ETSI TS 102 640-6-3 V1.1.1 (2011-09)

Electronic Signatures and Infrastructures (ESI);
Registered Electronic Mail (REM);
Part 6: Interoperability Profiles;
Sub-part 3: REM-MD SOAP Binding Profile

Reference
DTS/ESI-000069-1

Keywords
e-commerce, electronic signature, email, security,
trust services

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - NAF 742 C
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° 7803/88

Contents

Intellectual Property Rights

IPRs essential or potentially essential to the present document may have been declared to ETSI. The information pertaining to these essential IPRs, if any, is publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (http://ipr.etsi.org).

Pursuant to the ETSI IPR Policy, no investigation, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Foreword

This Technical Specification (TS) has been produced by ETSI Technical Committee Electronic Signatures and Infrastructures (ESI).

The present document is part 6, sub-part 3 of a multi-part deliverable. Full details of the entire series can be found in part 1 [1].

Introduction

The summarised scope of each part and sub-part can be found in part 1 [1] of this multi-part deliverable.

1 Scope

The purpose of the present document is to define specifications on how to carry REM Dispatches and REM-MD Messages between REM-MDs as XML Information Set as defined by the SOAP specification: "for exchanging structured and typed information between peers in a decentralized, distributed environment" (SOAP Version 1.2, Part 0: Primer), commonly called "Web Services". The present document comes as a completion of the current specifications (TS 102 640, especially parts 2 [2] and 5 [5]), which defines S/MIME envelopes as message format to be transported over SMTP protocol.

REM over SOAP will prove useful in several contexts, due to the fact that Web Services are largely considered a well established and flexible technology, providing detailed specifications for the different functional building blocks (addressing, security and trust, reliable delivery). Building blocks are combinable and open for extension/profiling according to the needs of specific application- and communication scenarios.

Several initiatives are ongoing pointing in this direction: we remark European projects SPOCS and STORK, which aim at bridging existing eDelivery systems in several European MSs. The necessity to have them all interchange trusted messages requires the involvement of "eDelivery Gateways" based on a "eDelivery meta-protocol", in order to avoid a non-scalable one-to-one bridging. Requirements for the meta-protocol normally involve the usage of a Web Services based transport (see e.g. STORK D6.4.1 [i.2], SPOCS D3.2 [i.1]). REM specifications as defined in TS 102 640 would be a natural candidate for the above meta-protocol role, once a proper binding to SOAP is defined.

Unlike the protocol stack defined for e-mail, standard Web Service specifications define no general message format to structure the content of more or less "unbounded" asynchronous exchange of messages and electronic documents: the SOAP body normally is seen as an opaque object, whose structure and semantics are agreed upon a specific Web Service provider and their respective consumers. Most of mentioned eDelivery solutions based on SOAP/Web Services define their domestic format for such general communication scenarios. To be able to provide interoperable message exchange functionality between such solutions as well as the SMTP/(S)MIME based world, the present document for REM/SOAP binding includes the definition of an XML-based exchange format for message contents, which may be used for mapping between different domestic and/or standardized message structures.

A further challenge of bridging the SMTP- and Web Services solutions is having to deal with different schemes of electronic addresses of end-entities (e.g. e-mail addresses as defined by RFC 5322 [11], URLs of http-resources, constructs following ISO/IEC 15459-3 [25] for unique identifier schemes). To this purpose, the definition of electronic addresses in REM has been extended to take into account the "addressing schema".

To meet the expectations above, the present document provides:

a) Rules for building a REM-MD Envelope (and, consequently, a REM Dispatch or a REM-MD Message) as well defined XML Information Sets (Infoset).

b) Rules for secure transport of the above REM-MD XML Infosets using SOAP, combined with appropriate bricks of the Web Services stack (profiling of WS-Addressing and WS-Security).

REM-MD Evidence formats respect TS 102 640-2 [2] specifications in xml flavour.

The structure of the present document is as follows:

- Clause 2 contains the list of normative and informative references.
- Clause 3 includes definitions of the relevant concepts to the present document and abbreviations.
- Clause 4 contains the specification of REM-MD XML Infosets to be used for enveloping messages. Specific syntax is addressed by annex A.
- Clause 5 contains the specification of the SOAP messages as exchanged between REM-MDs, which covers the profiling of the standard WS-bricks used. Profiling details are addressed by annex B.
- Clause 6 deals with the definition of Web Services for interoperability.
- Annex A provides XML Schema for REM XML Infosets as used inside SOAP messages.
- Annex B provides a profiling for WS Addressing inside SOAP header.

- Annex C provides WSDL specification, defining the REM-MD Web Service endpoint.

2 References

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the reference document (including any amendments) applies.

Referenced documents which are not found to be publicly available in the expected location might be found at http://docbox.etsi.org/Reference.

NOTE: While any hyperlinks included in this clause were valid at the time of publication ETSI cannot guarantee their long term validity.

2.1 Normative references

The following referenced documents are necessary for the application of the present document.

[1] ETSI TS 102 640-1: "Electronic Signatures and Infrastructures (ESI); Registered Electronic Mail (REM); Part 1: Architecture".

[2] ETSI TS 102 640-2: "Electronic Signatures and Infrastructures (ESI); Registered Electronic Mail (REM); Part 2: Data requirements, Formats and Signatures for REM".

[3] ETSI TS 102 640-3: "Electronic Signatures and Infrastructures (ESI); Registered Electronic Mail (REM); Part 3: Information Security Policy Requirements for REM Management Domains".

[4] ETSI TS 102 640-4: "Electronic Signatures and Infrastructures (ESI); Registered Electronic Mail (REM); Part 4: REM-MD Conformance Profiles".

[5] ETSI TS 102 640-5: "Electronic Signatures and Infrastructures (ESI); Registered Electronic Mail (REM); Part 5: REM-MD Interoperability Profiles".

[6] IETF RFC 2616: "Hypertext Transfer Protocol -- HTTP/1.1".

NOTE: Available at http://www.rfc-editor.org/rfc/rfc2616.txt.

[7] IETF RFC 2817: "Upgrading to TLS Within HTTP/1.1".

NOTE: Available at http://tools.ietf.org/html/rfc2817.

[8] IETF RFC 3061 (2001): "A URN Namespace of Object Identifiers".

[9] IETF RFC 3986: "Uniform Resource Identifier (URI): Generic Syntax".

[10] IETF RFC 4122 (2005): "A Universally Unique Identifier (UUID) URN Namespace".

NOTE: Available at http://www.ietf.org/rfc/rfc4122.txt.

[11] IETF RFC 5322: "Internet Message Format".

NOTE: Available at http://tools.ietf.org/html/rfc5322.

[12] OASIS Standard Specification: "OASIS Web Services Security (WSS) TC".

NOTE: Available at http://www.oasis-open.org/specs/index.php#wssv1.1.

[13] OASIS Standard Specification: "Web Services Reliable Messaging (WS-ReliableMessaging) Version 1.2", 2 February 2009.

NOTE: Available at http://docs.oasis-open.org/ws-rx/wsrm/v1.2/wsrm.pdf.

[14] OASIS Standard Specification: "Web Services Reliable Messaging Policy Assertion (WS-RM Policy) Version 1.1", 7 January 2008.

NOTE: Available at http://docs.oasis-open.org/ws-rx/wsrmp/v1.1/wsrmp.pdf.

