COMPETITIVENESS AND INNOVATION FRAMEWORK PROGRAMME

ICT Policy Support Programme (ICT PSP)

Preparing the implementation of the Services Directive

ICT PSP call identifier: ICT PSP-2008-2
ICT PSP main Theme identifier: CIP-ICT-PSP.2008.1.1

Project acronym: SPOCS

Project full title: Simple Procedures Online for Cross-border Services
Grant agreement no.: 238935

Specifications for interoperable access to eDelivery and eSafe systems

Appendix 3: eDelivery Interconnect Protocol and Gateway detailed Specification

- Corrigenda 2.0 March 2012 based on Corrigenda 1.2 -

Deliverable Id :	D3.2
Deliverable Name :	Specifications for interoperable access to eDelivery and eSafe systems
Status :	Corrigenda 2.0
Dissemination Level :	SPOCS internal and EU Commission
Due date of deliverable :	30th September 2010
Actual submission date :	
Work Package :	WP3: Interoperable delivery, eSafe, secure and interoperable exchanges and acknowledgement of receipt
Organisation name of lead contractor for this deliverable :	**BVA**
Author(s):	Jörg Apitzsch (DE FHB), Luca Boldrin (IT InfoCert), Arne Tauber (SPOCS AT)
Partner(s) contributing :	SPOCS.AT, DE BVA, DE Siemens, GR MINT, IT InfoCamere, NL MINEZ, PL ILIM

Abstract: This document Appendix 3 is part of the second deliverable in work package 3 of the EU co-funded project SPOCS. It describes specifications for the interoperability layer to connect existing eDelivery solutions based on a common eSecurity architecture. Based on the specifications open modules have been developed.
This revision 2.0 is a formal adjustment of the sections related to eDelivery, respecting the fact that major concept details meanwhile are adopted by the latest version of the specification ETSI TS 102 640 on “Registered E-Mail”.

History

Version	*Date*	*Modification reason*	*Modified by*
0.9	31.8.10	Aligned and finalised; ready for QA	Luca Boldrin Jörg Apitzsch
1.0	30.9.10	No modification regarding precedor submitted to and approved by EC	
1.1	26.2.11	Corrigenda according findings during implementation phase (11.2.11); review & QA 26.2.11	Jörg Apitzsch
1.2	21.7.11 – 11.8.11	Corrigenda according further findings during implementation and testing phases until June 2011: smaller schema detailing. Section 1.1 (SOAP faults) detailed	Jörg Apitzsch Review Unisystems S.A.
2.0	26.03.12	eDelivery adjustments with respect to latest revision of ETSI TS 102 640	Jörg Apitzsch

Table of Contents

List of Tables

List of Figures

List of Listings

List of Abbreviations

Abbreviation	Explanation
DNS	Domain Name System
EC	European Commission
eID	Electronic Identity
GW	(eDelivery) Gateway
IdP	Identity Provider
LSP	Large Scale Pilot
MD	(eDelivery) Management Domain
MS	Member State
PEPPOL	Pan-European Public Procurement Online
REM	Registered E-Mail
SD	Services Directive
SP	Service Provider
SPOCS	Simple Procedures Online for Cross-border Services
SSL	Secure Socket Layer
STORK	Secure Identity Across Borders Linked
TLS	Transport Layer Security
TSL	Trust-service Status List
TSP	Trusted Service Provider
WP<n>	Work Package (number n)
WSDL	Web Service Description Language

Further abbreviations used in this document are explained on first occurrence.

Document Structure of SPOCS D3.2

SPOCS deliverable D3.2 "Specifications for interoperable access to eDelivery and eSafe systems" consists of several documents.

Main part gives a complete description of the general context, functionality of solutions provided, their architectural details and covered security and trust establishment features.

Additional documents are provided for detailed technical specifications of the buildings blocks, considered security architecture modelling and development baselines and according operational policies.

The main document:

- SPOCS D3.2 Functional Specification, Architecture and Trust Model**Fehler! Verweisquelle konnte nicht gefunden werden.**

is accomplished by following separated appendix documents:

- Appendix 1: Security Architecture Development Process**Fehler! Verweisquelle konnte nicht gefunden werden.**
- Appendix 2: Trust-service Status List Profiling ("SPOCS TSL")**Fehler! Verweisquelle konnte nicht gefunden werden.**
- Appendix 3: eDelivery Interconnect Protocol and Gateway Specification**Fehler! Verweisquelle konnte nicht gefunden werden.**
- Appendix 4: eSafe – Operations in Detail**Fehler! Verweisquelle konnte nicht gefunden werden.**
- Appendix 5: SPOCS TSL Accreditation and Operation Policy
- Appendix 6: Security Model**Fehler! Verweisquelle konnte nicht gefunden werden.Fehler! Verweisquelle konnte nicht gefunden werden.**.

This document a corrigendum Appendix 3: eDelivery Interconnect Protocol and Gateway Specification**Fehler! Verweisquelle konnte nicht gefunden werden.**, part of the second deliverable of SPOCS WP3.

Referenced XML Namespaces

Prefix	XML Namespace	Reference
ds	http://www.w3.org/2000/09/xmldsig#	[3]
rem	http://uri.etsi.org/02640/v2#	[12]
remsoap	http://uri.etsi.org/02640/soapbinding/v1#	[46]
s12	http://www.w3.org/2003/05/soap-envelope	[14]
saml2	urn:oasis:names:tc:SAML:2.0:assertion	[15]
tsl	http://uri.etsi.org/02231/v2#	[9]
wsa	http://www.w3.org/2005/08/addressing	[5]
wsdl	http://www.w3.org/ns/wsdl	[52]
wsp	http://www.w3.org/ns/ws-policy	[50]
wsrm	http://docs.oasis-open.org/ws-rx/wsrm/200702	[48]
wsrmp	http://docs.oasis-open.org/ws-rx/wsrmp/200702	[49]
wsse	http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd	[6]
wsu	http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd	[6]
xs	http://www.w3.org/2001/XMLSchema	[20]

Table 1: Referenced Namespaces

eDelivery Interconnect Protocol and Gateway detailed Specificiation

As outlined in **Fehler! Verweisquelle konnte nicht gefunden werden.**, sections 2.3.3 and even more detailed in 3, the Interconnect Protocol is based on SOAP version 1.2. Profiling of SOAP and incorporated WS-* specifications and details of the Interconnect Protocol are specified in this appendix.

A base assumption about eDelivery solutions to be interconnected is that all of them are able to provide evidences about the status of the Dispatch delivery – that is about malfunctions on the message route as well as the fact that a Dispatch successfully was made available for the Recipient. Thus, these solutions are seen as basically functional conformant to the principles of the REM-specification TS 102 640 part 4 (see **Fehler! Verweisquelle konnte nicht gefunden werden.**), even though there are differences concerning the degree of conformance fulfilment.

Note on notation:

> For those parts of this appendix where referenced specifications are profiled, normative statements of requirements are presented in the following manner:
>
> Rnnnn - Statement text here
>
> where "nnnn" is replaced by a number that is unique among the requirements in this specification, thereby forming a unique requirement identifier.

1 SOAP Version, Transport and Fault Binding

R0010 - **Gateways** (GW) MUST support SOAP Version 1.2 according to **Fehler! Verweisquelle konnte nicht gefunden werden.** and constraints specified in WSI-Basic **Fehler! Verweisquelle konnte nicht gefunden werden.**, section Messaging with restriction R0020.

R0020 - The SOAP Message Transmission Optimization Mechanism **Fehler! Verweisquelle konnte nicht gefunden werden.** MUST be supported by conformant implementations.

R0030 - Transport binding is restricted to HTTP/1.1 **Fehler! Verweisquelle konnte nicht gefunden werden.**, which has performance advantages and is more clearly specified than HTTP/1.0. R1140 of WSI-Basic **Fehler! Verweisquelle konnte nicht gefunden werden.** (A MESSAGE SHOULD be sent using HTTP/1.1) – is superseded: A message MUST be sent using HTTP/1.1.

R0040 - HTTP/1.1 MUST be used with TLS mechanisms (HTTPS) according to **Fehler! Verweisquelle konnte nicht gefunden werden.**.

R0050 - Errors occurring while processing a SOAP message use the SOAP fault mechanisms. The SOAP fault block according to **Fehler! Verweisquelle konnte nicht gefunden werden.** MUST be used to report information about errors. The `s12:Fault` element MUST be carried in the SOAP body block of the network backchannel SOAP response message.

Such fault situations MUST be converted to appropriate REM-MD Evidences as defined by in **Fehler! Verweisquelle konnte nicht gefunden werden.** by SOAP message source Gateways. This is defined in details case by case below.

As specifications incorporated here in general define their own fault handling, this document only outlines additional fault situations specific to GW-GW communication.

R0060 - Gateways are stateless concerning the Transport Data.

Following messages MUST be delivered synchronously as response in the network backchannel:

- SOAP faults
- Evidence message signalling the acceptance or non-acceptance of a Dispatch message by the destination Gateway. Other evidences that can immediately be provided by the destination Gateway may also be delivered synchronously in the network backchannel.

Any other SOAP message MUST be delivered by opening a new https connection (asynchronous message exchange even for Dispatch respective Evidence type messages to be seen as response to a foregoing Dispatch message).

Following information for the sub elements of `s12:Fault` is supplied per fault described in this document:

Sub element	Possible / mandatory values
`../Code/Value`	as defined in SOAP12 **Fehler! Verweisquelle konnte nicht gefunden werden.**, 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 GW raising the fault

Table 2: SOAP fault subelements

In the fault message itself, the [Code] value MUST have a namespace prefix of **`s12:`**, the [Subcode] value namespace prefix MUST be **`remsoap:`**.

The optional **`s12:/Fault/Role`** element can be omitted as not interpreted in the actual version of this specification, which only deals with SOAP nodes in the role "Gateway".

It should be noted that implementations MAY provide second-level **`s12:/Fault/Detail`** fields, but they should be careful not to introduce security vulnerabilities when doing so (e.g. by providing too detailed information).

1.1 General Processing Error

If an unspecific and unrecoverable message processing error occurs on message entry, a fault MUST be generated and the message MUST be discarded.

NOTE: SOAP faults may already be thrown by WS-Stack implementations and send back to the source Gateway in network backchannel, before the specific Gateway implementation part gains control. Those SOAP faults MUST be converted be the source Gateway to an Evidence ***RelayToREMMDAcceptanceRejection*** as described below.

In reaction of a SOAP fault, a source GW MUST generate an Evidence with following details:

RelayToREMMDAcceptanceRejection	
Element	**Possible / mandatory values**
`EventCode`	**`http:uri.etsi.org/REM/Event#Rejection`**
`EventReasons`	*with at least one child element:*
`../EventReason`	*and child elements:*
`../../Code`	**`http:uri.etsi.org/REM/EventReason#R-REMMD_ Malfunction`**
`../../Details`	Concatenation of: *value of* **`s12:/Fault/code,`** “ SOAPfault from: ”, *value of* **`s12:/Fault/node`**[1] , “ text: “ ,

[1] Should be the URL of GW raising the fault

	value(s) of **`s12:/Fault/Text`**[2]
`../EventReason` *	*Optional further elements, if* **`s12:/Fault/Detail`** *elements present:*[3]
`../../Code`	Value of namespace URI **`in s12:/Fault/Detail`**
`../../Details`	Value to be taken from **`s12:/Fault/Detail`**

Table 3: REM-Event in case of technical malfunction

It is assumed, that in cases technical errors are detected under the control of the Gateway implementation part an Evidence ***RelayToREMMDAcceptanceRejection*** can be generated instead of a SOAP fault. This "rejection" Evidence SHOULD describe details of the fault situation in **`rem:EventReason`** according to TS 102 640 part 2. If needed, additional **`rem:EventReason/code`** URIs MAY be defined in customs namespaces[4].

1.2 Fault Delivery, Logging and Escalation

In general, the fault handling defined in SOAP **Fehler! Verweisquelle konnte nicht gefunden werden.**, section 5.4 "SOAP Fault" applies as well as the respective fault handlings defined by the specifications incorporated here. The fault MUST be logged by the node where the fault raises to be available for supervision and revision purposes. If faults appear at the destination Gateway, an according SOAP fault MUST be delivered in http backchannel of the underlying request, if not described different. Message processing MUST be aborted, if not specified otherwise for special situations in this document.

Though, there may exist situations where the possibility to deliver this information to the source Gateway of the underlying message does not exist. In this case, appropriate escalation mechanisms MUST be foreseen by conformant implementations to signal such situations to the system monitoring environment / operating personal; follow-up of this situation is up to the operating policies[5].

[2] Multiple element values must be concatenated

[3] Care must be taken here. Some WS-Stack implementations provide a stack trance in the Details element in XML format. Such information should not be included in the Evidence, better written to the log and handled by means of system supervision functionality.

[4] To be further detailed by specific implementations. Additional code values will be subject of future revision of the according Event reason code list of TS 102 640.

[5] Those should be made available online for all possible communication partners. Details are not addressed by this document.

2 Evidences

REM specifications provide a list of evidence types, to be generated on occurrence of delivery events according to the model described in TS 102 640 part 1.

For the purposes of SPOCS no need for further evidences is presently foreseen, while some evidences are not needed. As the Store&Notify (S&N) mode of operation is not supported between SPOCS Gateways, S&N evidences are not taken into account.

Receiving from non trusted systems is supported in order to account for delivery to those Realms which cannot claim a “REM level” delivery. While this is NOT RECOMMENDED at all, still we are not in the position to exclude that some country rest on a less-than-REM delivery mode (PEPPOL's "BusDox" being one possibility, if such types of Realms shall be connect via BusDox to the SPOCS eDelivery sphere).

Following list of evidences MUST be supported between SPOCS eDelivery Gateways:

1. SubmissionAcceptanceRejection
2. RelayToREMMDAcceptanceRejection
3. RelayToREMMDFailure
4. DeliveryNonDeliveryToRecipient
5. RetrievalNonRetrievalByRecipient
6. AcceptanceRejectionByRecipient
7. ReceivedByNonREMSystem

For the details of generation and validation of evidences the specification TS 102 640 applies.

Gateways MUST express in their according TSL entry, which evidences they are able to provide in response to a Dispatch respective which one are expected together with a Dispatch targeted to them (see **Fehler! Verweisquelle konnte nicht gefunden werden.**).

