocument_jpg</msg:Noun>
1184         <msg:Context>PRODUCTION</msg:Context>
1185         <msg:Timestamp>2012-11-30T09:33:47.658Z</msg:Timestamp>
1186         <msg:AckRequired>true</msg:AckRequired>
1187       <Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
1188         ...
1189       </Signature>
1190     </msg:Header>
1191     <msg:Request>
1192       <msg:ID idType="name">schedule_xyz.jpg</msg:ID>
1193     </msg:Request>
1194     <msg:Payload>
1195       <msg:Compressed>
1196 /9j/4AAQSkZJRgABAgEASABIAAD/wAALCAAGABcBAREA/9sAhAAGBAUGBQQGBgUGBwcGCAoRCwoJ
1197 CQoVDxAMERkWGhoYFhgXGx8oIRsdJR4XGCIvIyUpKiwTLBshMTQwKzQoKywrAQcHBwoJChQLCxQr
1198 HBgcHCsrKysrKysrKysrKysrKysrKysrKysrKysrKysrKysrKysrKysrKysrKysrKysrKysrKysrKyv/xADS
1199 AAABBQEBAQEBAQAAAAAAAAAAQIDBAUGBwgJCgsQAAIBAwMCAmFBQOEAAABfQECAwAEEQUSITFB
1200 BhNRYQcicRQyZGhCCNCscEVUtHwJDNicoIJChYXGBkaJSYnKCkqNDU2Nzg5OkNERUZHSElKU1RV
1201 VldYWVpjZGVmZ2hpanN0dXZ3eH16g4SFhoeIiYqSk5SVlpeYmZqio6Slppeqoqaqys7S1tre4ubrC
1202 w8TFxsfIycrS09TV1tfY2drh4uPk5ebn6Onq8fLz9PX29/j5+v/aAAgBAQAAPwDtfA+l6zHpXgv+
1203 y9euZdSu/A11/Z/22OHyrKURy7NuyIEqGK537zhR15zx8vidPFxwakvIb++vYIdQ1OCM3kLRvEBo
1204 10xUF55ncZYtueQn5iAAqK0Pixffb9F+JGla3qurR2+meX/AGs8SfaFPnXcMlh9niZ1VdkAZJMG
1205 PLNk+YRur//Z</msg:Compressed>
1206       <msg:Format>BINARY</msg:Format>
1207     </msg:Payload>
1208   </msg:RequestMessage>
1209 </soap:Body>
1210 </soap:Envelope>

```

1211

1212 **B.4 Query Data**1213 **B.4.1 List of Data Types Example**1214 **B.4.1.1 Request:**

```

1215 <soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
1216   <soap:Header/>
1217   <soap:Body>
1218     <msg:RequestMessage xmlns:msg="http://iec.ch/TC57/2011/schema/message">
1219       <msg:Header>
1220         <msg:Verb>get</msg:Verb>
1221         <msg:Noun>QueryData</msg:Noun>
1222         <msg:Context>PRODUCTION</msg:Context>
1223         <msg:Timestamp>2012-11-30T09:30:47.581Z</msg:Timestamp>
1224       </msg:Header>
1225       <msg:Request>
1226         <msg:Option>
1227           <msg:name>DataType</msg:name>
1228           <msg:value>listOfDataTypes</msg:value>
1229         </msg:Option>
1230       </msg:Request>
1231     </msg:RequestMessage>
1232   </soap:Body>
1233 </soap:Envelope>

```



1234

1235 **B.4.1.2 Response:**