[15] W3C Recommendation: "SOAP Message Transmission Optimization Mechanism" 25 January 2005.

NOTE: Available at http://www.w3.org/TR/soap12-mtom/.

[16] W3C Recommendation: "SOAP Version 1.2 Part 1: Messaging Framework (Second Edition)" 27 April 2007.

NOTE: Available at http://www.w3.org/TR/soap12-part1/.

[17] W3C Recommendation: "Web Services Addressing 1.0 - SOAP Binding" 9 May 2006.

NOTE: Available at http://www.w3.org/TR/2006/REC-ws-addr-soap-20060509/.

[18] W3C Note: "Web Services Description Language (WSDL) 1.1" 15 March 2001.

NOTE: Available at http://www.w3.org/TR/wsdl/.

[19] W3C Recommendation: "Web Services Policy 1.5 - Framework" 04 September 2007.

NOTE: Available at http://www.w3.org/TR/ws-policy/.

[20] W3C Working Draft: "MTOM Serialization Policy Assertion 1.1" 18 September 2007.

NOTE: Available at http://www.w3.org/TR/soap12-mtom-policy/.

[21] Web Services Interoperability Organization Working Group Draft WS-I: "Basic Profile 2.0" 2007-10-25.

NOTE: Available at http://www.ws-i.org/Profiles/BasicProfile-2_0(WGD).html.

[22] ISO 3166-1 (2006): "Codes for the representation of names of countries and their subdivisions -- Part 1: Country codes".

NOTE: Updates available at http://www.iso.org/iso/country_codes/updates_on_iso_3166.htm.

[23] ETSI TS 102 640-6-1: "Electronic Signatures and Infrastructures (ESI); Registered Electronic Mail (REM); Part 6: Interoperability Profiles; Sub-part 1: REM-MD UPU PReM Interoperability Profile".

[24] ETSI TS 102 640-6-2: "Electronic Signatures and Infrastructures (ESI); Registered Electronic Mail (REM); Part 6: Interoperability Profiles; Sub-part 2: REM-MD BUSDOX Interoperability Profile".

[25] ISO/IEC 15459-3:2006: "Information technology -- Unique identifiers -- Part 3: Common rules for unique identifiers".

2.2 Informative references

The following referenced documents are not necessary for the application of the present document but they assist the user with regard to a particular subject area.

[i.1] SPOCS D3.2 Functional Specification, Architecture and Trust Model. In particular Appendix 3: eDelivery Interconnect Protocol and Gateway Specification.

NOTE: Available at http://www.eu-spocs.eu/index.php?option=com_processes&task=streamFile&id=18&fid=699.

[i.2] STORK D6.4.1 - eDelivery Functional Specification, 08/11/2009.

NOTE: Available at https://www.eid-stork.eu/index.php?option=com_processes&act=list_documents&s=1&Itemid=60&id=312.

[i.3] ISO/IEC 27001:2005: "Information technology -- Security techniques -- Information security management systems -- Requirements".

3 Definitions and abbreviations

3.1 Definitions

For the purposes of the present document, the terms and definitions given in TS 102 640-1 [1] apply.

Throughout the present document a number of verbal forms are used, whose meaning is defined below.

- **shall, shall not:** indicate requirements strictly to be followed in order to conform to the present document and from which no deviation is permitted.
- **should, should not:** indicate that among several possibilities one is recommended as particularly suitable, without mentioning or excluding others, or that a certain course of action is preferred but not necessarily required, or that (in the negative form) a certain possibility or course of action is deprecated but not prohibited.
- **may, need not:** indicate a course of action permissible within the limits of the present document.

3.2 Abbreviations

For the purposes of the present document, the abbreviations given in TS 102 640-1 [1] and the following apply:

OID	Object Identifier
QES	Qualified Electronic Signature
R-REM-MD	Recipient's REM-MD
S-REM-MD	Sender's REM-MD
URI	Uniform Resource Identifier

4 XML-based REM-MD Envelope Structure Implementation

The present document provides a common format for electronic messages and documents, suitable for the exchange of those by means given by e-mail technology as well as Web Services technology.

This implies the definition of an xml REM-MD Envelope, as well as the capability to deal with different schemes in use for e-addresses of the nodes involved in the message flow (end entities and transfer agents).

The REM-MD envelope is modelled by XML Infosets to be carried in the body of a SOAP message. The `<s12:Body>` is the enveloping element for a REM Object. SOAP messages are used for REM Object transport between SOAP based instances of REM-MD. A REM Object is either a REM-Dispatch, which contains the original message or a REM-MD Message. Definitions of the above mentioned entities are provided in clause 3.1 of TS 102 640-1 [1].

4.1 REM Dispatch

The element <remsoap:REMDispatch> has the same purpose and carries the same information as the S/MIME structure defined in TS 102 640-2 [2], clause 6.

A <remsoap:REMDispatch> **shall** contain the original message <remsoap:OriginalMsg> in untouched format. Sender's REM-MD generates the <remsoap:REMDispatch> by creating REM-MD Evidence as specified in TS 102 640-1 [1] and TS 102 640-2 [2], which then are included in <remsoap:REMMDEvidenceList>. Sender's REM-MD **shall** include the <remsoap:MsgMetaData> as a child element of a <remsoap:REMDispatch>, which **should** be signed by the generating REM-MD instance.

Figure 1 gives an overview of high level structure of the <remsoap:REMDispatch> element:

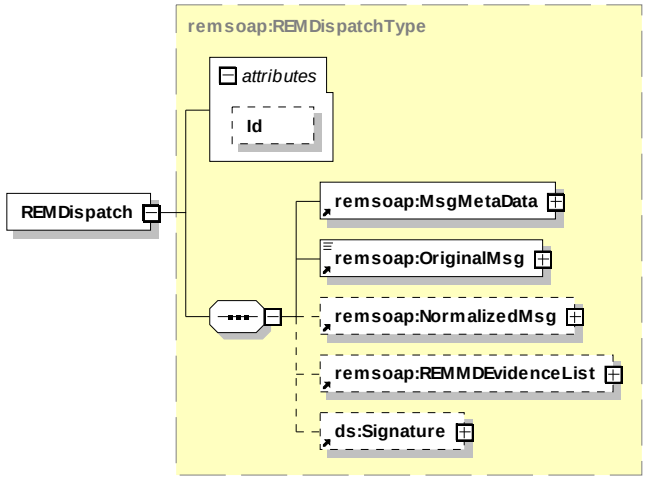

Figure 1: REMDispatch constituents

To facilitate message format conversion between REM-MDs each of which is using different packaging formats for the original message, sender's UA or sender'S REM-MD **may** produce a "normalized" (xml) form of the original message <remsoap:NormalizedMsg> as defined below. The usage of this normalized form is intended to disburden REM-MDs from the need to have knowledge of syntax and semantics of all foreign REM-MD message formats.

If the <remsoap:NormalizedMsg> complex element is produced directly by the sender's UA, it will coincide with the original message.

If present, the element <remsoap:NormalizedMsg> **shall** be included in the signature value calculation of the REMDispatch, to attest correct mapping of <remsoap:OriginalMsg> and <remsoap:NormalizedMsg> by processing REM-MD instance.