Listing of events REM-MD evidence types:

Event	Event description	REM-MD Evidence generated	Support
S-REM-MD Acceptance	The Sender’s GW receives a Dispatch from the national eDelivery realm. The GW acknowledges according to local procedures. The GW also produces a REM-MD Evidence to be incorporated in the message forwarded to remote GW. It is up to the Recipient’s GW to decide whether to manage this evidence according to the national eDelivery Realm	**Submission AcceptanceRejection** (for evidence incorporated in the message forwarded to remote GW)	MUST

Event	Event description	REM-MD Evidence generated	Support
	requirements.		
S-REM-MD Rejection	The Sender’s GW receives a Dispatch from the national eDelivery realm. The GW may reject the message, in which case it will respond according to local procedures. No message is forwarded to remote GW	--	--
R-REM-MD Acceptance	The Recipient’s GW receives a Dispatch from Sender’s GW. Acknowledge of acceptance is provided as a REM-MD Evidence returned back to the Sender’s GW (in the SOAP backchannel).	**RelayToREMMDAcceptanceRejection**	MUST
R-REM-MD Rejection	The Recipient’s GW receives a Dispatch from sender’s GW. Acknowledge of rejection is provided as a REM-MD Evidence returned back to the Sender’s GW (in the SOAP backchannel)	**RelayToREMMDAcceptanceRejection**	MUST
Expiration of time to deliver to R-REM-MD	The Sender’s GW cannot forward the Dispatch to the Recipient’s GW after a specified time. The Sender’s GW will notify the Sender according to local procedures.	-- (if national eDelivery were REM, a **RelayToREMMDFailure** evidence would be generated)	MAY
REM Object Delivery	The Recipient’s GW receives from destination MD a notification (according to local procedures) that the Dispatch has been delivered. The Recipient’s GW produces an Evidence back to the Sender’s GW. The Evidence SHOULD be returned in the SOAP backchannel whenever it is possible, otherwise the Evidence will be sent back in a separate message.	**DeliveryNonDeliveryToRecipient**	MUST

Event	Event description	REM-MD Evidence generated	Support
Non delivery within a given retention period	The Recipient's GW receives from destination realm a notification (according to local procedures) that the Dispatch has not been delivered yet OR the rRcipient's GW does not receive from destination MD any delivery notification (according to local procedures) after a defined time. The Recipient's GW produces an Evidence back to the Sender's GW. Evidence will be sent in a separate message	**DeliveryNonDelivery ToRecipient**	MAY
(message box) - Retrieval	The Recipient's GW receives from destination MD a notification (according to local procedures) that the Dispatch has been retrieved by the Recipient. The Recipient's GW produces an Evidence back to the Sender's GW. The Evidence will be sent in a separate message.	**RetrievalNonRetrieval ByRecipient**	MAY

Event	Event description	REM-MD Evidence generated	Support
(message box) - Expiration of time for Retrieval	The Recipient's GW receives from destination MD a notification (according to local procedures) that the Dispatch has not been retrieved yet OR the Recipient's GW does not receive from destination MD any retrieval notification (according to local procedures) after a defined time. The Recipient's GW produces an Evidence back to the Sender's GW. The Evidence will be sent in a separate message.	**RetrievalNonRetrieval ByRecipient**	MAY
(message box) - Retrieval by a REM Recipient's delegate	The recipient's GW Receives from destination MD a notification (according to local procedures) that the Dispatch has been retrieved by a Recipient's delegate. The Recipient's GW produces an Evidence back to the sender's GW. The Evidence will be sent in a separate message.	**RetrievalNonRetrieval ByRecipient**	MAY
Dispatch received from a non–REM system	The Recipient's GW receives a Dispatch from a sender's GW which interfaces a Sender's eDelivery MD, which does not qualify as REM. The recipient's GW produces an evidence of this event for the recipient's realm (if local procedures allow for that).	-- (if national eDelivery where REM, a ***ReceivedFrom-NonREMSystem*** Evidence would be produced)	MAY

Table 4: Evidence overview

2.1 Profiling of Evidence Components

REM evidences are built as a collection of "evidence components", according to the rules described in TS 102 640 part 2.

	Component Class	Id	Component
.REM-MD Evidence	Core Components	G00	REM-MD Evidence Identifier
		G01	REM-MD Evidence Type
		G02	REM Event
		G03	Reason code (see note below) NOTE: Preferably there would be only one (when applicable) G03 listing all remarked exceptions reason codes, but it cannot be excluded that one single message collects more than one G03.
		G04	REM-MD Evidence Version
		G05	Event Time
		G06	Transaction log information
	REM-MD Components	R01	Evidence issuer Policy Identifier
		R02	Evidence issuer Details
		R03	Signature by issuing REM-MD
	Identity Related Components	I00	Sender's details
		I01	Recipient's details
		I02	Recipient's delegate details
		I03	Recipient referred to by the Evidence
		I04	Sender Authentication details
		I05	Recipient Authentication details
	Messaging Components	M00	REM-MD Message/REM Dispatch details
		M01	Reply-to
		M02	Notification Message Tag
		M03	Message Submission Time
		M04	Forwarded to external system
	Extended	Enn	Space for private or public extensions to be added in the future by a set of users or by standardization bodies

Table 5: REM-MD Evidence generic template

The use of Evidence components in the context of SPOCS is constrained as follows:

Component		Value	Note[6]
• Core components			
G00	REM-MD Evidence Identifier	According to TS 102 640	
G01	REM-MD Evidence Type	According to TS 102 640	
G02	REM Event	According to TS 102 640	
G03	Reason code	According to TS 102 640	

[6] See section 5 for definition of message elements referenced here

Component		Value	Note[6]
G04	REM-MD Evidence Version	According to TS 102 640	
G05	Event Time	According to TS 102 640	
G06	Transaction log information	According to TS 102 640	
• **REM-MD Components**			
R01	Evidence issuer Policy Identifier	"http://uri.spocs-eu.eu/eDeliveryPolicy"	Will be changed after project conclusion, depending on the adopted sustainability model.
R02	Evidence issuer Details	According to TS 102 640	Since REM Evidence will only be generated by the Gateway, this field MUST contain the Gateway details. Value of `rem:EntityName` might be, for instance, "PEC_Realm_Gateway"
R03	Signature by issuing REM-MD	--	Not used
• **Identity related Components**			
I00	Sender's details	According to TS 102 640	Sender's details include: • electronic address (MUST) • postal address (MAY) • digital certificate info (MAY) • signature detail (MAY). Since different MD's may adopt different ways for sender identification, only electronic address is mandatory. Value MUST be identical to the one of `rem:AttributedElectronicAddress` of `reamsoap:From` in `remsoap:REMDispatch.`

Component		Value	Note[6]
I01	Recipient's details	According to TS 102 640	Recpient's details include: • electronic address (MUST) • postal address (MAY) • digital certificate info (MAY) • signature detail (MAY). Since different realm may adopt different ways for sender identification, only electronic address is mandatory. Value MUST be indentical to the one of **`remsoap:AttributedElectronicAddress`** of **`remsoap:Recipient`** in **`remsoap:REMDispatch.`** Fehler! Textmarke nicht definiert.
I02	Recipient's delegate details	According to TS 102 640	Only for gateways to realms which implement the delegate concept
I03	Recipient referred to by the Evidence	According to TS 102 640	Value MUST be aligned with body element **`rem:ReplyToAddress`** of **`rem:Evidence.`**[7]
I04	Sender Authentication details	According to TS 102 640	**`rem:AuthenticationMethod`** MUST match the QAA-level in the SAML sender vouches token. See section 4.1.1 for details.
I05	Recipient Authentication details	According to TS 102 640	If such elements are supplied, consider section 4.1.1 for the values of **`rem:AuthenticationMethod.`**
• **Messaging Components**			
M00	REM-MD Message/REMDispatch details	According to TS 102 640	Value of element **`rem:MessageIdentifierByREMMD`** MUST be taken from **`remsoap:REMDispatch/remsoap:MetaData/remsoap:MsgIdentification/remsoap:Message-ID.`** Algorithm used for **`ds:DigestMethod`** MUST be "SHA-256".
M01	Reply-to	Modified regarding to TS 102 640	If present in underlying REMDispatch, value of **`remsoap:REMDispatch/remsoap:MetaData/remsoap:Originators/remso`**

[7] see section 5.4 for details

Component		Value	Note[6]
			ap:Reply-To/remsoap:AttributedElectronicAddress must be supplied in element **rem:ReplyToAddress** of Evidence, the Evidence element **Reply-to** is not used in this profiling. If **remsoap:ReplyTo** is not present in REMDispatch, value of ...**remsoap:From/remsoap:AttributedElectronicAddress** must be taken.
M02	Notification Message Tag	“false”.	Only S&F mode applies – not used, as defaulted to "false" if absent.
M03	Message Submission Time	According to TS 102 640	MUST be taken from original eDelivery MD submission time **remsoap:InitialSend** of **remsoap:MetaData** in **remsoap:REMDispatch.**
M04	Forwarded to external system	According to TS 102 640	
• **Extended**			
Enn	Space for private or public extensions to be added in the future by a set of users or by standardiza-tion bodies		Some information which may fit here: • origin eDelivery MD name • origin eDelivery MD/Realm policy. Actually left open, to be specified in the piloting phase.

Table 6: Evidence components

3 SOAP Header Blocks

3.1 WS Security Header

Most constituents which MUST be used in this header are already described in **Fehler! Verweisquelle konnte nicht gefunden werden.**, section 3.4. Only some additional detailing is outlined here. The entries in **`/wsse:Security`** MUST conform to the WS Security specification **Fehler! Verweisquelle konnte nicht gefunden werden.**.

`/wsse:Security`

> This header block MUST be present, carrying source Gateway authentication information. This is the X509v3 certificate used be the source Gateway to sign the message (and SOAP body block constituents, explained below) and the according **`ds:Signature`** element.
>
> A **`wsu:TimeStamp`** MUST be present in this header.

The WS-Security header block MUST be built up and processed transparently by the generic part of the Gateway implementation to be provided by SPOCS WP3.

3.1.1 Security Token Type

To be extensible, the WS-Security specification has not bound to specific security token types. For the present version of this specification, we restrict to a token of type **`http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-x509-token-profile-1.0#X509v3;`** see **Fehler! Verweisquelle konnte nicht gefunden werden.** for details. This certificate MUST be the one exposed as signature certificate in the according SPOCS TSL entry for the Gateway building up the SOAP message.

3.1.2 Transport Signature

The signature element in the **`/wsse:Security`** header MUST cover the message parts **`wsu:Timestamp`**, WS-Addressing headers according to section [3.2] and **`s12:Body`**.

3.2 Addressing Gateways

For Inter-Gateway addressing, SPOCS eDelivery incorporates the WS-Addressing specification.

R0100 - **Gateways** MUST support WS-Addressing according to **Fehler! Verweisquelle konnte nicht gefunden werden.**. Constraints apply specified in WSI-Basic **Fehler! Verweisquelle konnte nicht gefunden werden.**, section 3.6 "Support for WS-Addressing Messaging" and section 3.7 "Use of WS-Addressing MAPs".

R0110 - **Gateways** MUST support WS-Addressing SOAP Binding according to **Fehler! Verweisquelle konnte nicht gefunden werden.**, whereby only the rules for binding to SOAP 1.2 apply.

Basically, **`wsa:To`** carries the URL of the destination Gateway and **`wsa:ReplyTo/wsa:Address`** outlines the one of the source Gateway of a message. The respective URL values are the ones of the element **`tsl:ServiceSupplyPoint`** of the according Gateway TSL entry.

See section **Fehler! Verweisquelle konnte nicht gefunden werden.** (Section 3.3) for the determination of the destination Gateway address.

The **wsa:MessageID** is the ID of the Inter-Gateway message and must not be confused with the initial ID assigned to the message by the originating MD.

This specification defines following 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 **Fehler! Verweisquelle konnte nicht gefunden werden.**,

R0120 - WS-Addressing headers not mentioned below are not used and MUST be omitted.

```
<wsa:To> xs:anyURI </wsa:To>
<wsa:ReplyTo>
  <wsa:EndpointReference>
    <wsa:Address> xs:anyURI </wsa:Address>
  </wsa:EndpointReference>
</wsa:ReplyTo> ?
<wsa:Action>
http://www.eu-spocs.eu/ws/2010/07/transport/messageTypes/Dispatch
     |
http://www.eu-spocs.eu/ws/2010/07/transport/messageTypes/Evidence
     |
http://www.w3.org/2005/08/addressing/fault |
http://www.w3.org/2005/08/addressing/soap/fault
</wsa:Action>
<wsa:MessageID> xs:anyURI </wsa:MessageID>
<wsa:RelatesTo RelationshipType="xs:anyURI"?> xs:anyURI
</wsa:RelatesTo> ?
```

Listing 1: WS-Addressing headers

/wsa:To

The message destination Gateway URI MUST be exposed in this SOAP header element which MUST be provided exactly once.

/wsa:ReplyTo

A dispatch or evidence message MUST carry one SOAP header element **wsa:Reply** of type **wsa:EndpointReferenceType**. If present, the source Gateway URI MUST be given in **/wsa:ReplyTo/wsa:EndpointReference/wsa:Address**; other optional sub-element or attributes defined for **wsa:EndpointReferenceType** MUST NOT be provided. SOAP faults to be delivered in the network backchannel SHOULD NOT carry this header element.

/wsa:Action

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

wsa:Action URIs assigned to Message Types
http://www.eu-spocs.eu/ws/2010/07/transport/messageTypes/REMDispatch
http://www.eu-spocs.eu/ws/2010/07/transport/messageTypes/REMMDMessage

wsa:Action SOAP error URIs[8]
`http://www.w3.org/2005/08/addressing/fault`
`http://www.w3.org/2005/08/addressing/soap/fault`

Table 7: Defined URIs for the WS Addressing Action element

If this header element has not one of the values above, the message MUST be discarded and the destination Gateway MUST send back an Evidence to the source Gateway with following details:

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

Table 8: REM-Event in case of WS-Addressing fault

`/wsa:MessageID`

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

`/wsa:RelatesTo ?`

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

In case of asynchronous responses (messages of type REMDispatch as well as REMMDMessage (Evidence)) this element MAY be omitted. Message

[8] As specified in **Fehler! Verweisquelle konnte nicht gefunden werden.**, section 6

[9] For details of URI definition see RFC 2396 **Fehler! Verweisquelle konnte nicht gefunden werden.**.