```

1236 <soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
1237   <soap:Header/>
1238   <soap:Body>
1239     <msg:ResponseMessage xmlns:msg="http://iec.ch/TC57/2011/schema/message">
1240       <msg:Header>
1241         <msg:Verb>reply</msg:Verb>
1242         <msg:Noun>QueryData</msg:Noun>
1243         <msg:Context>PRODUCTION</msg:Context>
1244         <msg:Timestamp>2012-11-30T09:30:48.004Z</msg:Timestamp>
1245       <Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
1246         ...
1247       </Signature>
1248     </msg:Header>
1249     <msg:Reply>
1250       <msg:Result>OK</msg:Result>
1251     </msg:Reply>
1252     <msg:Payload>
1253       <QueryData xmlns="urn:iec62325.504:messages:1:0">
1254         <RequestParameters>
1255           <Parameter>
1256             <name>DataType</name>
1257             <value>listOfDataTypes</value>
1258           </Parameter>
1259         </RequestParameters>
1260         <ParameterList xmlns="urn:iec62325.504:messages:1:0">
1261           <Parameter>
1262             <name>listOfDataTypes</name>
1263           </Parameter>
1264           <Parameter>
1265             <name>serverTimestamp</name>
1266           </Parameter>
1267           <Parameter>
1268             <name>parameterLimits</name>
1269           </Parameter>
1270         </ParameterList>
1271       </QueryData>
1272     </msg:Payload>
1273   </msg:ResponseMessage>
1274 </soap:Body>
1275 </soap:Envelope>

```

1276

1277 **B.4.2 Server Timestamp Request Example**1278 **B.4.2.1 Request:**

```

1279 <soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
1280   <soap:Header/>
1281   <soap:Body>
1282     <msg:RequestMessage xmlns:msg="http://iec.ch/TC57/2011/schema/message">
1283       <msg:Header>
1284         <msg:Verb>get</msg:Verb>
1285         <msg:Noun>QueryData</msg:Noun>
1286         <msg:Context>PRODUCTION</msg:Context>
1287         <msg:Timestamp>2012-11-30T09:30:47.581Z</msg:Timestamp>
1288       </msg:Header>
1289       <msg:Request>
1290         <msg:Option>
1291           <msg:name>DataType</msg:name>
1292           <msg:value>serverTimestamp</msg:value>
1293         </msg:Option>
1294       </msg:Request>
1295     </msg:RequestMessage>
1296   </soap:Body>
1297 </soap:Envelope>

```

1298

1299 **B.4.2.2 Response:**

```

1300 <soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
1301   <soap:Header/>
1302   <soap:Body>
1303     <msg:ResponseMessage xmlns:msg="http://iec.ch/TC57/2011/schema/message">
1304       <msg:Header>
1305         <msg:Verb>reply</msg:Verb>
1306         <msg:Noun>QueryData</msg:Noun>
1307         <msg:Context>PRODUCTION</msg:Context>
1308         <msg:Timestamp>2012-11-30T09:30:48.004Z</msg:Timestamp>
1309       <Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
1310         ...
1311       </Signature>
1312     </msg:Header>
1313     <msg:Reply>
1314       <msg:Result>OK</msg:Result>
1315     </msg:Reply>
1316     <msg:Payload>
1317       <QueryData xmlns="urn:iec62325.504:messages:1:0">
1318         <RequestParameters>
1319           <Parameter>
1320             <name>DataType</name>
1321             <value>serverTimestamp</value>
1322           </Parameter>
1323         </RequestParameters>
1324       </QueryData>
1325     </msg:Payload>
1326   </msg:ResponseMessage>
1327 </soap:Body>
1328 </soap:Envelope>

```

1329

1330 **B.4.3 Server Parameter Limits Request Example**1331 **B.4.3.1 Request:**

```

1332 <soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
1333   <soap:Header/>
1334   <soap:Body>
1335     <msg:RequestMessage xmlns:msg="http://iec.ch/TC57/2011/schema/message">
1336       <msg:Header>
1337         <msg:Verb>get</msg:Verb>
1338         <msg:Noun>QueryData</msg:Noun>
1339         <msg:Context>PRODUCTION</msg:Context>
1340         <msg:Timestamp>2012-11-30T09:30:47.581Z</msg:Timestamp>
1341       </msg:Header>
1342       <msg:Request>
1343         <msg:Option>
1344           <msg:name>DataType</msg:name>
1345           <msg:value>parameterLimits</msg:value>
1346         </msg:Option>
1347       </msg:Request>
1348     </msg:RequestMessage>
1349   </soap:Body>
1350 </soap:Envelope>

```

1351

1352

1353

1354 **B.4.3.2 Response:**