4.2 REM-MD Message

A <remsoap:REMMDMessage> contains REM-MD evidence as specified in TS 102 640-1 [1] and TS 102 640-2 [2], which again are included in <remsoap:REMMDEvidenceList>. The original message <remsoap:OriginalMsg> as well as the normalized correspondent <remsoap:NormalizedMsg> **may** be included in the <rem:REMMDMessage>. As meta data about the message a REM evidence is related to is contained in the REM-MD evidence itself, the element <rem:MsgMetaData> can be omitted in this case. <rem:REMMDMessage> **should** be signed by the generating REM-MD instance.

Figure 2 gives an overview of high level structure of the <rem:REMMDMessage> element.

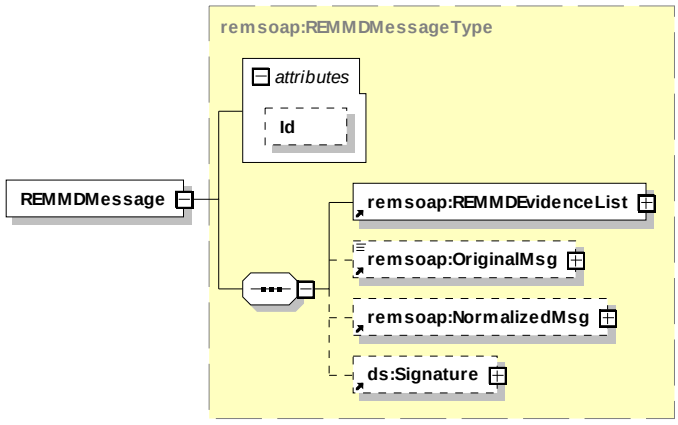

Figure 2: REMDMessage constituents

The detailed definition of <remsoap:REMDispatch> and <remsoap:REMMDMessage> in annex A is preceded by the description of complex types and elements reused and extensions defined for the original schema http://uri.etsi.org/02640/v1#.

5 Message Transport between REM-MD using SOAP

For the general layout of the SOAP Header and Body, message transport and –security mechanisms an overview is given in the following clauses. In detail, constituents and structure are formally described in WDSL 1.1 notation and according policies in annex C.

5.1 SOAP Version and Binding

REM-MDs **shall** support SOAP Version 1.2 according to [16] and constraints specified in WSI-Basic [21], section Messaging with the provision that the SOAP Message Transmission Optimization Mechanism [15] **shall** be supported by conformant implementations.

Transport binding is restricted to HTTP/1.1 [6].

5.2 SOAP Header

5.2.1 Addressing

For addressing a remote REM-MD, a SOAP header using WS-Addressing specification in inserted.

REM-MD **shall** support WS-Addressing according to [14]. Constraints apply specified in WSI-Basic [17], section 3.6 "Support for WS-Addressing Messaging" and section 3.7 "Use of WS-Addressing MAPs".

REM-MD **shall** support WS-Addressing SOAP Binding according to [21] and [15], whereby only the rules for binding to SOAP 1.2 apply.

Basically, <wsa:To> carries the URL of the destination REM-MD and <wsa:ReplyTo>/<wsa:Address> outlines the one of the source REM-MD of a message. The respective URL value **should** be the one of the service supply point element as defined in the according TSL entry for the REM-MD.

The <wsa:MessageID> is the ID of the message provided by the sender's REM-MD, not to be confused with the initial ID assigned to the message by the sender.

The present document defines some restrictions on the cardinality of WS-Addressing message addressing properties carried as SOAP header elements as outlined in Web Services Addressing 1.0 - SOAP Binding [17].

A specific profiling of WS-Addressing tags is provided in annex B.

5.2.2 WS Security header

The entries in <wsse:Security> **shall** conform to the WS Security specification [12].

<wsse:Security> header block **shall** be present, carrying authentication information of the message source REM-MD.

The authentication token is restricted to type http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-x509-token-profile-1.0#X509v3 **(s**ee [7] for details). This certificate **should** be the one exposed as signature certificate in the according TSL entry for the REM-MD building up the SOAP message.

A wsu:TimeStamp **shall** be present in this header, outlining the validity time span of the security semantics.

The mandatory signature element in the /wsse:Security header **shall** cover the message part wsu:TimeStamp. The X509 certificate mentioned before **shall** be used for signature generation.

Signature and encryption of the whole message **shall** be realized with TLS mechanisms (HTTPS) according to [7] and [8].

5.2.3 Use of WS-ReliableMessaging

Message of type REMDispatch and REMMDMessage **shall** be delivered in a controlled way exactly once. To ensure interoperability for these mechanisms, the present document incorporates the WS-ReliableMessaging specification (version 1.2), see [13] which is implemented by most of the WS-Stack implementations.

5.3 SOAP Body Format

The SOAP body content covers either a <rem:REMDispatch> or <rem:REMMDMessage> complex element, depending on the <wsa:Action> outlined in the SOAP header. These elements are detailed in clause 4.

5.4 SOAP Fault Binding

The management of errors occurring while processing a SOAP message uses the SOAP fault mechanisms. The SOAP fault block according to [16] **shall** be used to report information about errors. The <s12:Fault> element **shall** be carried in the SOAP body block of the network backchannel SOAP response message.

When the sending REM-MD gets such a SOAP fault, it **shall** produce the according REM-MD Evidence for the sender.

Following information for the subelements s12:Fault is supplied per fault described in the present document.

Table 1: SOAP fault subelements

Subelement	Possible / mandatory values
../Code/Value	as defined in SOAP12 [16], section 4.6.4
../Code/Value/Subcode	a local <xs:QName> assigned to the fault
../Reason/Text	reason explanation (in English)
../Fault/Node	URI of REM-MD raising the fault

In the fault message itself, the [Code] value **shall** have a namespace prefix of s12:, the [Subcode] value **shall** be taken from TS 102 640-2 [2], annex D.

The optional <s12:Role> element of <s12:Fault> can be omitted as not interpreted in the present document, which only deals with SOAP nodes in the role "REM-MD".

Implementations **may** provide second-level <s12:Detail> fields of <s12:Fault>.

5.4.1 General processing error

If an unspecific and unrecoverable message processing error occurs on a SOAP call, a SOAP fault **shall** be generated by the receiving REM-MD, which **shall** also discard the message. The fault **shall** have the following value for attributes:

Fault 1: TechnicalMalfunction	
../Code/Value	Receiver
../Code/Value/Subcode	`http:uri.etsi.org/REM/EventReason#R-REMMD_Malfunction`
../Reason/Text	Unspecific processing error

Implementations **may** provide second-level details fields, e.g. a stack trace, if this information does not lead to security vulnerabilities.

A source REM-MD in this case **shall** generate an Evidence with following details.

Table 2: REM-Event in case of technical malfunction

RelayToREMMDFailure	
Element	**Possible / mandatory values**
`EventCode`	`Rejection`
`EventReasons`	*with at least one child element:*
`../EventReason`	*and child elements:*
`../../Code`	`http:uri.etsi.org/REM/EventReason#R-REMMD_Malfunction`
`../../Details`	Value of s12:/Fault/node (URI of REM-MD raising the fault)
`../EventReason` *****	*Further elements, if s12:/Fault/Detail elements present:*
`../../Code`	Value of namespace URI `in s12:/Fault/Detail`
`../../Details`	Concatenation of element and attribute value of `s12:/Fault/Detail`, separated by "::"

6 REM Web Service Specification

Transmission of REM-MD Messages and REM Dispatches conformant to REM SOAP interoperability profile **shall** be performed according to a specific service interface. The interface shall be defined in conformance to [18].