[10] Complete pattern looks like **`uuid:<uuid-value>@<domainlabel>.<toplabel>`**; an example would be **`"uuid:9080F14F-6936-E2FF-0F5D-A022F573129D@edeliverygateway.pec.it"`**

relationship is defined on message metadata level and SHOULD be handled by the domestic eDelivery solutions itself (see section 6).

All WS-Addressing header blocks MUST be built up and processed transparently by the generic part of the Gateway implementation to be provided by SPOCS WP3.

3.3 Reliable Messaging

Network connections may break during a SOAP Request/Response message exchange. The WS Reliable Messaging specification (WS-RM) **Fehler! Verweisquelle konnte nicht gefunden werden.** addresses transparent control of reliable message delivery; mechanisms described in WS-RM MUST be supported.

In detail, Gateways MUST support the “ExactlyOnce Delivery Assurance”, what SHOULD be done by exposing an according WS-RM policy entry **`“wsrmp:RMAssertion/wsp:Policy/wsrmp:DeliveryAssurance/wsp:Policy/wsrmp:ExactlyOnce”`** in the WSDL file of Gateway instances; see **Fehler! Verweisquelle konnte nicht gefunden werden.** for details[11].

[11] More detailed behaviour like timeouts, number of retries etc. have to be configured for WS-RM implementations available. E.g. for the Metro implementation, see http://metro.java.net/guide/Configurable_features_summary.html

4 SAML Token for End Entity Authentication

The sender and recipient authentication token is a SAML assertion. It specifies that this assertion was issued by a certain gateway about the sender or recipient at a certain point in time and attests the respective end entities identity provided meets specified conditions ("sender vouches" token as defined in SAML specification **Fehler! Verweisquelle konnte nicht gefunden werden.**).

ETSI TS 102 640 part 2 **Fehler! Verweisquelle konnte nicht gefunden werden.** foresees elements for sender and recipient authentication details, based on the type **`rem:AuthenticationDetailsType`**. The SAML token described here MUST be inserted in the extension element **`rem:AdditionalDetails`** (type: **`xades;AnyType`**) of these elements. Other items of **`rem:Sender-rem:recipientAuthenticationDetails`** MUST be provided according to ETSI TS 102 640 part 2.

To expose sender's authentication details to the message destination MD, a REMDispatch MUST always be delivered in combination with the according Evidence **`rem:SubmissionAcceptanceRejection`**, which has to be generated by the source Gateway.[12]

The SAML token MUST be built up on base of attributes[13] about the token subject (required from the source MD) and processed transparently by the generic part of the Gateway implementation to be provided by SPOCS WP3. On the destination side Gateway, functionality MUST be provided to access the token validation result in a reliable manner as well as access to the token itself.

The definition of this token is aligned to and in many parts conformant with the specification of the SAML Assertion defined in the STORK D5.8.1b Interface Specification **Fehler! Verweisquelle konnte nicht gefunden werden.**. A major difference is the introduction of a new assertion attribute. Besides the QAA authentication level defined by STORK, a SPOCS sending Gateway may also provide a sender's QAA registration level.

4.1 SAML Assertion Profiling

In this document only those elements are detailed that differ from the STORK interface specification. Format details and semantics are further described in the OASIS SAML V2.0 core specification **Fehler! Verweisquelle konnte nicht gefunden werden.**.

Child elements of **`saml2:Assertion`**:

`/saml2:Issuer`

> This mandatory element MUST contain the URI that identifies the issuing gateway. This URI must be identical to the URI value defined within the SPOCS TSL (**`tsl:ServiceSupplyPoint`**, see **Fehler! Verweisquelle konnte nicht gefunden werden.**.

`/ds:Signature`

> Mandatory element; in order to use the SAML assertion as transferable token in other contexts, the assertion must be signed by the issuing GW. An XML

[12] This behaviour is defined as optional in the actual version of TS 102 640.

[13] Detailed parameters which MUST/SHOULD/MAY be provided will be defined and evaluated in the implementation phase.

Signature authenticates the issuing GW and ensures message integrity (signature over complete assertion). The signature must be an enveloped signature and applied to the **`saml2:Assertion`** element and all its children. The signature must contain a single **`../ds:Reference`** containing the **`saml2:Assertion/@ID`** attribute value and must be signed using the certificate defined within the GW's SPOCS TSL entry. (see D3.2 **Fehler! Verweisquelle konnte nicht gefunden werden.**).

`/saml2:Subject`

Within the SPOCS context, only the element **`saml2:NameID`** is used.

`/saml2:Subject/saml2:NameId`

Mandatory identifier that represents the Subject. The **`attribute @SPNameQualifier`** MUST NOT be used.

`/saml2:Subject/saml2:SubjectConfirmation`

This mandatory element provides means for verification of the correspondence between the SAML subject (sender) with the party whom the relying party is communicating with (destination GW).

`/saml2:Subject/saml2:SubjectConfirmation/@Method`

This attribute MUST be present with a value of "**`urn:oasis:names:tc:SAML:2.0:cm:sender-vouches`**", which denotes that the issuing Gateway vouches for the subject confirmation of the sender.

`/saml2:Subject/saml2:SubjectConfirmation/saml2:BaseId, ..saml2:NameId, ..saml2:EncryptedID`

Within SPOCS, these child elements of **`../saml2:Subject/saml2:SubjectConfirmation`** MUST NOT be used.

`/saml2:Subject/saml2:SubjectConfirmation/ saml2:SubjectConfirmationData`

This mandatory element specifies additional data allowing the SAML subject (sender) to be confirmed. Rules for attributes of this element:

Attribute	Support	Notes
`@NotBefore`	MUST	Subject (sender) cannot be confirmed before this time.
`@NotOnOrAfter`	MUST	Subject cannot be confirmed on or after this time.
`@Recipient`	MUST	URI reference of the gateway this assertion is being sent to.
`@InResponseTo`	MUST NOT	Id of the Request that requested this assertion.
`@Address`	MUST NOT	IP address of user that this assertion was issued to.

Table 9: SubjectConfirmationData attributes of a sender's SAML assertion

`/saml2:Conditions`

This mandatory element specifies conditions that must be evaluated when using the **`saml2:Assertion`**. Following attributes MUST be provided:

Attribute	Support	Notes
`@NotBefore`	MUST	Assertion not valid before this time
`@NotOnOrAfter`	MUST	Assertion not valid on or after this time

Table 10: Conditions Attributes of an Authentication Response

`/saml2:Conditions/saml2:AudienceRestriction`

Mandatory element; used to restrict the audience of this assertion to the specific destination domain by outlining its URI reference. This URI must be identical to the URI value defined within the SPOCS TSL (**`tsl:ServiceSupplyPoint`**, see **Fehler! Verweisquelle konnte nicht gefunden werden.**).

`/saml2:Conditions/saml2:OneTimeUse`

This element MUST NOT be used.

`/saml2:Advice`

This element MUST NOT be used.

`/saml2:AuthnStatement`

Mandatory element; it's attribute @**`SessionIndex`** MUST NOT be used.

`/saml2:AuthnStatement/saml2:SubjectLocality`

This element MUST NOT be used.

`/saml2:AuthnStatement/saml2:AttributeStatement`

This optional element contains several <**`saml2:Attribute`**> child elements carrying information associated with the SAML subject (sender).

`/saml2:AuthnStatement/saml2:AttributeStatement/saml2:Attribute`

A gateway MAY provide any **`saml2:Attribute`** element defined by STORK. However, at least one **`saml2:Attribute`** MUST be provided. If no identification information about the sender is at hand, a possible saml2:Attribute could be the sender's e-Address in RFC 822 format (see STORK http://www.stork.gov.eu/1.0/eMail attribute) or in general the sender's e-Identifier (see STORK http://www.stork.gov.eu/1.0/eIdentifier attribute).

Some solutions provide information about the end entity's initial registration process strongness. To be able to transfer this information, the following saml2:Attribute element is defined and MAY be provided:

`/saml2:Attribute/@Name`

Mandatory attribute with a value of **`"http://www.stork.gov.eu/1.0/citizenQAARegistrationLevel"`**

`/saml2:Attribute/@NameFormat`

Mandatory attribute with a value of "**`urn:oasis:names:tc:SAML:2.0:attrname-format:uri`**"

`/saml2:Attribute/@FriendlyName`

Optional attribute; if present it MUST have a value of "Citizen QAA Registration Level"

`/saml2:Attribute/saml2:AttributeValue`

The value of the elements denotes the registration strongness level in the format **`xs:positiveInteger`** with possible values 1 to 4 according to those used for STORK authentication (QAA) level **Fehler! Verweisquelle konnte nicht gefunden werden.**.

4.1.1 Mapping: STORK QAA Level to rem:AuthenticationMethod

As ETSI TS 102 640 makes no uses yet of STORK outcomes an SAML, following mapping is defined for the mapping of those values, which MUST be considered when producing the authentications details for REM Evidences:

STORK		REM	
Assurance	**QAA level**	**Meaning**	**Authentication Method**[14]
No or minimal	1	Not covered by REM, the optional elements **`Sender`**- resp. **`RecipientAuthenicationsDetails`** MUST NOT be provided is this case.	
Low	2	Basic mechanisms such as passwords for use of signature.	Basic
		Using enhanced authentication such as two factor mechanisms linked to a one time password.	Enhanced
Substantial	3	Strong authentication using client certificate via mutual SSL.	Strong
		Using advanced electronic signatures.	AdES
		Using advanced electronic signatures with Secure Signature Creation Devices (as defined in Directive 1999/93/EC [1]) or equivalent secure cryptographic device.	AdES-Plus
High	4	Using qualified[15] electronic signatures with Secure Signature Creation Devices and Qualified Certificates (or e.g. eID token issued by MS authorities).	QES

Table 11: Mapping of STORK QQA level to REM Authencation Method

When mapping from STORK QAA values to **`rem:AuthenticationMethod`**, it SHOULD always select the lowest according REM value, if no further information is available on the authentication method used.

[14] These values are defined as URIs with the prefix ""http:uri.etsi.org/REM/AuthMethod#"

[15] ETSI TS 102 640, part 2, says "advanced" here, what is assumed to be a typo.

Note: This mapping is a preliminary suggestion, as to be used in the piloting phase. It MUST be an issue of further related policy alignment between MS respective the different solutions relying on such mechanisms and attestations.

4.2 Sample SAML Assertion

```
<?xml version="1.0" encoding="UTF-8"?>
<saml2:Assertion ID="_33e1e51773f1c7f9efb5fe26c95aa8c9"
IssueInstant="2010-07-06T15:15:42.797Z" Version="2.0"
xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion">
   <saml2:Issuer
      Format="urn:oasis:names:tc:SAML:2.0:nameid-format:entity">
      https://gateway.zustellung.gv.at
   </saml2:Issuer>
   <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
      <ds:SignedInfo>
         <ds:CanonicalizationMethod
            Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/>
         <ds:SignatureMethod
               Algorithm=
                  "http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
         <ds:Reference URI="#_33e1e51773f1c7f9efb5fe26c95aa8c9">
            <ds:Transforms>
               <ds:Transform Algorithm=
         "http://www.w3.org/2000/09/xmldsig#enveloped-signature"/>
               <ds:Transform Algorithm=
                     "http://www.w3.org/2001/10/xml-exc-c14n#">
                  <ec:InclusiveNamespaces xmlns:ec=
                     "http://www.w3.org/2001/10/xml-exc-c14n#"
                  PrefixList=
                     "ds saml2 saml2p stork storkp spocs xs"/>
               </ds:Transform>
            </ds:Transforms>
            <ds:DigestMethod
               Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
            <ds:DigestValue>
               oCwegCdvfZVX8uSwPlouiboXxug=
            </ds:DigestValue>
         </ds:Reference>
      </ds:SignedInfo>
      <ds:SignatureValue>tWQDL1OlHt5NOu5G7z...</ds:SignatureValue>
      <ds:KeyInfo>
         <ds:X509Data>
            <ds:X509Certificate>MIIFbDCCBFSg...
            </ds:X509Certificate>
         </ds:X509Data>
      </ds:KeyInfo>
   </ds:Signature>
   <saml2:Subject>
      <saml2:NameID Format=
            "urn:oasis:names:tc:SAML:2.0:nameid-format:unspecified"
            NameQualifier="https://gateway.zustellung.gv.at">
         urn:oasis:names:tc:SAML:2.0:nameid-format:unspecified
      </saml2:NameID>
      <saml2:SubjectConfirmation Method=
        "urn:oasis:names:tc:SAML:2.0:cm:sender-vouches">
     <saml2:SubjectConfirmationData
             NotOnOrAfter="2010-07-25T15:20:42.797Z"
               Recipient="https://gateway.pec.it"/>
  </saml2:SubjectConfirmation>
</saml2:Subject>
<saml2:Conditions NotBefore="2010-07-02T15:15:42.798Z"
                    NotOnOrAfter="2010-07-05T15:20:42.797Z">
  <saml2:AudienceRestriction>
     <saml2:Audience>https://uri.pec.it</saml2:Audience>
  </saml2:AudienceRestriction>
</saml2:Conditions>
<saml2:AuthnStatement AuthnInstant="2010-07-06T15:15:42.797Z"/>
<saml2:AttributeStatement>
  <saml2:Attribute Name=
          "http://www.stork.gov.eu/1.0/citizenQAAlevel"
                    NameFormat=
          "urn:oasis:names:tc:SAML:2.0:attrname-format:uri">
         <saml2:AttributeValue
                  xmlns:xs="http://www.w3.org/2001/XMLSchema"
                    xmlns:xsi=
                  "http://www.w3.org/2001/XMLSchema-instance"
                  xsi:type="xs:string">4</saml2:AttributeValue>
          </saml2:Attribute>
 <saml2:Attribute Name="http://www.stork.gov.eu/1.0/surname"
               NameFormat=
             "urn:oasis:names:tc:SAML:2.0:attrname-format:uri">
     <saml2:AttributeValue
             xmlns:xs="http://www.w3.org/2001/XMLSchema"
               xmlns:xsi=
             "http://www.w3.org/2001/XMLSchema-instance"
        xsi:type="xs:string">
          Mustermann
     </saml2:AttributeValue>
  </saml2:Attribute>
  <saml2:Attribute Name=
          "http://www.stork.gov.eu/1.0/givenName"  NameFormat=
          "urn:oasis:names:tc:SAML:2.0:attrname-format:uri">
     <saml2:AttributeValue
             xmlns:xs="http://www.w3.org/2001/XMLSchema"
             xmlns:xsi=
          "http://www.w3.org/2001/XMLSchema-instance"
          xsi:type="xs:string">
          Max
     </saml2:AttributeValue>
  </saml2:Attribute>
  <saml2:Attribute Name=
        "http://www.stork.gov.eu/1.0/dateOfBirth"   NameFormat=
        "urn:oasis:names:tc:SAML:2.0:attrname-format:uri">
     <saml2:AttributeValue
             xmlns:xs="http://www.w3.org/2001/XMLSchema"
               xmlns:xsi=
          "http://www.w3.org/2001/XMLSchema-instance"
     xsi:type="xs:string">
          1980-12-31
        </saml2:AttributeValue>
     </saml2:Attribute>
     <saml2:Attribute Name=
          "http://www.stork.gov.eu/1.0/eIdentifier"   NameFormat=
          "urn:oasis:names:tc:SAML:2.0:attrname-format:uri">
        <saml2:AttributeValue
             xmlns:xs="http://www.w3.org/2001/XMLSchema"
               xmlns:xsi
             ="http://www.w3.org/2001/XMLSchema-instance"
        xsi:type="xs:string">
             BiDFch0qWafjp3moTxMkxEhajkliiKry5Yq6Kt9U
        </saml2:AttributeValue>
     </saml2:Attribute>
   </saml2:AttributeStatement>
</saml2:Assertion>
```