```

1355 <soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
1356   <soap:Header/>
1357   <soap:Body>

```

```

1358 <msg:ResponseMessage xmlns:msg="http://iec.ch/TC57/2011/schema/message">
1359   <msg:Header>
1360     <msg:Verb>reply</msg:Verb>
1361     <msg:Noun>QueryData</msg:Noun>
1362     <msg:Context>PRODUCTION</msg:Context>
1363     <msg:Timestamp>2012-11-30T09:30:48.004Z</msg:Timestamp>
1364   <Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
1365     ...
1366   </Signature>
1367   </msg:Header>
1368   <msg:Reply>
1369     <msg:Result>OK</msg:Result>
1370   </msg:Reply>
1371   <msg:Payload>
1372     <QueryData xmlns="urn:iec62325.504:messages:1:0">
1373       <RequestParameters>
1374         <Parameter>
1375           <name>DataType</name>
1376           <value>parameterLimits</value>
1377         </Parameter>
1378       </RequestParameters>
1379       <ParameterList xmlns="urn:iec62325.504:messages:1:0">
1380         <Parameter>
1381           <name>MaxNumMessagesInListResponse</name>
1382           <value>3000</value>
1383         </Parameter>
1384         <Parameter>
1385           <name>MaxTimeIntervalInDaysInListRequest</name>
1386           <value>3</value>
1387         </Parameter>
1388       </ParameterList>
1389     </QueryData>
1390   </msg:Payload>
1391 </msg:ResponseMessage>
1392 </soap:Body>
1393 </soap:Envelope>
1394

```

1395

### 1396 B.4.4 Generic Query Example

#### 1397 B.4.4.1 Request:

```

1398 <soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
1399   <soap:Header/>
1400   <soap:Body>
1401     <msg:RequestMessage xmlns:msg="http://iec.ch/TC57/2011/schema/message">
1402       <msg:Header>
1403         <msg:Verb>get</msg:Verb>
1404         <msg:Noun>QueryData</msg:Noun>
1405         <msg:Context>PRODUCTION</msg:Context>
1406         <msg:Timestamp>2012-11-30T09:30:47.581Z</msg:Timestamp>
1407       </msg:Header>
1408       <msg:Request>
1409         <msg:StartTime>2012-11-26T23:00:00Z</msg:StartTime>
1410         <msg:EndTime>2012-11-27T23:00:00Z</msg:EndTime>
1411         <msg:Option>
1412           <msg:name>DataType</msg:name>
1413           <msg:value>exampleWithParameters</msg:value>
1414         </msg:Option>
1415         <msg:Option>
1416           <msg:name>Parameter_1</msg:name>
1417           <msg:value>a value for parameter</msg:value>
1418         </msg:Option>
1419         <msg:Option>
1420           <msg:name>OtherParameter</msg:name>
1421           <msg:value>a value for other parameter</msg:value>
1422         </msg:Option>
1423       </msg:Request>

```

```

1424     </msg:RequestMessage>
1425     </soap:Body>
1426 </soap:Envelope>

```

1427

**B.4.4.2 Response:**

```

1429 <soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
1430   <soap:Header/>
1431   <soap:Body>
1432     <msg:ResponseMessage xmlns:msg="http://iec.ch/TC57/2011/schema/message">
1433       <msg:Header>
1434         <msg:Verb>reply</msg:Verb>
1435         <msg:Noun>QueryData</msg:Noun>
1436         <msg:Context>PRODUCTION</msg:Context>
1437         <msg:Timestamp>2012-11-30T09:30:48.004Z</msg:Timestamp>
1438       <Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
1439         ...
1440     </Signature>
1441     </msg:Header>
1442     <msg:Reply>
1443       <msg:Result>OK</msg:Result>
1444     </msg:Reply>
1445     <msg:Payload>
1446       <QueryData xmlns="urn:iec62325.504:messages:1:0">
1447         <RequestParameters>
1448           <Parameter>
1449             <name>DataType</name>
1450             <value>exampleWithParameters</value>
1451           </Parameter>
1452           <Parameter>
1453             <name>Parameter_1</name>
1454             <value>a value for parameter</value>
1455           </Parameter>
1456           <Parameter>
1457             <name>OtherParameter</name>
1458             <value>a value for other parameter</value>
1459           </Parameter>
1460         </RequestParameters>
1461         <n1:AnyKindOfDocument xmlns:n1="namespace declaration">
1462           ...
1463         </n1:AnyKindOfDocument>
1464       </QueryData>
1465     </msg:Payload>
1466   </msg:ResponseMessage>
1467 </soap:Body>
1468 </soap:Envelope>

```

1469

1470

1471

## 1472 B.5 Fault

### 1473 B.5.1 SOAP 1.2

1474 The following shows a fault message returned by a "List" operation where the given  
1475 TimeInterval is not valid, using SOAP 1.2:

```

1476 <soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
1477   <soap:Body>
1478     <soap:Fault>
1479       <soap:Code>
1480         <soap:Value>soap:Receiver</soap:Value>
1481       </soap:Code>
1482       <soap:Reason>
1483         <soap:Text xml:lang="es">EPF-04</soap:Text>
1484       </soap:Reason>
1485       <soap:Detail>
1486         <msg:FaultMessage
1487 xmlns:msg="http://iec.ch/TC57/2011/schema/message">
1488           <msg:Reply>
1489             <msg:Result>FAILED</msg:Result>
1490             <msg:Error>
1491               <msg:code>EPF-04</msg:code>
1492               <msg:details>Invalid Time Interval: 2011-03aT22:00Z2011-03-
1493 02T12:00Z </msg:details>
1494             </msg:Error>
1495           </msg:Reply>
1496         </msg:FaultMessage>
1497       </soap:Detail>
1498     </soap:Fault>
1499   </soap:Body>
1500 </soap:Envelope>

```

1501

### 1502 B.5.2 SOAP 1.1

1503 The following shows a fault message returned by a "List" operation where the given  
1504 TimeInterval is not valid, using SOAP 1.1:

```

1505 <soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
1506   <soap:Body>
1507     <soap:Fault>
1508       <faultcode>soap:Receiver</faultcode>
1509       <faultstring>EPF-04</faultstring>
1510       <detail>
1511         <msg:FaultMessage
1512 xmlns:msg="http://iec.ch/TC57/2011/schema/message">
1513           <msg:Reply>
1514             <msg:Result>FAILED</msg:Result>
1515             <msg:Error>
1516               <msg:code>EPF-04</msg:code>
1517               <msg:details>Invalid Time Interval: 2011-03aT22:00Z2011-03-
1518 02T12:00Z </msg:details>
1519             </msg:Error>
1520           </msg:Reply>
1521         </msg:FaultMessage>
1522       </detail>
1523     </soap:Fault>
1524   </soap:Body>
1525 </soap:Envelope>

```

1526

## 1527 B.6 Digital Signature

### 1528 B.6.1 Basic Example

1529 This is an example of a response to the Put service:

```

1530 <SOAP-ENV:Envelope xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
1531 xmlns:SOAP-ENV="http://www.w3.org/2003/05/soap-envelope">
1532   <SOAP-ENV:Header>
1533   </SOAP-ENV:Header>
1534   <SOAP-ENV:Body>
1535     <msg:ResponseMessage
1536 xmlns:msg="http://iec.ch/TC57/2011/schema/message">
1537       <msg:Header>
1538         <msg:Verb>reply</msg:Verb>
1539         <msg:Noun>Acknowledgement_MarketDocument</msg:Noun>
1540         <msg:Context>PRODUCTION</msg:Context>
1541         <msg:Timestamp>2014-04-15T13:06:29.885Z</msg:Timestamp>
1542         <Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
1543           <SignedInfo>
1544             <CanonicalizationMethod
1545 Algorithm="http://www.w3.org/TR/2001/REC-xm1-c14n-20010315"/>
1546             <SignatureMethod
1547 Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"></SignatureMethod>
1548             <Reference URI="">
1549               <Transforms>
1550                 <Transform
1551 Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"/>
1552                 <Transform Algorithm="http://www.w3.org/TR/2001/REC-xm1-
1553 c14n-20010315"></Transform>
1554               </Transforms>
1555               <DigestMethod
1556 Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
1557                 <DigestValue>g87DHbL2vOP4N4sCUBjby/M6K48=</DigestValue>
1558               </Reference>
1559             </SignedInfo>
1560
1561             <SignatureValue>sl3aQ+UsBzesV6EQfGidB7Iuyk/iLIQP1VlKtyOEUEphGfP3y7no2tV9tw+mH2
1562 FmId90hKCSurwnkKpKSryUNUDI9i5NPz+g+sI4N/hz0rVsSfh/CZV1ZaboVdVbngv/yc7JE77NFwR
1563 e6kOAmWnUvnSywx162ZOasuEXxc+olkD+7HtPzAR86JhB2oAqqdtcuc+FGuMlnsCi6p5whZRq99rkH
1564 vxnp3w2Y6b84nlJm/FJD9RsZJ0UInXVLMgALUmbxClKlKXwFUyLjSqY/SrFjFzkM0KshgWiFKOYhw
1565 GhUy0+UNCpRcGncOZj61tIuYD8bT205JITgSkdHO4bRmw==</SignatureValue>
1566             <KeyInfo>
1567               <X509Data>
1568                 <X509IssuerSerial>
1569                   <X509IssuerName>CN=testCA, O=test Domain</X509IssuerName>
1570                   <X509SerialNumber>131</X509SerialNumber>
1571                 </X509IssuerSerial>
1572                 <X509SubjectName> UID=myname, OU=intern, OU=myou, OU=users,
1573 O=test, C=test</X509SubjectName>
1574                 <X509Certificate>
1575 MIIICOTCCAaKgAwIBAAIEU04yjTANBgkqhkiG9w0BAQUFADBhMQ0wCwYDVQQIEwRL
1576 RVkhMQ8wDQYDVQQHEwZTRUNSRVQxDDAKBgNVBAoTAlRIRTEXMBUGA1UECXMOWU9V
1577 IEhBVkUgRk9VTkQxGDAWBgNVBAMTD0NPTkdsSQVRVTEFUSU90UzAeFw0xNDA0MTYw
1578 NzM0MzdaFw0xNTA4MjkwZmZlMzdaMGExDjALBgNVBAgTBETFWSEwDzANBgNVBAcT
1579 BlNFQ1JFVDEMAoGA1UEChMDVEhFMRCwFQYDVQQLEw5ZT1UgSEFWRSBGT1VORDEY
1580 MByGA1UEAxMPQ09OR1JBVFVtQVRJT05TMiGfMA0GCSqGSIb3DQEBAQUAA4GNADCB
1581 iQKBgQCQSGndj2T9o19clanXRkLeiVnFc8K1BhthMddZQ6kMZWcrFhKdBmrA0Fok
1582 JLYttpkVLCYHqcSP3+MygBITfPkhhoTPhPT2IjN4899cDGw10zG4fHjJF3Vn3S4v
1583 +TiUKilb0yfVHPL2KqSgld2X49yaNoc+97Rp2adYPuWxnLNHhWIDAQABMA0GCSqG
1584 SIb3DQEBAQUAA4GBAAn55WsZTpnRBkDhP9EwsMAqOf+4nGzYn+zqeKrczico8ylj
1585 9ndLF171aFMxQLVd8u8EaW8kGbo92Chh4z4vm9y4AFCFEhM5EwBA8w9tOw5l7oU9
1586 +NmMg6owGY6m+vP3l623C/lbv66dQOHUXLO5H/VVTOUGBoYoGXBTaC1sGvm8
1587 </X509Certificate>
1588               </X509Data>
1589             </KeyInfo>
1590           </Signature>
1591         </msg:Header>
1592         <msg:Reply>
1593           <msg:Result>OK</msg:Result>
1594         </msg:Reply>
1595         <msg:Payload>
1596 <Acknowledgement_MarketDocument xmlns="urn:iec62325.351:tc57wg16:451-
1597 1:acknowledgementdocument:6:0">

```