The present clause describes the interface operations which shall be implemented by REM-MDs:

- *AcceptREMDispatchOperation*
- *AcceptREMMDMessageOperation*

Specification of request/response elements, as well as a complete wsdl file are provided in annex C. The specification conforms to WS-Policy according to [19], [20], [21] and [22].

6.1 AcceptREMDispatchOperation

The **AcceptREMDispatchOperation** is invoked by the sender's REM-MD on a recipient's REM-MD in order to send a REM Dispatch to a given destination. The operation is implemented as a SOAP call, according to the specifications of clause 5, where the request contains a REM Dispatch, while the response contains REM MD Message; both objects are defined in annex A.

The request (REM Dispatch) contains the original message (possibly in addition to normalized form) plus REM-MD Evidence objects (in the normal case a *SubmissionAcceptanceRejection* evidence is expected - this is for the recipient to have a proof of the message submission by the sender).

The backchannel response will normally contain a *RelayToREMMDAcceptanceRejection* evidence - this is for the sender's REM-MD to have a proof of the take in charge by the recipient's REM-MD.

Therefore, to send a Dispatch to a user on a different REM-MD, the sender's REM-MD **shall** call the *AcceptREMDispatchOperation* method published by the recipient's REM-MD server. The recipient's REM-MD, upon receiving method call from the sender's REM-MD, **shall** extract original/normalized message and evidence from the REM Dispatch and forward the message to the End User according to its policies. The recipient's REM-MD **shall** also generate the appropriate REM-MD Evidence and send them back to sender's REM-MD either on the backchannel or by calling sender's REM-MD's *AcceptREMMDMessageOperation*.

6.2 AcceptREMMDMessageOperation

The *AcceptREMMDMessageOperation* operation is normally invoked by the recipient's REM-MD on the sender's REM-MD in order to provide some evidence related to events on a REM Dispatch which has been previously transmitted by the local REM-MD to the remote REM-MD (via *AcceptREMDispatchOperation*). The operation is implemented as a one-way SOAP call, according to the specification of clause 7.

Recipient's REM-MD call to sender's REM-MD *AcceptREMDispatchOperation* may happen before or after the message has been forwarded to the recipient, hence evidence **shall** be in the form of one or more REM-MD Evidences as defined in TS 102 640-2 [2].

Annex A (normative): Specifications for XML-based REM-MD Envelope

A.1 Namespace for the elements specified in the present document

For The XML namespace URI that **shall** be used by implementations of the present document:

- http://uri.etsi.org/02640/soapbinding/v1#

The following namespace declarations apply for the XML Schema definitions throughout the present document:

Table A.1: Namespaces

Namespace's URI	Namespace's prefix
http://uri.etsi.org/02640/v1#	rem
http://www.w3.org/2001/XMLSchema	xs
http://www.w3.org/2000/09/xmldsig#	ds
http://uri.etsi.org/02231/v2#	tsl
http://uri.etsi.org/01903/v1.3.2#	xades
http://www.w3.org/2005/05/xmlmime	xmime
http://www.w3.org/2003/05/soap-envelope	s12
http://uri.etsi.org/02640/soapbinding/v1#	remsoap
http://www.w3.org/XML/1998/namespace	xml
http://uri.etsi.org/02640/metadata#v1	ns1
http://uri.etsi.org/02640/remmdsoaptemplate	tns
http://www.w3.org/2007/05/addressing/metadata	wsam
http://schemas.xmlsoap.org/wsdl/	wsdl
http://schemas.xmlsoap.org/wsdl/soap	soap
http://schemas.xmlsoap.org/ws/2004/08/addressing	wsa
http://schemas.xmlsoap.org/ws/2004/09/policy	wsp
http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd	wsu
http://schemas.xmlsoap.org/ws/2005/07/securitypolicy	sp
http://schemas.sun.com/2006/03/wss/server	sc
http://docs.oasis-open.org/ws-rx/wsrmp/200702	wsrmp
http://schemas.xmlsoap.org/ws/2004/09/policy/optimizedmimeserialization	wspmtom

A.2 Element <REMDispatch> details

The <REMMDDispatch> element is the container for the XML REM-MD Dispatch contents.

Below follows the schema definition for the data type:

```
<xs:element name="REMDispatch" type="remsoap:REMDispatchType"/>
<xs:complexType name="REMDispatchType">
        <xs:sequence>
            <xs:element ref="remsoap:MsgMetaData"/>
            <xs:element ref="remsoap:OriginalMsg"/>
            <xs:element ref="remsoap:NormalizedMsg" minOccurs="0"/>
            <xs:element ref="remsoap:REMMDEvidenceList" minOccurs="0"/>
            <xs:element ref="ds:Signature" minOccurs="0"/>
        </xs:sequence>
        <xs:attribute name="Id" type="xs:ID"/>
    </xs:complexType>
```

The @Id attribute **shall** be present whenever the signature on the envelope is present. It is provided for referencing purposes from the ds:Signature element to be applied finally by the REM-MD that generates the REM-MD Dispatch.

Clauses below provide details of the <REMDispatch> child elements.

A.2.1 Element <MsgMetaData>

This element contains metadata related to transport of the message. Elements mimic those defined in RFC 5322 [11].

```
<xs:element name="MsgMetaData">
        <xs:complexType>
            <xs:sequence>
                <xs:element ref="remsoap:DeliveryConstraints"/>
                <xs:element ref="remsoap:Originators"/>
                <xs:element ref="remsoap:Destinations"/>
                <xs:element ref="remsoap:MsgIdentification"/>
            </xs:sequence>
        </xs:complexType>
</xs:element>
```

Clauses below provide details for each child element of <MsgMetaData>.

A.2.1.1 Element <DeliveryConstraints>

```
<xs:element name="DeliveryConstraints">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="Origin" type="xs:dateTime" minOccurs="0"/>
                <xs:element name="InitialSend" type="xs:dateTime"/>
                <xs:element name="ObsoleteAfter" type="xs:date" minOccurs="0"/>
                <xs:element ref="xades:Any" minOccurs="0"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
```

This complex element carries following delivery time stamps and constraints:

- The optional origination date <Origin> specifies the xs:dateTime at which the sender of the message indicated that the Original Message was complete and ready to enter the REM-MD system.
- <InitialSend> indicates the mandatory xs:dateTime at which the Sender's REM-MD initiated delivery.
- The sender **may** provide an element <ObsoleteAfter> as xs:Date. If the optional <ObsoleteAfter> element is present, it indicates the date and time of latest recipient's access to the message as requested by the sender of the original message. The means used by the sender to indicate her REM-MD this date and time are out of the scope of the present document. This information is useful for instance, for delivery monitoring and escalation routines.
- The optional element <xades:Any> may be used for extensions to be defined on mutual agreement between REM MD's, based on schema definitions in namespaces other than used in the present document. This extension point **may** be used to define further delivery constraints like e.g. delivery priority, recipient authentication strength level to access the message.