Listing 2: Example SAML assertion

4.3 SAML Token Validation

A destination Gateway MUST validate the SAML token of incoming messages – format, token signature, validity time constraints. If an error is detected, an Evidence MUST be raised and send back to the message source Gateway in the SOAP response. The message MUST be discarded.

RelayToREMMDAcceptanceRejection	
Element	**Possible / mandatory values**
EventCode	**http:uri.etsi.org/REM/Event#Rejection**
EventReasons	*with one child element:*
../EventReason	*and child elements:*
../../Code	**http:uri.etsi.org/REM/EventReason#PolicyViolation**
../../Details	String "SAML token: " concatenated with the XPath pointing to the element causing the validation fault.

Table 12: REM-Event in case of SAML token validation fault

5 SOAP Body Layout

The SOAP body content covers either a **remsoap:REMDispatch** or **remsoap:REMMDMessage** complex element, depending on the **wsa:Action** outlined in the SOAP header.

According mapping from/to Realm's domestic is a matter of the specification for a concrete Gateway – as well as for the inbound as for the outbound case. The generic part of the Gateway implementation MUST provide "setter" and "getter" functionality for each single element and attribute described above.

remsoap:REMDispatch as well as **remsoap:REMMDMessage** MUST be signed by the source Gateway performing the conversion from the domestic to the "normalized" format and/or generating the evidences. First of all, this is done be using the according WS-Security feature as described in section 3. The **ds:Reference** element of the **ds:Signature** element in the WS Security header points to s12:Body element.

As Evidences may be subject to sustainability and persistence in order to serve as proof for e.g. cases of disputes, a Gateway generating an Evidence MUST supply an enveloped signature over the Evidence complex structure in the according Evidenc **ds:Signature** element.

remsoap:REMDispatch and **rem:REMEvidenceType** contain elements describing their Senders and Recipients in different roles. Structure and semantics of this type is defined first, followed by the subsections for **remsoap:REMDispatch** and **remsoap:REMMDMessage**.

5.1 eDelivery Actor Electronic Addresses

To be able to support different formats of e-addresses**Fehler! Verweisquelle konnte nicht gefunden werden.**, TS 102 640 defines following complex type for e-addresses:

```
<xs:complexType name="AttributedElectronicAddressType">
  <xs:simpleContent>
    <xs:extension base="tsl:NonEmptyURIType">
      <xs:attribute name="scheme" type="tsl:NonEmptyStringType"
             default="mailto"/>
        <xs:annotation>
          <xs:documentation>defaults to mailto, if not present
          </xs:documentation>
        </xs:annotation>
      <xs:attribute name="DisplayName"
          type="tsl:NonEmptyStringType"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>
```

Listing 3: Type definition for rem:AttributedElectronicAddress

Description of the outline above:

/rem:AttributedElectronicAddressType

The mandatory, non-empty e-address in the format **`tsl:NonEmptyURIType`**[16]

`/rem:AttributedElectronicAddress/@scheme`

Non-empty attribute of type **`xs:string`**[17], outlining the scheme of the e-address. Value defaults to mailto, if attribute not present. See section 3.3 of **Fehler! Verweisquelle konnte nicht gefunden werden.** for details on e-address schemes.

If this attribute is missing in an e-address element or the addressed GW is not able to serve the schema outlined here, the message MUST be discarded and the destination GW MUST send back an Evidence to the source GW with following details:

RelayToREMMDAcceptanceRejection	
Element	**Possible / mandatory values**
`EventCode`	**`http:uri.etsi.org/REM/Event#Rejection`**
`EventReasons`	*with at least two child elements:*
`../EventReason`	**`http:uri.etsi.org/REM/EventReason #R_REMMD_NotIdentified`**
`../EventReason`	*and child elements:*
`../../Code`	**`http://www.w3.org/2005/08/addressing/soap/fault`**
`../../Details`	Text of "Address scheme not supported or missing: ", concatenated with XPath pointing to the according parent element of **`rem:AttributedElectronicAddress`** causing the fault (one of those in **`../Destinations`**, **`../Originators`** in **`remsoap:Dispatch/remsoap:MetaData`**) respective **`rem:Evidence/rem:EvidenceRecipient`**[17]

Table 13: REM-Event in case of unknown scheme for e-address

`/rem:AttributedElectronicAddress/@DisplayName`

Optional attribute of type **`tsl:NonEmptyStringType`**, carrying a "display-name" related to the e-address (as known from standard e-mail).

GWs MUST express in their TSL-entry the supported address-scheme used on the interoperability layer. A source GW must decide, if the e-address conversion implemented here can deal with the address-scheme supported by the target GW.

[16] **`tsl:NonEmptyURIType`** and **`tsl:NonEmptyStringType`** are defined in ETSI TS 102 231 **Fehler! Verweisquelle konnte nicht gefunden werden.**

[17] See following subsection for definition of these elements

It is left to the concrete GW implementation, if and how conversion is done to the domestic e-address format. Knowledge of the e-address schemes used in foreign Realms is not needed.

Note: Handshake of Gateways concerning e-address scheme

Source Gateways MUST verify whether one of the e-address schemes exposed in the according TSL entry of the destination Gateway is understood and according e-address conversion can be performed; otherwise an Evidence of non acceptance MUST be returned to the source MD.

5.2 Keywords to classify payload carried in the message

RFC 5322 and predecessors define the “keyword” tag, a string comma separated values, which may be used to assert e-Mails certain classificatory information on header level.

Comparable information items can be found in a couple of eDelivery solutions assessed by SPOCS WP3, first of all to be mentioned here the BusDox specification **Fehler! Verweisquelle konnte nicht gefunden werden.** with the elements ProcessIdentifier und DocumentIdentifier, which allow to assert the payload to a certain business process and outline the type of document carried in a message.

Interoperable exchange of classifications should always rely on according agreement of terms used for category assignment and their semantics. To be able to carry keywords and their context information in a generic, extensible manner, TS 102 640 defines the type **`remsoap:KeywordType`**.

For the @meaning attribute of **`remsoap:KeywordType`**,eSPOCS WP3 foresees following values:

`ProcessIdentifier`

SHOULD be used to denote the business process identifier the message is related to. This could e.g. a file- or record number of on administrational process[18].

`DocumentIdentifier`

SHOULD by used to denote a type of document carried by this message.[19]

Document type values (including their scheme) SHOULD be well defined in the context of clear-cut business scenarios, at least if to be used for process automation (e.g. according distribution/ further routing at recipient's side).[20]

`SimulateEvidences`

[18] For SPOCS, this SHALL be used to outine the distinct identifier assigned to a business application process, as usually will be assigned by most of the PSC portal solutions. In other scenarios, it SHOULD be used according comparable semantics, as e.g. as defined in BusDox for "ProcessIdentifier"; see BusDox specifications **Fehler! Verweisquelle konnte nicht gefunden werden.**. Further standardisation activities should be initiated here to come to broadly accepted, well-known enumerations of possible values and their meaning.

[19] This may e.g. be "tax declaration" or "VCD" (virtual company dossier); semantically matching the BusDox "DocumentIdentifier". Further standardisation activities should be initiated here to come to broadly accepted, well-known enumerations of possible values and their meaning.

[20] e.g. in the e-procurement context of PEPPOL, this is in focus of the CEN BII workshop, see http://spec.cenbii.eu/

SHOULD NOT be used, except for testing purposed. This keyword value is intended for test and emulation of eDelivery Gateway functionality, in detail explained in the eDelivery Gateway software documentation.

This list may be extended according to needs of business scenarios to be served in the future.

5.3 REMDispatch Overview

Complex element **`remsoap:REMDispatch`** carries Dispatch-items describing the message content, the message content itself as well as possible attachments in a normalized format; in addition, the original message in the domestic source MD format MUST[21] be included.

To give a brief overview, we precede the scheme outline with following graphical overview:

[21] It is strongly recommended to archive the unchanged original message at recipient's side, even if the recipients domestic eDelivery system is not able to handle for format of the original message. It may be needed as source of proof in case of disputes.

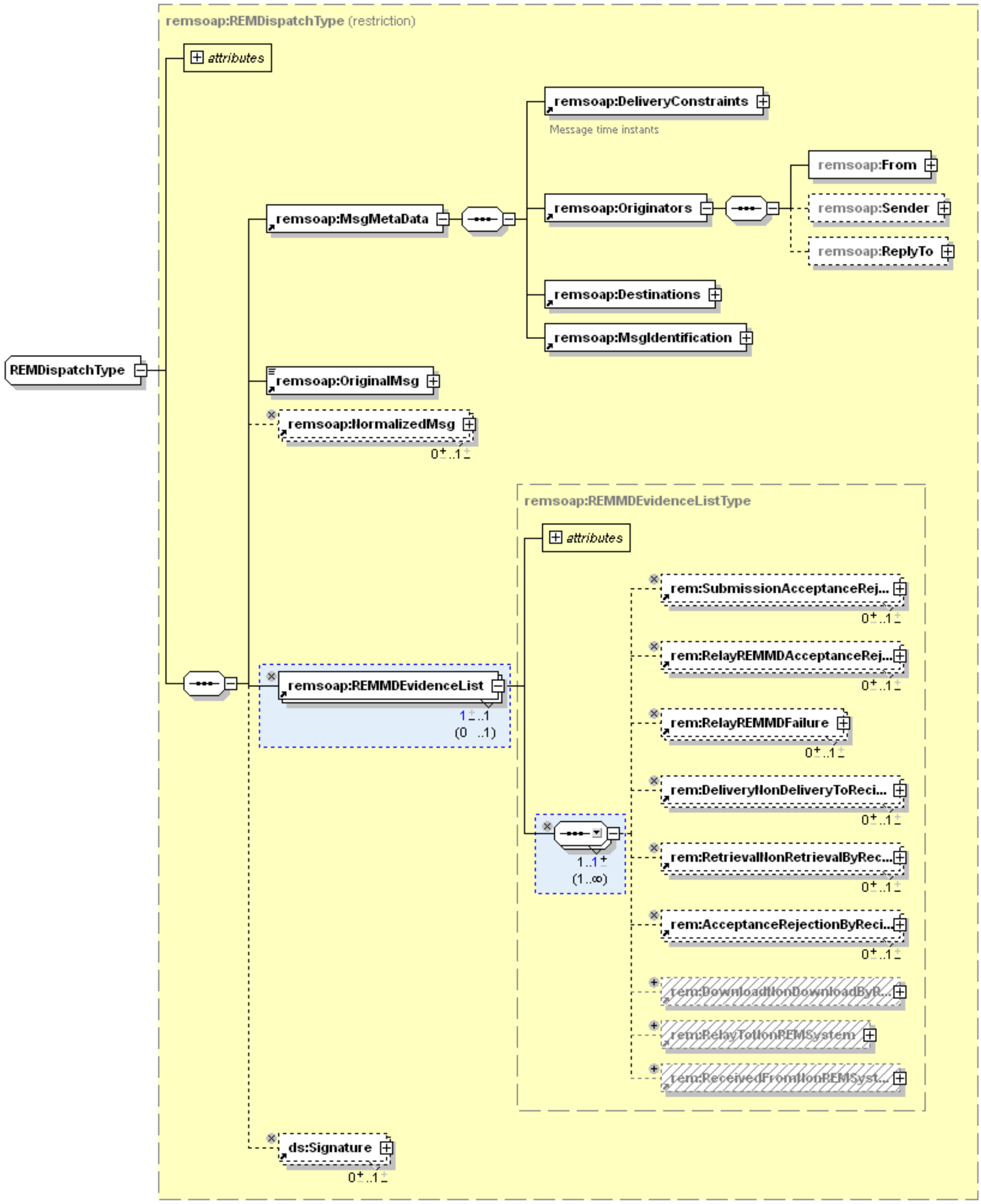
remsoap:REMDispatchType (restriction)
attributes
REMDispatchType
remsoap:MsgMetaData
remsoap:DeliveryConstraints
Message time instants
remsoap:Originators
remsoap:From
remsoap:Sender
remsoap:ReplyTo
remsoap:Destinations
remsoap:MsgIdentification
remsoap:OriginalMsg
remsoap:NormalizedMsg
0..1
remsoap:REMMDEvidenceListType
attributes
remsoap:REMMDEvidenceList
1..1
(0 ..1)
rem:SubmissionAcceptanceRej...
0..1
rem:RelayREMMDAcceptanceRej...
0..1
rem:RelayREMMDFailure
0..1
rem:DeliveryNonDeliveryToReci...
0..1
1..1
(1..∞)
rem:RetrievalNonRetrievalByRec...
0..1
rem:AcceptanceRejectionByReci...
0..1
rem:DownloadNonDownloadByR...
rem:RelayToNonREMSystem
rem:ReceivedFromNonREMSyst...
ds:Signature
0..1

Figure 1: REMDispatch element overview

Detailed schema description of the structure shown above is given in TS 201 640 [46].