```
1598 <mRID>ACK_Schedule_D_20140416</mRID>
1599 <createdDateTime>2014-04-15T13:07:30Z</createdDateTime>
1600 <sender_MarketParticipant.mRID codingScheme="A01">10XEXAMPLE-EIC-P
1601 </sender_MarketParticipant.mRID>
1602 <sender_MarketParticipant.marketRole.type>A04
1603 </sender_MarketParticipant.marketRole.type>
1604 <receiver_MarketParticipant.mRID codingScheme="A01">10XEXAMPLE-EIC-P
1605 </receiver_MarketParticipant.mRID>
1606 <receiver_MarketParticipant.marketRole.type>A04
1607 </receiver_MarketParticipant.marketRole.type>
1608 <received_MarketDocument.mRID>Schedule_D_20140416
1609 </received_MarketDocument.mRID>
1610 <received_MarketDocument.revisionNumber>1
1611 </received_MarketDocument.revisionNumber>
1612 <received_MarketDocument.type>A51</received_MarketDocument.type>
1613 <received_MarketDocument.createdDateTime>2014-04-15T13:06:29Z
1614 </received_MarketDocument.createdDateTime>
1615 <Reason>
1616 <code>A01</code>
1617 </Reason>
1618 </Acknowledgement_MarketDocument>
1619 </msg:Payload>
1620 </msg:ResponseMessage>
1621 </SOAP-ENV:Body>
1622 </SOAP-ENV:Envelope>
```

1623  
1624

## Annex C (Informative)

### Java code Examples

1625  
1626  
1627  
1628

#### 1629 C.1 Sign

1630 The following simplified code excerpt signs a message following the rules described in this  
1631 Technical Specification:

1632

```
1633 /**  
1634 * Signs the given xml document using the given private key and certificate.  
1635 * @param msgAsDocument The document to be signed, the result of the process  
1636 * will be returned in this parameter.  
1637 * @param privateKey The private key to be used for signature.  
1638 * @param cert The certificate to be used for signature.  
1639 * @throws Exception If unable to sign the document.  
1640 * @see #signString(StringBuilder, RSAPrivateKey, X509Certificate)  
1641 */  
1642 public static void signDocument(final Document msgAsDocument, final  
1643 RSAPrivateKey privateKey, final X509Certificate cert) throws Exception {  
1644  
1645     try {  
1646         XMLSignatureFactory fac = XMLSignatureFactory.getInstance("DOM");  
1647  
1648         List<Transform> trfLst = new ArrayList<>();  
1649         trfLst.add(fac.newTransform(Transform.ENVELOPED,  
1650 (TransformParameterSpec) null));  
1651  
1652         trfLst.add(fac.newCanonicalizationMethod(CanonicalizationMethod.INCLUSIVE,  
1653 (C14NMethodParameterSpec) null));  
1654  
1655         Reference ref = fac  
1656             .newReference("", fac.newDigestMethod(DigestMethod.SHA1,  
1657 null), trfLst, null, null);  
1658  
1659         SignedInfo si =  
1660 fac.newSignedInfo(fac.newCanonicalizationMethod(CanonicalizationMethod.INCLUSI  
1661 VE, (C14NMethodParameterSpec) null),  
1662 fac.newSignatureMethod(SignatureMethod.RSA_SHA1, null),  
1663             Collections.singletonList(ref));  
1664  
1665         Node headerNode = null;  
1666         NodeList nl =  
1667 msgAsDocument.getElementsByTagNameNS("http://iec.ch/TC57/2011/schema/message",  
1668 "Header");  
1669         if (nl.getLength() == 1) {  
1670             headerNode = nl.item(0);  
1671         } else {  
1672             throw new Exception("Invalid document.");  
1673         }  
1674  
1675         DOMSignContext dsc = new DOMSignContext(privateKey, headerNode);  
1676  
1677         KeyInfoFactory keyInfoFactory = fac.getKeyInfoFactory();  
1678         List<Object> x509Content = new ArrayList<>();  
1679  
1680         x509Content.add(keyInfoFactory.newX509IssuerSerial(cert.getIssuerDN().getName(  
1681 ), cert.getSerialNumber()));  
1682         x509Content.add(cert.getSubjectX500Principal().getName());  
1683         x509Content.add(cert);  
1684         X509Data xd = keyInfoFactory.newX509Data(x509Content);  
1685  
1686         KeyInfo keyInfo =  
1687 keyInfoFactory.newKeyInfo(Collections.singletonList(xd));  
1688
```