A.2.1.2 Element <Originators>

The <Originators> element carries message source information elements as foreseen in RFC 5322 [11].

```
<xs:element name="Originators">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="From" type="rem:EntityDetailsType"/>
                <xs:element name="Sender" type="rem:EntityDetailsType" minOccurs="0"/>
                <xs:element name="ReplyTo" type="rem:EntityDetailsType" minOccurs="0"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
```

The mandatory element <From> outlines the originator of a message - which in terms of the present document is the initial Sender.

The optional element <Sender> outlines the entity responsible for the actual transmission of the message, e.g. a delegate of the entity outlined in the element <From>. It **should not** be used, if identical to the value of <From>.

The optional element <ReplyTo> outlines then entity replies to the message **should** be sent to.

A.2.1.3 Element <Destinations>

The <Destinations> element carries information elements for the intended Recipients of a message; Recipients may be in different roles.

```
<xs:element name="Destinations">
    <xs:complexType>
        <xs:sequence>
            <xs:element ref="remsoap:Recipient"/>
            <xs:element name="OtherRecipients">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element name="To" type="rem:EntityDetailsType" maxOccurs="unbounded"/>
                        <xs:element name="Cc" type="rem:EntityDetailsType" minOccurs="0"
maxOccurs="unbounded"/>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
        </xs:sequence>
    </xs:complexType>
</xs:element>
```

```
    <xs:element name="Recipient" type="rem:EntityDetailsType"/>
```

Mandatory element <remsoap:Recipient> outlines the ultimate Recipient of the REMDispatch. It is assumed, REM MD's forward separate REMDispatches per Recipient outlined by the Sender (REMDispatch is cloned by source REM-MD per entity given in <Destinations> initially by the Sender). This element **shall** be supplied with one of the values of the <OtherRecipients> child elements <To> or <Cc> described below. Both are included in the <MetaData> element to transmit the complete initial destination information to each destination REM-MD for further consumption (e.g. final delivery to the entity described by <remsoap:Recipient>).

A.2.1.4 Element <MsgIdentification>

This element carries the Dispatch identifier and optional correlation information.

```
<xs:element name="MsgIdentification">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="Message-ID" type="xs:string"/>
                <xs:element name="In-Reply-To" type="xs:string" minOccurs="0"
maxOccurs="unbounded"/>
                <xs:element name="References" type="xs:string" minOccurs="0" maxOccurs="unbounded"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
```

Element `<Message-Id>` carries the mandatory ID of this message as assigned in the sender's REM-MD, its type is `xs:string`. It is strongly advised that ID generation **should** facilitate non-ambiguous cross-REM-MD identification of messages (e.g. according to RFC 4122 [10], concatenated by "@" with the MD identifier URI).

Elements `<In-Reply-To>` are used to identify the message(s) to which the new message is a reply, its type is `xs:string`. It **shall** be present if this information is present in the original message as submitted by the Sender.

Elements `<References>` are used to identify the messages/REMDispatches (and even REM-MD Evidence) inside a "thread" of conversation, type is `<xs:string>`. It **shall** be present, if this information is present in the original message as submitted by the Sender.

A.2.2 Element <OriginalMsg>

This element **shall** carry the original message as provided by the sender in untouched, base64 encoded binary format.

```
<xs:element name="OriginalMsg" type="remsoap:OriginalMsgType"/>

<xs:complexType name="OriginalMsgType">
    <xs:simpleContent>
        <xs:extension base="xs:base64Binary">
            <xs:attribute name="ContentType" type="tsl:NonEmptyString" use="required"/>
            <xs:attribute name="Size" type="xs:positiveInteger" use="required"/>
        </xs:extension>
    </xs:simpleContent>
</xs:complexType>
```

The `remsoap:OriginalMsgType` is based on type `xs:base64binary`, extended by following attributes:

- `@xmime:ContentType`, mandatory attribute outlining the mime Content-Type of the attachment. Its type is `tsl:NonEmptyString`.
- `@Size`, mandatory attribute of type `xs:positiveInteger` outlining the size of the original Dispatch in bytes before base64-encoding, e.g. useful for implementations to facilitate streaming of such elements.

A.2.3 Element <NormalizedMsg>

Element `<NormalizedMsg>` is the container for carrying a original message in a normalized format between REM-MDs. REM-MDs may choose different formats to present the information to their users.

```
<xs:element name="NormalizedMsg">
    <xs:complexType>
        <xs:sequence>
            <xs:element ref="remsoap:Informational" minOccurs="0"/>
            <xs:element name="Text" minOccurs="0" maxOccurs="unbounded">
                <xs:complexType>
                    <xs:simpleContent>
                        <xs:extension base="xs:string">
                            <xs:attribute name="format" use="required">
                                <xs:simpleType>
                                    <xs:restriction base="xs:string">
                                        <xs:enumeration value="text"/>
                                        <xs:enumeration value="html"/>
                                    </xs:restriction>
                                </xs:simpleType>
                            </xs:attribute>
                        </xs:extension>
                    </xs:simpleContent>
                </xs:complexType>
            </xs:element>
            <xs:element ref="xades:Any" minOccurs="0" maxOccurs="unbounded"/>
            <xs:element ref="remsoap:Attachment" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
```

A.2.3.1 Element <Informational>

```
<xs:element name="Informational">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="Subject" type="xs:string" minOccurs="0">
                <xs:annotation>
                    <xs:documentation>Message subject text</xs:documentation>
                </xs:annotation>
            </xs:element>
            <xs:element name="Comments" type="xs:string" minOccurs="0">
                <xs:annotation>
                    <xs:documentation>Comments like "message correlates to" text</xs:documentation>
                </xs:annotation>
            </xs:element>
            <xs:element name="Keywords" type="remsoap:KeywordType" minOccurs="0"
maxOccurs="unbounded">
                <xs:annotation>
                    <xs:documentation>keyword, sep. bei comma</xs:documentation>
                </xs:annotation>
            </xs:element>
        </xs:sequence>
    </xs:complexType>
</xs:element>
```

Element `<Informational>` provides a structure to carry meta information about the message content, which are:

A.2.3.1.1 Element `<Subject>`

Subject (aka "about") of the message, optional element of type `xs:string`.

A.2.3.1.2 Element `<Comments>`

Additional comments concerning the message (see RFC 5322 [11]), optional element of type `xs:string`.

A.2.3.1.3 Element <Keywords>

RFC 5322 [11] and predecessors define the "keyword" tag, a string of comma separated values, which may be used to assert certain classificatory information on header level for internet messages. The present document provides for a tag with the same purpose, which can be useful, for instance, to assert the payload to a certain business process and outline the type of document carried in a message. (consider, e.g. PEPPOL ProcessType and DocumentType).

Interoperable exchange of classifications should always rely on according agreement or specification of terms used for category assignment and their semantics. To be able to carry keywords and their context information in a generic, extensible manner, following type is defined:

```
<xs:complexType name="KeywordType">
        <xs:simpleContent>
            <xs:extension base="tsl:NonEmptyString">
                <xs:attribute name="scheme" type="tsl:NonEmptyString"/>
                <xs:attribute name="meaning"/>
            </xs:extension>
        </xs:simpleContent>
    </xs:complexType>
```

@meaning is a mandatory attribute expresses the meaning of a keyword. @meaning **may** carry any xs:string. Mandatory attribute @scheme allows implementations to assign a scheme to the value of keywords**,** its type is tsl:NonEmptyString.