SPOCS eDelivery defines a restriction to the original ETSI scheme: only one Evidence element is allowed to be carried along with a REMDispatch, which in the general case should be a **`rem:SubmissionAcceptanceRejection`**, generated by the Gateway of the sending eDelivery realm.

5.4 REMMDMessage Overview

The **`remsoap:REMMMDMessage`** is specified in detail in ETSI STS 102 640 part 2 **Fehler! Verweisquelle konnte nicht gefunden werden.**. For sake of simplification, we restrain to the graphical overview of the structure of **remsoap:REMMDMessage** shown below.

SPOCS eDelivery restricts to carry only elements of type **rem:EvidenceType** as outlined above in section [2].

The optional **ds:Signature** element may be applied in addition to the **ds:Signature** of the invidual Evidence, if more then one Evidence elements are included in REMMDMessage. Individual elements of **rem:EvidencType** elements SHOULD be signed for reasons outlined initial in section [5].

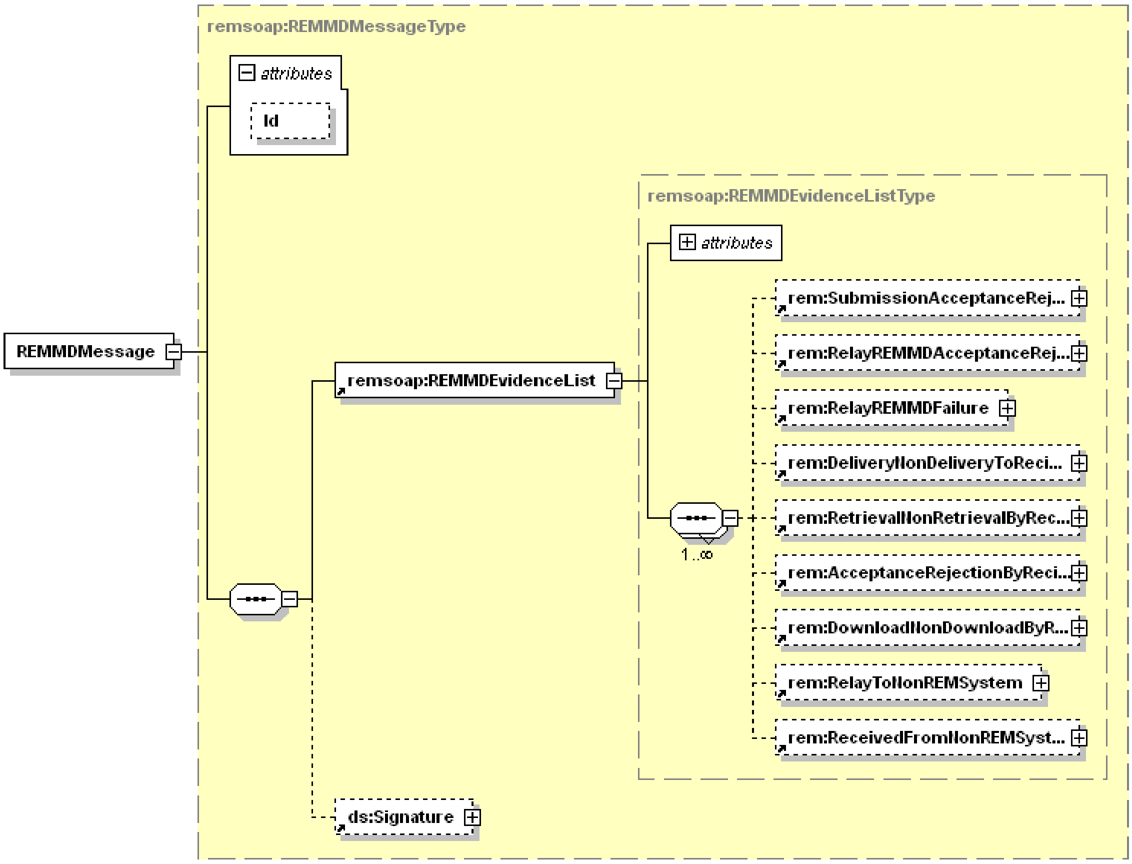
remsoap:REMMDMessageType
attributes
Id
REMMDMessage
remsoap:REMMDEvidenceListType
attributes
remsoap:REMMDEvidenceList
rem:SubmissionAcceptanceRej...
rem:RelayREMMDAcceptanceRej...
rem:RelayREMMDFailure
rem:DeliveryNonDeliveryToReci...
rem:RetrievalNonRetrievalByRec...
rem:AcceptanceRejectionByReci...
rem:DownloadNonDownloadByR...
rem:RelayToNonREMSystem
rem:ReceivedFromNonREMSyst...
1..∞
ds:Signature

Figure 2: REMMDMessage Overview

5.5 Interconnect Protocol Schema

The schemes below are available online via the SPOCS Reference Environment.

5.5.1 **TS 102 640 SOAP Binding Schema**

```
<?xml version="1.0" encoding="UTF-8"?>
<!--REM schema for SOAP body - last edited by Joerg Apitzsch/bos as of 2012-02-02-->
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
xmlns:rem="http://uri.etsi.org/02640/v2#" xmlns:remsoap="http://uri.etsi.org/02640/soapbinding/v1#"
xmlns:xmime="http://www.w3.org/2005/05/xmlmime" xmlns:tsl="http://uri.etsi.org/02231/v2#"
xmlns:xml="http://www.w3.org/XML/1998/namespace" xmlns:xades="http://uri.etsi.org/01903/v1.3.2#"
targetNamespace="http://uri.etsi.org/02640/soapbinding/v1#" elementFormDefault="qualified"
attributeFormDefault="unqualified">
   <xs:import namespace="http://www.w3.org/XML/1998/namespace"
schemaLocation="http://www.w3.org/2001/xml.xsd"/>
   <xs:import namespace="http://www.w3.org/2000/09/xmldsig#"
schemaLocation="http://www.w3.org/TR/2002/REC-xmldsig-core-20020212/xmldsig-core-schema.xsd"/>
   <xs:import namespace="http://www.w3.org/2005/05/xmlmime"
schemaLocation="http://www.w3.org/2005/05/xmlmime"/>
   <xs:import namespace="http://uri.etsi.org/02231/v2#"
schemaLocation="http://uri.etsi.org/02231/v3.1.2/ts_102231v030102_xsd.xsd"/>
   <xs:import namespace="http://uri.etsi.org/01903/v1.3.2#"
schemaLocation="http://uri.etsi.org/01903/v1.3.2/XAdES.xsd"/>
   <xs:import namespace="http://uri.etsi.org/02640/v2#" schemaLocation="ts02640_V2.xsd"/>
   <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>
   <xs:element name="DeliveryConstraints">
      <xs:annotation>
         <xs:documentation>Message time instants</xs:documentation>
      </xs:annotation>
      <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"/>
            <!--Extension point may be used for other contraints, based on schema - e.g. to express message
priority-->
         </xs:sequence>
      </xs:complexType>
   </xs:element>
   <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>
   <xs:element name="Recipient" type="rem:EntityDetailsType"/>
   <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="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>
   <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>
   <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:annotation>
        <xs:documentation>Size in bytes</xs:documentation>
      </xs:annotation>
    </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:annotation>
        <xs:documentation>i.e. offer, billl</xs:documentation>
      </xs:annotation>
      <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>
  <xs:element name="Attachment" type="remsoap:AttachmentType"/>
  <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>
   <xs:element name="NormalizedMsg">
      <xs:complexType>
         <xs:sequence>
            <xs:element ref="remsoap:Informational" minOccurs="0"/>
            <xs:element name="Text" minOccurs="0" maxOccurs="unbounded">
               <xs:annotation>
                  <xs:documentation>
                     The message text
                  </xs:documentation>
               </xs:annotation>
               <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>
      <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>
      <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>
   <xs:element name="REMDispatch" type="remsoap:REMDispatchType"/>
   <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>
   <xs:element name="REMMDMessage" type="remsoap:REMMDMessageType"/>
   <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"/>
   </xs:complexType>
   <xs:element name="REMMDEvidenceList" type="remsoap:REMMDEvidenceListType"/>
</xs:schema>
```

Listing 4; TS 102 640 SOAP Binding Schema

5.5.2 TS 102 640 Evidence Schema

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xml="http://www.w3.org/XML/1998/namespace"
xmlns:xades="http://uri.etsi.org/01903/v1.3.2#" xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
xmlns:tsl="http://uri.etsi.org/02231/v2#" xmlns:xs="http://www.w3.org/2001/XMLSchema"
```

```
xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion" xmlns:rem="http://uri.etsi.org/02640/v2#"
targetNamespace="http://uri.etsi.org/02640/v2#" elementFormDefault="qualified"
attributeFormDefault="unqualified">
  <xs:import namespace="http://uri.etsi.org/01903/v1.3.2#"
schemaLocation="http://uri.etsi.org/01903/v1.3.2/XAdES.xsd"/>
  <xs:import namespace="http://www.w3.org/XML/1998/namespace"
schemaLocation="http://www.w3.org/2001/xml.xsd"/>
  <xs:import namespace="http://uri.etsi.org/02231/v2#"
schemaLocation="http://uri.etsi.org/02231/v3.1.2/ts_102231v030102_xsd.xsd"/>
  <xs:import namespace="http://www.w3.org/2000/09/xmldsig#"
schemaLocation="http://www.w3.org/TR/2002/REC-xmldsig-core-20020212/xmldsig-core-schema.xsd"/>
  <xs:import namespace="urn:oasis:names:tc:SAML:2.0:assertion" schemaLocation="http://docs.oasis-
open.org/security/saml/v2.0/saml-schema-assertion-2.0.xsd"/>
  <!-- List of evidence -->
  <xs:element name="SubmissionAcceptanceRejection" type="rem:REMEvidenceType"/>
  <xs:element name="RelayREMMDAcceptanceRejection" type="rem:REMEvidenceType"/>
  <xs:element name="RelayREMMDFailure" type="rem:REMEvidenceType"/>
  <xs:element name="DeliveryNonDeliveryToRecipient" type="rem:REMEvidenceType"/>
  <xs:element name="DownloadNonDownloadByRecipient" type="rem:REMEvidenceType"/>
  <xs:element name="RetrievalNonRetrievalByRecipient" type="rem:REMEvidenceType"/>
  <xs:element name="AcceptanceRejectionByRecipient" type="rem:REMEvidenceType"/>
  <xs:element name="RelayToNonREMSystem" type="rem:REMEvidenceType"/>
  <xs:element name="ReceivedFromNonREMSystem" type="rem:REMEvidenceType"/>
  <!-- EvidenceType definition -->
  <xs:complexType name="REMEvidenceType">
    <xs:sequence>
      <xs:element ref="rem:EventCode" minOccurs="0"/>
      <xs:element ref="rem:EventReasons" minOccurs="0"/>
      <xs:element name="EvidenceIdentifier" type="xs:string"/>
      <xs:element ref="rem:EvidenceIssuerPolicyID" minOccurs="0"/>
      <xs:element ref="rem:EvidenceIssuerDetails"/>
      <xs:element ref="rem:SenderAuthenticationDetails" minOccurs="0"/>
      <xs:element ref="rem:RecipientAuthenticationDetails" minOccurs="0"/>
      <xs:element name="EventTime" type="xs:dateTime"/>
      <xs:element name="SubmissionTime" type="xs:dateTime" minOccurs="0"/>
      <!-- ReplyTo type changed from xs:string to rem:AttributedElectronicAddressType in version #2 -->
      <xs:choice minOccurs="0">
        <xs:element name="ReplyTo" type="xs:string"/>
        <xs:element name="ReplyToAddress" type="rem:AttributedElectronicAddressType"/>
      </xs:choice>
      <xs:element ref="rem:SenderDetails"/>
      <xs:element ref="rem:RecipientsDetails"/>
      <xs:element ref="rem:RecipientsDelegatesDetails" minOccurs="0"/>
      <xs:element name="EvidenceRefersToRecipient" type="xs:integer" minOccurs="0"/>
      <xs:element ref="rem:SenderMessageDetails" minOccurs="0"/>
      <xs:element ref="rem:NotificationMessageDetails" minOccurs="0"/>
      <xs:element name="ForwardedToExternalSystem" type="xs:string" minOccurs="0"/>
      <xs:element ref="rem:TransactionLogInformation" minOccurs="0"/>
      <xs:element ref="rem:Extensions" minOccurs="0"/>
      <xs:element ref="ds:Signature" minOccurs="0"/>
    </xs:sequence>
    <xs:attribute name="version" type="xs:string" use="required"/>
    <xs:attribute name="Id" type="xs:ID" use="optional"/>
  </xs:complexType>
  <!-- EventCode -->
  <xs:element name="EventCode" type="xs:anyURI"/>
  <!-- EventReasons -->
  <xs:element name="EventReasons" type="rem:EventReasonsType"/>
  <xs:complexType name="EventReasonsType">
    <xs:sequence>
      <xs:element ref="rem:EventReason" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
  <xs:element name="EventReason" type="rem:EventReasonType"/>
  <xs:complexType name="EventReasonType">
    <xs:sequence>
      <xs:element name="Code" type="xs:anyURI"/>
      <xs:element name="Details" type="xs:string" minOccurs="0"/>
      </xs:sequence>
   </xs:complexType>
   <!-- EvidenceIssuerPolicyID-->
   <xs:element name="EvidenceIssuerPolicyID" type="rem:EvidenceIssuerPolicyIDType"/>
   <xs:complexType name="EvidenceIssuerPolicyIDType">
      <xs:sequence>
         <xs:element name="PolicyID" type="xs:anyURI" maxOccurs="unbounded"/>
      </xs:sequence>
   </xs:complexType>
   <!-- EntityDetailsType -->
   <xs:element name="EvidenceIssuerDetails" type="rem:EntityDetailsType"/>
   <xs:complexType name="EntityDetailsType">
      <xs:sequence>
         <xs:element ref="rem:NamesPostalAddresses" minOccurs="0"/>
         <xs:choice minOccurs="0" maxOccurs="unbounded">
            <xs:element ref="rem:AttributedElectronicAddress"/>
            <xs:element ref="tsl:ElectronicAddress"/>
         </xs:choice>
         <xs:element ref="rem:CertificateDetails" minOccurs="0"/>
         <xs:element ref="xades:Any" minOccurs="0"/>
      </xs:sequence>
   </xs:complexType>
   <!-- AttributedElectronicAddressType - introduced in Version #2 as an alternative to
tsl:ElectronicAddress -->
   <xs:complexType name="AttributedElectronicAddressType">
      <xs:simpleContent>
         <xs:extension base="tsl:NonEmptyURIType">
            <xs:attribute name="scheme" type="xs:string" default="mailto">
               <xs:annotation>
                  <xs:documentation>Defaults to mailto, if not present</xs:documentation>
               </xs:annotation>
            </xs:attribute>
            <xs:attribute name="DisplayName" type="tsl:NonEmptyString"/>
         </xs:extension>
```