```

1689     XMLSignature signature = fac.newXMLSignature(si, keyInfo);
1690     signature.sign(dsc);
1691
1692     } catch (Exception e) { //Simplified exception handling for brevity
1693         throw e;
1694     }
1695 }
1696

```

## 1697 C.2 Verify

1698 The following simplified code excerpt verifies a message's signature following the rules  
 1699 described in this Technical Specification

```

1700
1701 /**
1702  * Verifies the signature of the given signed document.
1703  * @param msgAsDocument The document to be validated.
1704  * @throws Exception If the document cannot be validated
1705  *         or if its signature is invalid.
1706  */
1707 public static void verifyDocument(final Document msgAsDocument) throws
1708 Exception {
1709
1710     try {
1711         Node signatureNode = null;
1712         NodeList nl = msgAsDocument.getElementsByTagNameNS (XMLSignature.XMLNS,
1713 "Signature");
1714         if (nl.getLength() == 1) {
1715             signatureNode = nl.item(0);
1716         } else {
1717             throw new Exception("Invalid document.");
1718         }
1719
1720         XMLSignatureFactory fac = XMLSignatureFactory.getInstance("DOM");
1721         DOMValidateContext valContext = new DOMValidateContext(new
1722 KeyValueKeySelector(), signatureNode);
1723         XMLSignature signature = fac.unmarshalXMLSignature(valContext);
1724         boolean coreValidity = signature.validate(valContext);
1725
1726         if (coreValidity == false) {
1727             StringBuilder sb = new StringBuilder("Signature validation failed.
1728 Signature status: ");
1729             sb.append(signature.getSignatureValue().validate(valContext));
1730             sb.append(". ");
1731             Iterator<?> i =
1732 signature.getSignedInfo().getReferences().iterator();
1733             for (int j = 0; i.hasNext(); j++) {
1734                 boolean refValid = ((Reference)
1735 i.next()).validate(valContext);
1736                 sb.append("ref[").append(j).append("] validity status:
1737 ").append(refValid);
1738             }
1739             throw new Exception(sb.toString());
1740         }
1741     } catch (Exception e) { //Simplified exception handling for brevity
1742         throw e;
1743     }
1744 }

```

1745  
 1746

1747 **Annex D**  
1748 **(Informative)**

1749 **Regarding Near Real-time Communications**  
1750

1751 The web services presented in this Technical Specification allow for the implementation of  
1752 Near Real-Time (or even Real Time) communications with responses in the range of seconds.  
1753 However specific operational limits cannot and should not be set in stone in this Technical  
1754 Specification as these may vary greatly depending on the specific business rules that apply to  
1755 the exchanged payload.

1756 For example: the time constraints of a continuous intraday market are completely different  
1757 from a yearly capacity auction.

1758 Parties responsible for managing systems with hard time constraints will need to make sure  
1759 that their systems are fast enough to provide the necessary throughput. The communication  
1760 layer, being direct communication from client to server, and being a synchronous invocation,  
1761 will ensure that the additional delay introduced by the communication layer will be negligible,  
1762 thus the term Near Real-Time.

1763 Note that of the different services presented, the preferred one for hard real-time  
1764 requirements would be “Put” since the client has an immediate response to the submission of  
1765 a payload. List and Get services used to poll data should be used when more relaxed  
1766 deadlines are allowed.

1767 The “Get” service can only guarantee Near Real-Time in the sense that when a client  
1768 requests a message, it will receive it with as small delay as possible, but not in the sense of it  
1769 being part of a more complex communication (or dialog) involving several messages  
1770 exchanged between client and server.