A.2.3.2 Element <Text>

The <Normalized Msg> element may contain as many <Text> elements carrying the textual parts, as derived from the original message (see RFC 5322 [11]), its type is xs:string. If present, this element **shall** carry an attribute format of type xs:string with possible values "text" or "html", indicating the format of the <Text> element content.

A.2.3.3 Element <xades:Any>

Optional extensibility elements according to XML schema of type xades:AnyType. These optional container elements are intended to carry structured, XML-formatted REMDispatch parts (if not seen as attachments).

A.2.3.4 Element <remsoap:Attachment>

REMDispatch messages must able to deal with attachments of any format. The following type is defined for this purpose, able to carry some attribute about a specific attachment and either the encoded attachment itself or a pointer to the according MIME boundary of the original message - which in this case **shall** be carried along with the converted normalized format.

```
<xs:complexType name="AttachmentType">
    <xs:choice>
        <xs:element name="Content-ID-Ref" type="xs:string"/>
        <xs:element name="Embedded" type="xs:base64Binary"/>
    </xs:choice>

    <xs:attribute name="Id" type="xs:ID"/>
    <xs:attribute name="Size" type="xs:positiveInteger" use="required"> </xs:attribute>

    <xs:attribute ref="xmime:contentType" use="required"/>
    <xs:attribute name="Filename" use="required">
        <xs:simpleType>
            <xs:restriction base="xs:string">
                <xs:minLength value="1"/>
            </xs:restriction>
        </xs:simpleType>
    </xs:attribute>
    <xs:attribute name="Content_Description">
        <xs:simpleType>
            <xs:restriction base="xs:string">
                <xs:minLength value="1"/>
            </xs:restriction>
        </xs:simpleType>
    </xs:attribute>

    <xs:attribute name="Encoding">
        <xs:simpleType>
            <xs:restriction base="xs:string">
                <xs:length value="1"/>
            </xs:restriction>
        </xs:simpleType>
    </xs:attribute>

    <xs:attribute ref="xml:lang"/>

</xs:complexType>
```

`<remsoap:AttachmentType>` is a `xs:choice` of either:

- `/remsoap:Embedded` Element carrying an attachment in encoded format, type `xs:base64Binary`; or
- `/remsoap:Content-ID-Ref` Element of type `xs:string` carrying the MIME boundary value of the attachment in the original message.

The following attributes are defined for the `<remsoap:AttachmentType>`**:**

- `@Id` - optional attribute of type `xs:ID` to be used for referencing purposes.
- `@Size` - this mandatory attribute of type `xs:positiveInteger` outlines the size of an attachment in kilobytes, e.g. useful to facilitate streaming of such elements.
- `@xmime:contentType` - mandatory attribute outlining the MIME Content-Type of the attachment. Type derived from `xs:string`.
- `@Content_Description` - optional attribute of type `xs:string` outlining the "intent" of the attachment (e.g. "application form", "statement of claim", "invoice").
- `@Encoding` - optional attribute of type `xs:string` outlining the initial mime: Content-Transfer-Encoding of the attachment.
- `@xml:lang` - optional attribute `xml:lang` outlining the language used in the attachment.

A.2.4 Element <REMMDEvidenceList>

This optional element is a container of a sequence of REM Evidence, which may be carried along with the REMDispatch. See TS 102 640-2 [2] for formats and semantics. If this element is present, it **shall** contain at least one REM Evidence.

```
<xs:element name="REMMDEvidenceList" type="remsoap:REMMDEvidenceListType"/>

<xs:complexType name="REMMDEvidenceListType">
    <xs:sequence maxOccurs="unbounded">
        <xs:element ref="rem:SubmissionAcceptanceRejection" minOccurs="0"/>
        <xs:element ref="rem:RelayREMMDAcceptanceRejection" minOccurs="0"/>
        <xs:element ref="rem:RelayREMMDFailure" minOccurs="0"/>
        <xs:element ref="rem:DeliveryNonDeliveryToRecipient" minOccurs="0"/>
        <xs:element ref="rem:RetrievalNonRetrievalByRecipient" minOccurs="0"/>
        <xs:element ref="rem:AcceptanceRejectionByRecipient" minOccurs="0"/>
        <xs:element ref="rem:DownloadNonDownloadByRecipient" minOccurs="0"/>
        <xs:element ref="rem:RelayToNonREMSystem" minOccurs="0"/>
        <xs:element ref="rem:ReceivedFromNonREMSystem" minOccurs="0"/>
    </xs:sequence>
    <xs:attribute name="Id" type="xs:ID" use="required"/>
</xs:complexType>
```

The element <REMEvidenceList> element **shall** be provided with an unambiguous @Id attribute value for referencing purposes.

A.2.5 Element <ds:Signature>

The entire <REMDispatch> **should** be signed by the generating REM-MD instance. The signature **shall** be an enveloped signature covering all sub-elements of <REMDispatch>. For details concerning the <ds:Signature> element the guidelines given in TS 102 640-2 [2] apply.

A.3 Element <REMMDMessage> details

Below follows the schema definition for the data type:

```
<xs:element name="REMMDMessage" type="remsoap:REMMDMessageType"/>
<xs:complexType name="REMMDMessageType">
        <xs:sequence>
            <xs:element ref="remsoap:REMMDEvidenceList"/>
            <xs:element ref="ds:Signature" minOccurs="0"/>
        </xs:sequence>
        <xs:attribute name="Id" type="xs:ID"/>
</xs:complexType>
```

Details of the sequence elements are described above in clause A.2.

A <REMMDMessage> **shall** at least contain one element of type rem:EvidenceType in the sequence of <remsoap:REMMDEvidenceList>.

The entire <REMMDMessage> **should** be signed by the generating REM-MD instance. For details concerning the <ds:Signature> element the guidelines given in TS 102 640-2 [2] apply.

Annex B (normative): WS Addressing specification

The following WS-Addressing headers **shall** be used. WS-Addressing headers not mentioned below are not used and **shall** be omitted:

```
wsa:To
wsa:ReplyTo
wsa:Action
wsa:MessageID
wsa:RelatesTo
```

B.1 Element <wsa:To>

The message destination REM-MD URI **shall** be exposed in this SOAP header element which **shall** be provided exactly once.

B.2 Element <wsa:ReplyTo>

A REM Dispatch or REM-MD Message **shall** carry one SOAP header element `<wsa:Reply>` of type `wsa:EndpointReferenceType`. If present, the source REM-MD URI **shall** be given in `/wsa:ReplyTo/wsa:EndpointReference/wsa:Address`; other optional sub-element or attributes defined for wsa:EndpointReferenceType **shall not** be provided. SOAP faults to be delivered in the network backchannel **should not** carry this header element.

B.3 Element <wsa:Action>

This mandatory element of type `xs:anyURI` denotes the type of the message (REM Dispatch, Evidence only or SOAP processing error) and **shall** carry one of the values outlined in table B.1. A message **shall** carry exactly one `/wsa:Action` SOAP header element.