```
    </xs:simpleContent>
  </xs:complexType>
  <xs:element name="AttributedElectronicAddress" type="rem:AttributedElectronicAddressType"/>
  <xs:element name="NamesPostalAddresses" type="rem:NamesPostalAddressListType"/>
  <xs:complexType name="NamesPostalAddressListType">
    <xs:sequence>
      <xs:element ref="rem:NamePostalAddress" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
  <xs:element name="NamePostalAddress" type="rem:NamePostalAddressType"/>
  <xs:complexType name="NamePostalAddressType">
    <xs:sequence>
      <xs:element ref="rem:EntityName" minOccurs="0"/>
      <xs:element ref="rem:PostalAddress" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
  <xs:element name="EntityName" type="rem:EntityNameType"/>
  <xs:complexType name="EntityNameType">
    <xs:sequence>
      <xs:element name="Name" type="tsl:NonEmptyString" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute ref="xml:lang" use="optional"/>
  </xs:complexType>
  <xs:element name="PostalAddress" type="rem:PostalAddressType"/>
  <xs:complexType name="PostalAddressType">
    <xs:sequence>
      <xs:element name="StreetAddress" type="tsl:NonEmptyString" maxOccurs="unbounded"/>
      <xs:element name="Locality" type="tsl:NonEmptyString"/>
      <xs:element name="StateOrProvince" type="tsl:NonEmptyString" minOccurs="0"/>
      <xs:element name="PostalCode" type="tsl:NonEmptyString"/>
      <xs:element name="CountryName" type="tsl:NonEmptyString"/>
    </xs:sequence>
    <xs:attribute ref="xml:lang" use="optional"/>
  </xs:complexType>
  <xs:element name="CertificateDetails" type="rem:CertificateDetailsType"/>
  <xs:complexType name="CertificateDetailsType">
    <xs:choice>
      <xs:element name="X509Certificate" type="xs:base64Binary"/>
      <xs:element name="CertID" type="xades:CertIDType"/>
      <xs:element ref="rem:CertIDAndSignature"/>
    </xs:choice>
  </xs:complexType>
  <xs:element name="CertIDAndSignature" type="rem:CertIDAndSignatureType"/>
  <xs:complexType name="CertIDAndSignatureType">
    <xs:sequence>
      <xs:element name="IssuerSerial" type="xades:DigestAlgAndValueType"/>
      <xs:element name="tbsCertificateDigestDetails" type="xades:DigestAlgAndValueType"/>
      <xs:element ref="rem:CertSignatureDetails"/>
    </xs:sequence>
  </xs:complexType>
  <xs:element name="CertSignatureDetails" type="rem:CertSignatureDetailsType"/>
  <xs:complexType name="CertSignatureDetailsType">
    <xs:sequence>
      <xs:element ref="ds:SignatureMethod"/>
      <xs:element ref="ds:SignatureValue"/>
    </xs:sequence>
  </xs:complexType>
  <!-- AuthenticationDetailsType -->
  <xs:element name="SenderAuthenticationDetails" type="rem:AuthenticationDetailsType"/>
  <xs:element name="RecipientAuthenticationDetails" type="rem:AuthenticationDetailsType"/>
  <xs:complexType name="AuthenticationDetailsType">
    <xs:sequence>
      <xs:choice>
        <xs:sequence>
          <xs:element name="AuthenticationTime" type="xs:dateTime"/>
          <xs:element name="AuthenticationMethod" type="xs:anyURI"/>
        </xs:sequence>
```

```
            <!-- saml:Assertion - introduced in Version #2 as an alternative to
rem:AuthenticationTime/Method -->
            <xs:element ref="saml:Assertion"/>
         </xs:choice>
         <xs:element name="AdditionalDetails" type="xades:AnyType" minOccurs="0"/>
      </xs:sequence>
   </xs:complexType>
   <!-- EntityDetailsListType -->
   <xs:element name="SenderDetails" type="rem:EntityDetailsType"/>
   <xs:element name="RecipientsDetails" type="rem:EntityDetailsListType"/>
   <xs:complexType name="EntityDetailsListType">
      <xs:sequence>
         <xs:element name="EntityDetails" type="rem:EntityDetailsType" maxOccurs="unbounded"/>
      </xs:sequence>
   </xs:complexType>
   <!-- RecipientsDelegatesDetailsType -->
   <xs:element name="RecipientsDelegatesDetails" type="rem:RecipientsDelegatesType"/>
   <xs:complexType name="RecipientsDelegatesType">
      <xs:sequence maxOccurs="unbounded">
         <xs:element name="DelegateDetails" type="rem:EntityDetailsType"/>
         <xs:element name="DelegatingRecipients" type="rem:ListOfIntegers"/>
      </xs:sequence>
   </xs:complexType>
   <xs:simpleType name="ListOfIntegers">
      <xs:list itemType="xs:integer"/>
   </xs:simpleType>
   <xs:element name="SenderMessageDetails" type="rem:MessageDetailsType"/>
   <xs:element name="NotificationMessageDetails" type="rem:MessageDetailsType"/>
   <xs:complexType name="MessageDetailsType">
      <xs:sequence>
         <xs:element name="MessageSubject" type="xs:string"/>
         <xs:element name="UAMessageIdentifier" type="xs:string" minOccurs="0"/>
         <xs:element name="MessageIdentifierByREMMD" type="xs:string"/>
         <xs:element ref="ds:DigestMethod" minOccurs="0"/>
         <xs:element ref="ds:DigestValue" minOccurs="0"/>
      </xs:sequence>
      <xs:attribute name="isNotification" type="xs:boolean" use="required"/>
   </xs:complexType>
   <xs:element name="TransactionLogInformation" type="rem:TransactionLogInformationType"/>
   <xs:complexType name="TransactionLogInformationType">
      <xs:sequence>
         <xs:element ref="rem:TransactionLog" maxOccurs="unbounded"/>
      </xs:sequence>
   </xs:complexType>
   <xs:element name="TransactionLog" type="xades:AnyType"/>
   <xs:element name="Extensions" type="rem:ExtensionsListType"/>
   <xs:complexType name="ExtensionsListType">
      <xs:sequence maxOccurs="unbounded">
         <xs:element ref="rem:Extension"/>
      </xs:sequence>
   </xs:complexType>
   <xs:element name="Extension" type="rem:ExtensionType"/>
   <xs:complexType name="ExtensionType">
      <xs:complexContent>
         <xs:extension base="xades:AnyType">
            <xs:attribute name="isCritical" type="xs:boolean" use="optional"/>
         </xs:extension>
      </xs:complexContent>
   </xs:complexType>
</xs:schema>
```

Listing 5: TS 102 640 Evidence Schema

5.5.3 SPOCS Restriction of TS 102 640 SOAP Binding Schema

```
<?xml version="1.0" encoding="UTF-8"?>
<!--SPOCS redefine of REM schema for SOAP body - last edited by Joerg Apitzsch/bos as of 2012-03-21-->
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:remsoap="http://uri.etsi.org/02640/soapbinding/v1#" xmlns:rem="http://uri.etsi.org/02640/v2#"
xmlns:ds="http://www.w3.org/2000/09/xmldsig#" targetNamespace="http://uri.etsi.org/02640/soapbinding/v1#"
elementFormDefault="qualified" attributeFormDefault="unqualified">
   <xs:import namespace="http://www.w3.org/2000/09/xmldsig#"
schemaLocation="http://www.w3.org/TR/2002/REC-xmldsig-core-20020212/xmldsig-core-schema.xsd"/>
   <xs:import namespace="http://uri.etsi.org/02640/v2#" schemaLocation="ts02640_V2.xsd"/>
   <xs:redefine schemaLocation="ts102640_soap_body.xsd">
      <xs:complexType name="REMDispatchType">
         <xs:complexContent>
            <xs:restriction base="remsoap: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"/>
                  <xs:element ref="ds:Signature" minOccurs="0"/>
               </xs:sequence>
            </xs:restriction>
         </xs:complexContent>
      </xs:complexType>
      <xs:complexType name="REMMDEvidenceListType">
         <xs:complexContent>
            <xs:restriction base="remsoap:REMMDEvidenceListType">
               <xs:sequence>
                  <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:sequence>
          </xs:restriction>
        </xs:complexContent>
      </xs:complexType>
    </xs:redefine>
</xs:schema>
```

Listing 6: SPOCS Restriction of TS 102 640 SOAP Binding Schema

6 Summary of Characteristics of piloting eDelivery Solutions to be interconnected

The objective of this section is to present the different national approaches of the message envelope formats, the metadata and structure as well as the addressing mechanisms.

It classifies the transport metadata for the solutions of the piloting partners IT PEC, DE OSCI, AT, eGIF and EPUAP. It also describes the overall structure of the message, the envelopes and the metadata containers in regards to semantics and format.

eDelivery incorporates the addressing of the endpoints. The communication occurs through interoperable endpoints and specifically is based on a transport solution, which incorporates the addressing of these endpoints and messaging systems. The solutions use either a SMTP- or HTTP-based transport layer.

To achieve the goal of interoperable interconnection of the existing eDelivery system, a short overview is given on the different addressing formats and mechanisms in use.

Detailed mapping to the SPOCS Interconnect Protocol will be a matter of amending specifications to be made per solution to be attached to the SPOCS eDelivery sphere.

6.1 Austrian Solution

Austria’s eDelivery system is based upon a Central Lookup Service, which includes the information of all recipients registered with any Austrian eDelivery service provider. Senders must register with the Central Lookup Service in order to search for recipients. More specific, senders before delivering a message, check for a recipient by querying the Central Lookup Service through a HTTPs GET request. An error is returned from the Central Lookup Service in case the recipient cannot be found or the recipient cannot be uniquely identified. A selection of eDelivery SP's follows from the list that has been returned by the Central Lookup Service.

6.1.1 Messages

Senders transmit messages directly to the recipient’s eDelivery Service Provider. The transmission is carried out using the SOAP with attachments (SwA) protocol. Documents like attachments are carried as SwA MIME parts. The meta-data is carried as well-defined XML structure within the SOAP body. This meta-data contains the recipient’s unique identifier, the sender’s and recipient’s identity information, the delivery quality indicating whether a receipt is required or not, message-ids, etc.

6.1.2 Evidences

The eDelivery SP sends back a signed receipt if a sender requests it. This is done either via e-mail or a web-service interface (depends on the sender’s demands). The signed receipt is a non-repudiation of receipt message based on a well-defined XML structure and must be signed by the recipient using her national eID. In case the message is fetched using a mail client, the receipt must be signed by the eDelivery SP on behalf of the recipient. The signed XML structure contains a reference to the pertaining message-id, the sender’s and recipient’s identity information as well as information about the notifications that have been carried out by the eDelivery SP to inform that the message can be picked up. In case a recipient does not pick up a delivery, a non-delivery evidence is sent back to the sender (in the same way as the proof of receipt).

6.1.3 **Addressing**

The Austrian system defines two steps for addressing recipients. In a first step, senders must search for recipients using a Central Lookup Service. This service returns the addresses of the eDelivery SPs a recipient is registered with. If a recipient cannot be found, an appropriate error is returned. Second, senders must use these returned address data to transmit a message to the final eDelivery SP.

Senders may use the following address criteria to search for recipients at the Central Lookup Service:

- Unique ID (sector identifier) [this identifier may only be used by Austrian public authorities]
- Given name, family name, date of birth, address (e-mail, telephone number or postal address)

The Lookup Service returns a list of those eDelivery Service Providers a recipient is registered with. Each list entry (eDelivery SP) contains the following elements:

- Recipient's ID (Base64 String)
- URL (HTTPs address) of the eDelivery SP web-service location (message is sent to this service)
- Encryption certificate & supported MIME types

Senders transmit the delivery to the given web-service as SwA request. The delivery contains the recipient's ID within the meta-data of the SOAP payload.

6.1.4 **Metadata summary**

Name	**Comment**
MessageID	Unique ID of dispatch message
Relations [1..n]	A dispatch may be related to particular other dispatches (a list of related message-ids)
MessageParts [1..n] (= attachments/payload)	
ID	Unique ID of attachment
Name	File name of attachment
Size	File size of attachment
Description	Short description (e.g. for visual presentation)
Languages [1..n]	Supported languages of attachment (e.g. is document in DE/FR/EN)
Digest (Value / Type)	Hash value (value + algorithm) for attachment integrity
MimeType	Mime type of attachment
Content	Reference to MTOM attachment
Recipient	
ID	Unique ID (e.g. REM address, …)
Demographics data	Name, Date of birth, postal address, telephone, electronic address

Name	Comment
Sender	
ID / Demographics data	Same as recipient
Delegate data	ID / demographics data for a sender's delegate
Reply-To address	Similar to use in e-mail world
Authentication Level	STORK QAA Level of sender
SourceDSP	Data about the sender's DSP (e.g. policy ID)
Subject	Dispatch subject (similar to e-mail for visual presentation)
Delivery Options	
Personal	If this delivery can only be picked-up by recipient herself and not by a delegate
PrivateDelivery	Is this dispatch intended for private sector or for public sector?
RecipientAuthLevelRequired	Required STORK QAA level for recipient
TimeConstraints	Time constraints for sending back evidences etc..
SendingTimeStamp	Sending timestamp
SourceSignatures	Encapsulated signatures of original domain-specific dispatch message
Signature	Message signature

Table 14 AT: eDelivery - message metadata

6.2 German Solution (OSCI)

The German OSCI eDelivery system is a federated one, organized in domains. Each domain has a IdP, following the concepts of WS-Trust and WS-Federation. The IdP offers participants lookup functionality in form of an Attribute Service (AS). For authentication and authorization purposes, a Security Token Service (STS) is provided. Domain participants must register with the Identity Provider (IdP. For correct recipient addressing, senders query the AS through Service Provisioning Markup Language (SPML) requests. The AS provides a mapping of user-friendly addresses to the technical e-addresses used by the OSCI protocol.

6.2.1 **Messages**

Senders transmit messages directly to the recipient's eDelivery Service Provider. The transmission is carried out through SOAP protocol. Message payload is carried in the SOAP body. By use of MTOM, additional documents like attachments are carried as MIME parts on the wire. The payload related metadata is carried as well-defined XML structure within the SOAP body.

The only metadata visible on the SOAP header are

- Referenced (foregoing) messages

□ a reference to the business scenario type a message is related to (this is predefined, extendable list in form of URIs)

□ message time stamps, applied by different nodes on the message route

  - message obsolete after (optionally applied by sender)
  - message delivered (applied on acceptance by targeted eDelivery system
  - message pulled by recipient (applied by eDelivery system, when recipients pulls message out of his inbox)
  - message receipted (applied by recipient, only for his own tracing purposes

In addition, a sender uses dedicated headers to demand for a "delivery receipt", "fetched notification", "reception receipt" (see next section).

6.2.2 Evidences

If demanded for, the receipts / notifications mentioned above, "receipts" or a notification is delivered to the sender's inbox. An XML-format is defined for theses items, containing signatures on a qualified timestamp on demand of the initial sender.

The "Delivery Receipt" must be provided by the targeted domain system on acceptance of the message.

The "Reception Receipt" must be provided by the recipient after successful receiption and decryption of the message.

Both Delivery Receipt and Reception Receipt reference the payload send in the receipt signature, in addition to message id, sender/recipient and message time stamps.

A "Fetched Notification" has to be send to the sender by the recipient domain system, when the recipient pulls the message out of his inbox. A Fetched Notification has only informal value, is not protected by a signature.

It's left to implementations, to provide additional evidence functionality, e.g. a notification of message entrance to the recipient's inbox, delivered to the recipient by means of e-mail, SMS or other channels.

Delivery faults are covered by SOAP faults already defined by the WS-* bricks used, OSCI defines some SOAP faults in addition. Such faults, delivered to the sender, are not designed for provability of unsuccessful communication; they only serve operational issues. Thus, faults are not signed, no format for persistence of such evidence type messages is defined.

Addressing

Based on WS-*, the German OSCI infrastructure uses http/SOAP for technical addressing of the recipients by individual URLs, carried in wsa:To. WS-Addressing and WS-Addressing SOAP-Binding is profiled, the more generic construct of wsa:EndpointReference is used for outlining the senders address(es) wsa:From and wsa:ReplyTo.

Multiple recipients or recipients in cc are not foreseen on the protocol and delivery mechanisms level – implementation of such features is left to providers of OSCI-based products.

Outside the protocol layer, end-user applications supporting OSCI offer "friendly names" for OSCI communication domain participants, which are mapped to the technical addressing format internally. Thus, even e-mail like addresses following certain naming conventions can be used for end-users.

OSCI foresees to extend a wsa:EndpointReference by a wsa:ReferenceParameter to point to a business scenario the message is related to. But, in the present context, such a "classification" of the message payload must be seen as related to the message metadata and not to addressing itself.

Further on, OSCI defines distinct types of messages – e.g. request, responses, receipt and fault -, outlined by according URIs of the wsa:Action element. In the present context, attention must exclusively be given to the request-, receipt and probably fault-type messages.

6.2.3 Metadata summary

An OSCI Message is a SOAP(1.2)-Message. The body contains the payload (Message and Attachments, using MTOM), which must be encrypted for the recipient, when using open networks (Internet) for delivery.

The SOAP Header is visible for nodes on the transport route (transport encryption, e.g. https, from node to node)

Name	Semantics	Format/ Reference
MessageID	Unique ID of dispatch message	wsa:MessageID
Relations [0..n]	A dispatch may be related to particular other dispatches (a list of related message-ids)	wsa:RelatesTo
MessageTimeStamps	Applied by nodes on the route	
LifeTime	Message obsolete after, applied by Sender	xs:date
Delivered	Message Accepted at recipients MsgBox, applied by MsgBox	xs:dateTime
Fetched by recipient	Message fetched from MsgBox by recipient, applied by MsgBox	xs:dateTime
Receipted	Message reception time, applied by recipient	xs:dateTime
Security	Applied by sender	WSS header
Security Timestamp	Timestamp for security token	wsu:Timestamp
SAML-Token	Token authenticating sender, including strength of authentication and initial registration (STORK QAA Level of sender) and sym. key for transport encryption/signature	saml:Assertion
Signature	Transport signature element over all headers and body	xdsig
SAML-Token for recipient	May be submitted to recipient to be used to deliver the answer (the one-time) right to store answer in senders MsgBox, including proof token	saml:Assertion

Name	Semantics	Format/ Reference
Sender (=Reply-To address)	Applied by sender, in addition see SAML assertion above	wsa:ReplyTo (URI)
Subject		
Type of message	Outlining, if message is a normal dispatch or a receipt or an error, which occurred on the message route	Wsa:action (URI)
Related BusinessScenario	Only pointing to a defined "BusinessScenarioType" (which points to a specifc scheme (xsd) for the body layout)	Wsa:Reference-parameter related to wsa:To.
Delivery Options	OSCI only carries demand for receipts/ notifications, supplied by sender	
Demand for DeliveryReceipt	Sender wants an acceptance receipt from the recipients MsgBox	See osci spec.
Demand for ReceptionReceipt	Sender wants an reception receipt, to be delivered form recipient at successful message reception	See osci spec.
Demand for FetchedNorificatioon	Sender wants a notification from recipients MsgBox, when recipient pulls the message out	See osci spec.

Table 15: DE OSCI eDelivery - message metadata

Errors occurred (asynchronously) on the message route and receipts requested for evidences are delivered as a new dispatch to the initial sender. Those dispatches have an according different value in wsa: action element (normal dispatch: OSCI Request).

6.3 Greek Solution (ERMIS)

The Greek eDelivery system is an integrated part of the national portal ERMIS (Greek PSC), which provides eDelivery functions to any registered user (service provider). ERMIS is exchanging messages with other information systems that have been registered in the Greek Interoperability Registry. In the case that an ERMIS user (service provider) wants to send a message to a specific public authority he will select one of the available services that are offered and afterwards he will select the public authority that will receive that message. When an ERMIS user is choosing the public authority, indirectly, he/she is choosing the corresponding information system.

ERMIS is communicating with other information systems using PL SQL web services in the case of Citizen Service Centers information system and SOAP 1.2 messages in any other case. Usually the message exchange is done in the context of a predefined business case using available Web Services that are described with WSDL. The message exchange usually related with the process (i.e. process oriented information) and can be found using a UDDI protocol enhanced with information stored in the Greek interoperability registry. Finally in the Greek solution specific messages are used for document transfer with evidence from an information system to another. These messages are process independent and are described in the following paragraph.

A message may be transferred automatically as an integrated process of the information systems, by authorized parties or by the original sender of this.

6.3.1 Messages

The messages that are used for document exchange independently of the process are the following:

1) The ‘StartTransaction’ service for establishing communication channel among two different information systems. The message contains the following items:
    - The transaction ID
    - The name and the ID of the public authority that is sending the message.
    - The name and the ID of the public authority that is the recipient of the message
    - The date and time
    - The status of the transaction as a response (well received message or error code)

    Each public authority is related with the corresponding information system.
2) The ‘InsertCoverLetter’ service for transferring documents and attachments. This message has additionally to the previous one the following elements:
    - The purpose of the document
    - The additional attached files (if any)
3) The ‘EndTransaction’ service for closing the communication channel. This message is the same as the ‘StartTransaction’ message and is used for closing the communication channel.

6.3.2 Evidences

The evidence that is produced among information systems comes from the response of the above mentioned messages regarding the status of the transaction. As long as the message is well received the status is O.K. otherwise an error code will be transferred. Provided that the information systems have SSL certificates the messages are singed using the SSL certificate by the information systems. Based on the sender’s and the receiver’s demands the signed messages can be kept as a proof of the transaction.

The evidence for the end users is either a reference number for the case (or the transaction) received by ERMIS portal or a signed receipt sent by the recipient of the message.

These services are agnostic and can be used for any message exchange.

6.3.3 Addressing

Regarding the Greek case the eDelivery and eSafe solution are part of the National portal ERMIS (Greek PSC). Each registered user has a private space for storing documents in ERMIS. Whenever an answer from public administration information system shall be delivered to a registered user two main steps are followed:

1) In the first step, the information system that produces the answer transfers an electronic message to ERMIS. The communication is done either by using predefined

PL SQL web services or predefined SOAP 1.2 messages. The exact structure of the messages depends on the business case.

2) In the second step the message is transferred to the private space of the registered user. The only information that is needed for that is; either the user's ERMIS 'username' or the request ID that initiated the response from the other information system.

Moreover in the Greek case a registered user can transfer an electronic document to another public authority information system. In that case a predefined service for a specific business case is used in order to upload the documents to the ERMIS portal. Afterwards ERMIS is transferring the document using SOAP 1.2 messages to the other information system.

ERMIS is delivering messages only to specific information systems that have been registered in the Greek interoperability registry. These systems are regarded as trusted and the communication among them and ERMIS is done through a secure channel (VPN and / or SSL). One can find the registered information systems using special search services in the interoperability registry.

6.3.4 Metadata summary

Name	Comment
MessageID	Unique ID of dispatch message
Relations [1..n]	A dispatch may be related to particular other dispatches
MessageParts [1..n] (= attachments/payload)	
ID	Unique ID of attachment
Name	File name of attachment
Size	File size of attachment
Description	Short description (e.g. for visual presentation)
Validity period	XML
Mime type	File or XML type
File type	In case of file this could be PDF or XML
Digest (Value / Type)	Hash value and algorithm
Content	Base64
Recipient	
ID	Unique ID (username or number of CSC and username employee)
Sender	
ID	Unique ID (username)
Reply-to-address	Similar to use in e-mail world
Contact	Available delivery operations

Name	Comment
Authentication Level	STORK QAA Level of sender
Subject	Dispatch subject (similar to e-mail for visual presentation)
Delivery Options	
Personal	If this delivery can only be picked-up by recipient herself and not by a delegate
RecipientAuthLevelRequired	Required STORK QAA level for recipient
TimeConstraints	Time constraints for sending back evidences etc.
SendingTimeStamp	Sending timestamp
SourceSignatures	Encapsulated signatures of original domain-specific dispatch message
Signature	Message signature

Table 16: GR eDelivery - message metadata

6.4 Italian Solution (PEC)

The technical and organizational rules implemented in Italy guarantee that registered e-mail can have the same status as traditional paper based registered mail. The Italian PEC system is based on the following Internet standards:

1. RFC 2822 Internet Message Format
2. RFC 2849 The LDAP Data Interchange Format (LDIF) - Technical Specification
3. RFC 2633 S/MIME Version 3 Message Specification

6.4.1 Messages

The PEC system generates messages conforming to the MIME standard. There is a clear text part describing the message and a further set of attachments (the original message, the certification data, etc.), depending on the message type. The message set is put in a S/MIME v3 envelope (the transport envelope), digitally signed by the sender's provider.

The Italian solution is based on SMTP/POP3/IMAP protocols, where a PEC (eDelivery infrastructure) specific message envelope carries security information. The applied concepts are very similar to the ETSI REM specification. The transport envelope is a message created by the sender's provider carrying the original message and the attached certification data. The same data (and several others) are attached to the receipt in XML format in order to be machine-readable as well (see evidences).

6.4.2 Evidences

The PEC system provides evidences (similar format as messages) for the following events:

- Non-acceptance by the sender's eDelivery SP
- Acceptance by the sender's eDelivery SP
- Expiration of time to delivery to recipient's eDelivery SP

- Message forward to recipient’s eDelivery SP
- Non-acceptance due to virus
- Non-delivery due to virus
- Message delivery
- Non-Delivery

Besides a descriptive text, evidences also contain a machine-readable XML attachment for automatic processing.

6.4.3 Addressing

The PEC addressing is based on the RFC 822 e-mail address format (e.g. claudio.corti@pec-domain.it).

All managed registered PEC domains, i.e. the part after the “@” sign, are listed within a central LDAP server operated by CNIPA (national IT center for the public administration). The sender’s eDelivery provider can thus lookup the responsible eDelivery service provider for taking over a message for a specific managed PEC domain.

Besides addressing recipients in the e-mail “To:” field, PEC also allows the addressing of multiple recipients using the e-mail “Cc:” field.

6.4.4 Metadata summary

REM-MD Evidence	Component Class	Id	Component
	Core Components	G00	REM-MD Evidence Identifier
		G01	REM-MD Evidence Type
		G02	REM Event
		G03	Reason code (see note below) NOTE: Preferably there would be only one (when applicable) G03 listing all remarked exceptions reason codes, but it cannot be excluded that one single message collects more than one G03.
		G04	REM-MD Evidence Version
		G05	Event Time
		G06	Transaction log information
	REM-MD Components	R01	Evidence issuer Policy Identifier
		R02	Evidence issuer Details
		R03	Signature by issuing REM-MD
	Identity Related Components	I00	Sender's details
		I01	Recipient's details
		I02	Recipient's delegate details
		I03	Recipient referred to by the Evidence
		I04	Sender Authentication details
		I05	Recipient Authentication details
	Messaging Components	M00	REM-MD Message/REM Dispatch details
		M01	Reply-to
		M02	Notification Message Tag
		M03	Message Submission Time
		M04	Forwarded to external system
	Extended	Enn	Space for private or public extensions to be added in the future by a set of users or by standardization bodies

Table 17: IT eDelivery - message metadata

6.5 Polish Solution (ePUAP)

The eDelivery component of ePUAP is a centralized solution based on SOAP and the open WS-* protocol stack. Thus, the solution fits in the interoperability efforts driven e.g. by the Web Services Interoperability Organisation (WS-I); a mature and modern architecture is given here with interfaces already foreseen to interconnect to external systems.

Transport level security and authentication are provided by means of WS-Security and SAML-Token - the latter is being useable for authentication when connecting external systems. Support for digital signatures on payload level is given, too; ePUAP is using TSLs for foreign signature validation.

Access to messages is given through the ePUAP web portal functionalities. Due to the underlying SOAP message format, standard e-mail-Clients can not be used for access. Particularly when connecting MIME/SMPT/POP3 based infrastructures to ePUAP, besides protocol converters addressing and authentication token mapping must be considered carefully.

6.5.1 Messages

The standard for the (signed) message format is S/MIME enveloped (multipart/signed). The message is a set of several parts: a clear text part, for the human reader, describing the message and a further set of attachments (the original message, the certification data, etc.), depending on the message type. The message set is put in a S/MIME v3 envelope (the transport envelope), digitally signed by the sender's provider

Documents are transferred with SOAP messages.