Table B.1: Defined URIs for the WS Addressing Action element

wsa:Action URIs assigned to Message Types
http://uri.etsi.org/02640/v1#/transport/messageTypes/REMDispatch
http://uri.etsi.org/02640/v1#/transport/messageTypes/REMMDMessage
wsa:Action SOAP error URIs
http://www.w3.org/2005/08/addressing/fault
http://www.w3.org/2005/08/addressing/soap/fault

If this header element has not one of the values above or as defined by WS ReliableMessaging [13], the message **shall** be discarded and the destination REM-MD **shall** send back an Evidence to the source REM-MD with following details.

Table B.2: REM-Event in case of WS-Addressing fault

RelayToREMMDAcceptanceRejection	
Element	**Possible / mandatory values**
EventCode	http:uri.etsi.org/REM/Event#Rejection
EventReasons	*with child elements:*
../EventReason	http:uri.etsi.org/REM/EventReason #InvalidMessageFormat
../EventReason	*and child elements:*
../../Code	http://www.w3.org/2005/08/addressing/ fault
../../Details	Invalid action URI

B.4 Element <wsa:MessageID>

This mandatory element of type xs:anyURI **shall** carry a unique message ID (UUID) according to IETF RFC "A Universally Unique Identifier (UUID) URN Namespace" [10] preceded by the string "uuid:" To ensure uniqueness across domains, this value **shall** be followed by a concatenation of "@", domainlabel, ".", toplabel of the message originating REM-MD (see RFC 3986 [9]). A message **shall** carry exactly one /wsa:MessageID SOAP header element. It **shall** be generated by the source REM-MD respective destination REM-MD in case of a synchronous SOAP fault response.

B.5 Element <wsa:RelatesTo>

These optional element of type <xs:anyURI> **shall** be included, if a message is a SOAP fault message generated by the destination REM-MD while processing an incoming message. In this case, it **shall** carry the value of the <wsa:MessageID> SOAP header of the incoming message.

Annex C (normative): Web Service specification

The REM MD SOAP Service description in this annex follows the WSDL 1.1 notation.

C.1 AcceptREMDispatchOperation Element

```
<wsdl:operation name="AcceptREMDispatchOperation">
    <wsdl:input name="AcceptDispatchRequest" message="tns:REMDispatch"/>
    <wsdl:output name="ResponseToDispatch" message="tns:REMMDMessage"/>
</wsdl:operation>
```

Input is defined as a `<remsoap:REMDispatch>`

```
<wsdl:message name="DispatchRequest">
    <wsdl:part name="Dispatch" element="remsoap:REMDispatch" />
</wsdl:message>
```

Output is defined as a `<remsoap:REMMDMessage>`

```
<wsdl:message name="REMMDMessage">
    <wsdl:part name="Evidence" element="remsoap:REMMDMessage"/>
</wsdl:message>
```

C.2 AcceptREMMDMessageOperation Element

```
<wsdl:operation name="AcceptREMMDMessageOperation">
    <wsdl:input name="AcceptEvidenceRequest" message="tns:REMMDMessage"/>
</wsdl:operation>
```

This is a one-way operation. Input is defined as a `<remsoap:REMMDMessage>`

C.3 REM MD SOAP Service WSDL template

The listing below is intended to be used as a template for specific instances of REM MD SOAP Services. Entries in bold italic must to be replaced by values to be defined for a concrete instance.

NOTE 1: The address of a concrete service entry point shall follow the rule <domain-name>/REMMD/soapentry. The assumption is, a REM MD SOAP instance is bound to the DNS, thus a DNS NSLOOKUP on base of <domain-name> will deliver the IP address of the service instance. The fixed local service entry part /REMMD/soapentry will assure a standard location for the exposure of wsdl and metadata file of the service instances.

```
<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions xmlns:tns="http://uri.etsi.org/02640/remmdsoaptemplate"
xmlns:wsam="http://www.w3.org/2007/05/addressing/metadata"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap12/"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:remsoap="http://uri.etsi.org/02640/soapbinding/v1#"
xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy" xmlns:wsu="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd"
xmlns:sp="http://schemas.xmlsoap.org/ws/2005/07/securitypolicy"
xmlns:sc="http://schemas.sun.com/2006/03/wss/server" xmlns:wsrmp="http://docs.oasis-open.org/ws-
rx/wsrmp/200702"
xmlns:wspmtom="http://schemas.xmlsoap.org/ws/2004/09/policy/optimizedmimeserialization"
name="REMMD_SOAP_Template" targetNamespace="http://uri.etsi.org/02640/remmdsoaptemplate">
    <wsdl:types>
        <xs:schema targetNamespace="http://uri.etsi.org/02640/remmdsoaptemplate">
            <xs:import namespace="http://uri.etsi.org/02640/soapbinding/v1#"
schemaLocation="REM_Schema.xsd"/>
        </xs:schema>
    </wsdl:types>
    <wsdl:message name="REMDispatch">
        <wsdl:part name="Dispatch" element="remsoap:REMDispatch"/>
    </wsdl:message>
    <wsdl:message name="REMMDMessage">
        <wsdl:part name="Evidence" element="remsoap:REMMDMessage"/>
    </wsdl:message>
    <wsdl:portType name="InterREMMDPortType">
        <wsdl:operation name="AcceptREMDispatchOperation">
            <wsdl:input name="AcceptDispatchRequest" message="tns:REMDispatch"/>
            <wsdl:output name="ResponseToDispatch" message="tns:REMMDMessage"/>
        </wsdl:operation>
        <wsdl:operation name="AcceptREMMDMessageOperation">
            <wsdl:input name="AcceptEvidenceRequest" message="tns:REMMDMessage"/>
        </wsdl:operation>
    </wsdl:portType>
    <wsdl:binding name="StandardBinding" type="tns:InterREMMDPortType">
        <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
        <wsp:PolicyReference URI="#TransportBindingPolicy"/>
        <wsdl:operation name="AcceptREMDispatchOperation">
            <soap:operation soapAction="urn:#AcceptREMDispatchOperation" style="document"/>
            <wsdl:input>
                <soap:body parts="Dispatch" use="literal"/>
            </wsdl:input>
            <wsdl:output>
                <soap:body parts="Evidence" use="literal"/>
            </wsdl:output>
        </wsdl:operation>
        <wsdl:operation name="AcceptREMMDMessageOperation">
            <soap:operation soapAction="urn:#AcceptREMMDMessageOperation" style="document"/>
            <wsdl:input>
                <soap:body parts="Evidence" use="literal"/>
            </wsdl:input>
        </wsdl:operation>
    </wsdl:binding>
    <wsdl:service name="REMMDSOAPService">
        <wsdl:port name="InterREMMDPort" binding="tns:StandardBinding">
            <soap:address location="https://localhost:8444/REMMD/soapentry?wsdl"/>
        </wsdl:port>
    </wsdl:service>
    <!-- Policy for https-binding, ws-rm, mtom and REMMD signature certificate (endording token)
used to sign wsu-timestamp (implicitly done by WS-Stack implementation, e.g. Metro) -->
    <wsp:Policy wsu:Id="TransportBindingPolicy">
        <wsp:ExactlyOne>
            <wsp:All>
                <wsam:Addressing wsp:Optional="false"/>
                <wspmtom:OptimizedMimeSerialization/>
                <wsrmp:RMAssertion>
                    <wsp:Policy>
                        <wsrmp:SequenceTransportSecurity/>
                        <wsrmp:DeliveryAssurance>
                            <wsp:Policy>
                                <wsrmp:ExactlyOnce/>
                            </wsp:Policy>
                        </wsrmp:DeliveryAssurance>
                    </wsp:Policy>
                </wsrmp:RMAssertion>
                <sp:TransportBinding
xmlns:sp="http://schemas.xmlsoap.org/ws/2005/07/securitypolicy">
```

```
                    <wsp:Policy>
                        <sp:TransportToken>
                            <wsp:Policy>
                                <sp:HttpsToken RequireClientCertificate="false"/>
                            </wsp:Policy>
                        </sp:TransportToken>
                        <sp:AlgorithmSuite>
                            <wsp:Policy>
                                <sp:Basic256Sha256/>
                            </wsp:Policy>
                        </sp:AlgorithmSuite>
                        <sp:Layout>
                            <wsp:Policy>
                                <sp:Strict/>
                            </wsp:Policy>
                        </sp:Layout>
                        <sp:IncludeTimestamp/>
                    </wsp:Policy>
                </sp:TransportBinding>
                <sp:EndorsingSupportingTokens>
                    <wsp:Policy>
                        <sp:X509Token sp:IncludeToken="http://docs.oasis-open.org/ws-sx/ws-
securitypolicy/200702/IncludeToken/AlwaysToRecipient">
                            <wsp:Policy>
                                <sp:WssX509V3Token10/>
                                <sp:RequireIssuerSerialReference/>
                            </wsp:Policy>
                        </sp:X509Token>
                    </wsp:Policy>
                </sp:EndorsingSupportingTokens>
                <!-- Example from Metro implementation - CallbackHandler to access the REM MD
certificate
                <sc:CallbackHandlerConfiguration>
                    <sc:CallbackHandler name="xwssCallbackHandler"
classname="org.etsi.ts02640.callbackhandler.REMMDCallbackHandler"/>
                </sc:CallbackHandlerConfiguration-->
            </wsp:All>
        </wsp:ExactlyOne>
    </wsp:Policy>
    <!-- Enveloped signature, to protect the whole wsdl instance -->
    <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
schemaLocation="http://www.w3.org/TR/2002/REC-xmldsig-core-20020212/xmldsig-core-schema.xsd"/>
</wsdl:definitions>
```

NOTE 2: A WSDL file of a REM MD Service instances **should** contain an enveloped signature element. The X509 certificate used for applying the signature should be the one outline in the according TSL entry <ServiceDigitalIdentity>. Signature guidelines given in TS 102 640-2 [2] apply.

Annex D (informative): Bibliography

ETSI TS 101 903: "Electronic Signatures and Infrastructures (ESI); XML Advanced Electronic Signatures (XAdES)".

ETSI TS 102 176-1: "Electronic Signatures and Infrastructures (ESI); Algorithms and Parameters for Secure Electronic Signatures; Part 1: Hash functions and asymmetric algorithms".

ETSI TS 102 231 (V3.1.2): "Electronic Signatures and Infrastructures (ESI); Provision of harmonized Trust-service status information".

ETSI TS 102 904: "Electronic Signatures and Infrastructures; Profiles of XML Advanced Electronic Signatures based on TS 101 903 (XAdES)".

IETF RFC 2119: "Key words for use in RFCs to Indicate Requirement Levels; (last visited on 08th May 2010)".

NOTE: Available at http://tools.ietf.org/html/rfc2119.

IETF RFC 4051 (2005): "Additional XML Security Uniform Resource Identifiers".

NOTE: Available at http://www.ietf.org/rfc/rfc4051.txt.

OASIS Standard Specification Web Services Security SAML Token Profile 1.1, incorporating Approved Errata, 1 November 2006.

NOTE: Available at http://docs.oasis-open.org/wss/v1.1/wss-v1.1-spec-errata-os-SAMLTokenProfile.pdf.

OASIS Standard Specification Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.0, 15 March 2005.

NOTE: Available at http://docs.oasis-open.org/security/saml/v2.0/saml-core-2.0-os.pdf.

OASIS Standard Specification Web Services Security X.509 Certificate Token Profile 1.1, incorporating Approved Errata, 1 November 2006.

NOTE: Available at http://docs.oasis-open.org/wss/v1.1/wss-v1.1-spec-errata-os-x509TokenProfile.pdf.

OASIS Standard Specification WS-SecurityPolicy 1.2, 30 April 2007.

NOTE: Available at http://docs.oasis-open.org/ws-sx/ws-securitypolicy/v1.2/ws-securitypolicy.pdf.

W3C Recommendation XML Signature Syntax and Processing (Second Edition).

NOTE: Available at http://www.w3.org/TR/xmldsig-core/.

XML Signature Syntax and Processing Version 1.1, W3C Candidate Recommendation 03 March 2011.

NOTE: Available at http://www.w3.org/TR/2010/WD-xmldsig-core1/.

W3C Recommendation Web Services Addressing 1.0.

NOTE: Available at http://www.w3.org/TR/ws-addr-core/.

W3C Extensible Markup Language (XML) 1.0 (Fifth Edition), W3C Recommendation 26 November 2008.

NOTE: Available at http://www.w3.org/TR/xml/.

Directive 1999/93/EC of the European Parliament and of the Council of 13 December 1999 on a Community framework for electronic signatures.

AFNOR AC Z74-600-3 (2005): "Electronic attestations of anteriority, deposit, withdrawal and receipt - Part 3: format of attestations".

ETSI TS 101 733: "Electronic Signatures and Infrastructures (ESI); CMS Advanced Electronic Signatures (CAdES)".

ETSI TS 102 778 (Parts 1 to 5): "Electronic Signatures and Infrastructures (ESI); Electronic Signatures and Infrastructures (ESI); PDF Advanced Electronic Signature Profiles".

ETSI TR 102 605: "Electronic Signatures and Infrastructures (ESI); Registered E-Mail".

IETF RFC 2046: "Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types".

NOTE: Available at http://tools.ietf.org/html/rfc2046.

IETF RFC 5751: "Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 3.2 Message Specification".

IETF RFC 5652: "Cryptographic Message Syntax (CMS)".

OSCI-Transport - Version 2.0, Edition 4 - Web Services Profiling and Extensions Specification, OSCI Steering Office 2010.

NOTE: Available at http://www.xoev.de/sixcms/media.php/13/OSCI20_WS-ProfilingAndExtensionSpecification_Edition4.pdf.

SPOCS Project, D3.1: "Assessment of eDelivery systems and specifications required for interoperability".

NOTE: Available at http://www.eu-spocs.eu/index.php?option=com_processes&task=showDocument&did=198&id=18&Itemid=1.

STORK D2.3 - STORK Quality authenticator scheme, 2009-03-03.

NOTE: Available at https://www.eid-stork.eu/index.php?option=com_processes&Itemid=&act=streamDocument&did=577.

STORK D5.1.8.b - Interface Specification, 31/7/2009.

NOTE: Available at https://www.eid-stork.eu/index.php?option=com-processes&Itemid=&act=streamDocument&did=960.

W3C Recommendation XML-binary Optimized Packaging, 25 January 2005.

NOTE: Available at http://www.w3.org/TR/xop10/.

W3C Working Group Note Describing Media Content of Binary Data in XML, 5 May 2005.

NOTE: Available at http://www.w3.org/TR/xml-media-types/.

History

Document history		
V1.1.1	September 2011	Publication