```
<binding name="skrytkaBinding" type="tns:skrytka">
        <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
        <wsp:PolicyReference URI="#x509Token"/>
        <operation name="nadaj">
            <soap:operation soapAction="http://ws.epuap.gov.pl/skrytka/nadaj"/>
            <input>
                <soap:header message="tns:nadajRequest" part="podmiot" use="literal"/>
                <soap:header message="tns:nadajRequest" part="adresSkrytki" use="literal"/>
                <soap:header message="tns:nadajRequest" part="adresOdpowiedzi" use="literal"/>
                <soap:header message="tns:nadajRequest" part="czyProbne" use="literal"/>
                <soap:header message="tns:nadajRequest" part="daneDodatkowe" use="literal"/>
                <soap:body parts="dokument" use="literal"/>
            </input>
            <output>
                <soap:body parts="odpowiedz" use="literal"/>
            </output>
            <fault name="fault">
                <soap:fault name="fault" use="literal"/>
            </fault>
        </operation>
    </binding>
```

6.5.2 Metadata summary

Name	Semantics	Format/ Reference
MessageID		
Relations [0..n]		
MessageTimeStamps		
LifeTime		
Delivered		
Fetched by recipient		
Receipted		
Security		
Security Timestamp		
SAML-Token		
Signature		
SAML-Token for recipient		
Security Token X509	wsp:Policy wsu:Id="x509Token" <wsse:TokenType> "http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-x509-token-profile-1.0#X509v3" </wsse:TokenType>	
Sender (=Reply-To address)	<part name="adresOdpowiedzi"	

Name	Semantics	Format/ Reference
	element="ob:AdresOdpowiedzi">	
Sender (Party)	<part name="podmiot" element="ob:IdentyfikatorPodmiotu"> Each sender has a context of entity – party to which he is assigned to.	
Recipient (=To address)	<part name="adresSkrytki" element="ob:AdresSkrytki">	
Subject		
Type of message		
Related BusinessScenario		
Delivery Options		
Demand for DeliveryReceipt		
Demand for ReceptionReceipt		
Demand for FetchedNorificatioon		
Is trial	<part name="czyProbne" element="ob:CzyProbne"> Sends the Message without the payload.	
Additional Information	<part name="daneDodatkowe" element="ob:DaneDodatkowe"> Additional information – any XML document.	
Payload	<part name="dokument" element="ob:Dokument"> Document payload.	

Table 18: PL eDelivery - message metadata

6.6 Overall Transport Message Metadata Summary

The following table presents the different types of metadata for each country. There has been a combination of the metadata elements in order to achieve better classification and presentation of these. The countries that support these elements are defined below.

No.	Metadata elements	Semantics	Countries
1.	**MessageID**	**Unique ID of dispatch message**	**AT, DE, GR, IT, PL**
2.	**Recipient**	**This element represents the participant identifier of the ultimate recipient.**	**AT, DE, GR, IT, PL**
3.	**Sender**	**Same as recipient**	**AT, DE, GR, IT, PL**

No.	Metadata elements	Semantics	Countries
4.	Subject	Dispatch subject (similar to e-mail for visual presentation)	AT, DE, GR, IT, PL
5.	Delivery	Delivery options	AT, DE, GR, IT, PL
6.	Security	Security options	DE, PL
7.	Message timestamps	Sending timestamps	AT, DE, GR, IT, PL
8.	Source signatures	Encapsulated signatures of original domain-specific dispatch message	AT, DE, GR, IT, PL
9.	Signature	Message signature	AT, DE, GR, IT, PL
10.	Additional information	Additional information	IT, PL

Table 19: Items of message metadata supported per eDelivery solution (piloting ones)

For each one of these elements, each country defines the metadata according to its needs. Apparently these metadata may differ from one solution to another.

References

[1] RFC 2119: Key words for use in RFCs to Indicate Requirement Levels; http://tools.ietf.org/html/rfc2119 (last visited on 08th May 2010)

[2] RFC 2046: Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types; http://tools.ietf.org/html/rfc2046 (last visited on 08th May 2010)

[3] XML Signature Syntax and Processing (Second Edition): http://www.w3.org/TR/xmldsig-core/ (last visited on 08th May 2010)

[4] RFC 1951: DEFLATE Compressed Data Format Specification version 1.3; http://tools.ietf.org/html/rfc1951 (last visited on 08th May 2010)

[5] W3C Recommendation Web Services Addressing 1.0; http://www.w3.org/TR/ws-addr-core/ (last visited on 08th May 2010)

[6] OASIS Standard WS-Security; http://www.oasis-open.org/committees/download.php/16790/wss-v1.1-spec-os-SOAPMessageSecurity.pdf (last visited on 08th March 2012)

[7] SPOCS Project, D3.1 "Assessment of eDelivery systems and specifications required for interoperability", http://www.eu-spocs.eu/index.php?option=com_processes&task=showDocument&did=198&id=18&Itemid=1 (last visited on 20th May 2010)

[8] SPOCS Project, D5.2 parts "Functional requirements for WP3 eDelivery" and "Functional requirements for WP3 eSafe", http://www.eu-spocs.eu/index.php?option=com_processes&task=streamFile&id=18&fid=761 (last visited on 20th March 2012)

[9] ETSI TS 102 231, v3.1.2, Electronic Signatures and Infrastructures (ESI); Provision of harmonized Trust-service status information; http://www.etsi.org/deliver/etsi_ts/102200_102299/102231/03.01.02_60/ts_102231v030102p.pdf (last visited on 20th May 2010)

[10] PEPPOL Deliverable D1.1 Requirements for Use of Signatures in Public Procurement Processes, Part 4: Architecture and Trust Models; http://www.peppol.eu/work_in_progress/wp-1-esignature/results/d1-1-part-4-architecture-and-trust-models (last visited on 20th May 2010)

[11] PEPPOL BusDox v. 1.0 specifications, http://www.peppol.eu/work_in_progress/wp8-Solutions%20architecture %2C%20design%20and%20validation/specifications/v1-0-specifications (last visited on 21th May 2010)

[12] ETSI TS 102 640-2 V2.2.1, Electronic Signatures and Infrastructures (ESI); Registered Electronic Mail (REM); Part 2: Data Requirements and Formats for Signed Evidences for REM; http://www.etsi.org/deliver/etsi_ts/102600_102699/10264002/02.02.01_60/ts_10264002v020201p.pdf (last visited on 26th March 2012)

[13] SOAP Message Transmission Optimization Mechanism, W3C Recommendation 25 January 2005, http://www.w3.org/TR/soap12-mtom/

[14] SOAP Version 1.2 Part 1: Messaging Framework (Second Edition), W3C Recommendation 27 April 2007, http://www.w3.org/TR/soap12-part1/

[15] Assertions and Protocol for the OASIS Security Assertion Markup Language (SAML) V1.2, OASIS Standard, 2 September 2003, http://www.oasis-open.org/committees/download.php/3406/oasis-sstc-saml-core-1.1.pdf, (last visited on 21th May 2010)

[16] Web Services Addressing 1.0 – SOAP Binding, W3C Recommendation 9 May 2006, http://www.w3.org/TR/2006/REC-ws-addr-soap-20060509/, (last visited on 21th May 2010)

[17] Web Services Security SAML Token Profile 1.1, OASIS Standard Specification incorporating Approved Errata, 1 November 2006, http://docs.oasis-open.org/wss/v1.1/wss-v1.1-spec-errata-os-SAMLTokenProfile.pdf, (last visited on 21th May 2010)

[18] Web Services Security X.509 Certificate Token Profile 1.1, OASIS Standard Specification, incorporating Approved Errata, 1 November 2006, http://docs.oasis-open.org/wss/v1.1/wss-v1.1-spec-errata-os-x509TokenProfile.pdf, (last visited on 21th May 2010)

[19] World Wide Web Consortium. XML Encryption Syntax and Processing, W3C Recommendation, 10.12.2002; http://www.w3.org/TR/2002/REC-xmlenc-core-20021210/, (last visited on 21th May 2010)

[20] World Wide Web Consortium. Extensible Markup Language (XML) 1.0 (Fourth Edition), T. Bray, J. Paoli, C. M. Sperberg-McQueen, and E. Maler, Editors. 10 February 1998, revised 16 August 2006; http://www.w3.org/TR/2006/REC-xml-20060816/, (last visited on 21th May 2010)

[21] RFC 2368, The mailto URL scheme; http://www.rfc-editor.org/rfc/rfc2368.txt

[22] RFC 2616, Hypertext Transfer Protocol -- HTTP/1.1; http://www.rfc-editor.org/rfc/rfc2616.txt, (last visited on 21th May 2010)

[23] RFC 2817, Upgrading to TLS Within HTTP/1.1; http://tools.ietf.org/html/rfc2817, (last visited on 21th May 2010)

[24] RFC 4122, A Universally Unique Identifier (UUID) URN Namespace, The Internet Engeneering Task Force July 2005, http://www.ietf.org/rfc/rfc4122.txt, (last visited on 21th May 2010)

[25] OSCI-Transport - Version 2.0, Edition 3 - Web Services Profiling and Extensions Specification, OSCI Steering Office 2010, http://www.osci.eu/transport/osci20/20100427/OSCI20_WS-ProfilingAndExtensionSpecification_Edition3.pdf, (last visited on 21th May 2010)

[26] RFC 822, Standard for the format of ARPA Internet text messages, The Internet Engeneering Task Force August 1982, http://www.ietf.org/rfc/rfc0822.txt, (last visited on 21th May 2010)

[27] ETSI TS 101 903: XML Advanced Electronic Signatures, V1.4.1 2009-06; http://pda.etsi.org/exchangefolder/ts_101903v010401p.pdf (last visited on 9th July 2010)

[28] RFC 822, Standard for the format of ARPA Internet text messages, The Internet Engineering Task Force August 1982, http://www.ietf.org/rfc/rfc0822.txt, (last visited on 21th May 2010)

[29] WS-I Basic Profile 2.0, Working Group Draft, 2007-10-25, Web Services Interoperability Organization, http://www.ws-i.org/Profiles/BasicProfile-2_0(WGD).html, (last visited on 19th July 2010)

[30] SOAP Message Transmission Optimization Mechanism, W3C Recommendation 25 January 2005, http://www.w3.org/TR/soap12-mtom/ (last visited on 27th July 2010)

[31] XML Binary Optimized Packaging, W3C Recommendation 25 January 2005, http://www.w3.org/TR/xop10/ (last visited on 27th July 2010)

[32] RFC 5322: Internet Message Format; http://tools.ietf.org/html/rfc5322 (last visited on

28th July 2010)

[33] Describing Media Content of Binary Data in XML, W3C Working Group Note, 5 May 2005, http://www.w3.org/TR/xml-media-types/ (last visited on 29th July 2010)

[34] RFC 2396: Uniform Resource Identifiers (URI): Generic Syntax, http://www.ietf.org/rfc/rfc2396.txt, (last visited on 11th August 2010)

[35] STORK D5.1.8.b - Interface Specification, 31/7/2009, https://www.eid-stork.eu/index.php?option=com_processes&Itemid=&act=streamDocument&did=960 (last visited on 08th August 2010)

[36] STORK D6.4.1 - eDelivery Functional Specification, 08/11/2009, https://www.eid-stork.eu/index.php?option=com_processes&Itemid=&act=streamDocument&did=971 (last visited on 08th August 2010)

[37] ISO 3166-1 Country Codes, Version2006, lat update 2009-10-23, http://www.tm-xml-wiki.org/wiki/TM-XML_ISO_3166_Country_Code_XSD (last visited on 18th August 2010)

[38] STORK D2.3 - STORK Quality authenticator scheme, 2009-03-03, https://www.eid-stork.eu/index.php?option=com_processes&Itemid=&act=streamDocument&did=577 (last visited on 18th August 2010)

[39] Secure Hash Standard, Federal Information Processing Standards Publication 180-2 (extended to include SHA-384, SHA-256, and SHA-512), 2002 August 1, http://csrc.nist.gov/publications/fips/fips180-2/fips180-2withchangenotice.pdf (last visited on 27th August 2010)

[40] RFC 4051, Additional XML Security Uniform Resource Identifiers, D. Eastlake 3rd, April 2005, http://www.ietf.org/rfc/rfc4051.txt (last visited on 27th August 2010)

[41] XML Signature Syntax and Processing Version 1.1, W3C Working Draft 04 February 2010, D. Eastlake 3rd et. al., http://www.w3.org/TR/2010/WD-xmldsig-core1-20100204/ (last visited on 27th August 2010)

[42] RFC3447, Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1, J. Jonsson et.al., http://www.ietf.org/rfc/rfc3447.txt (last visited on 27th August 2010)

[43] Specification for the Advanced Encryption Standard AES), 26 November 2001, http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf (last visited on 27th August 2010)

[44] eps e-payment standard Pflichtenheft V2.3, Joachim Geisler, Christian Matschi, March 2009, http://www.stuzza.at/1111_DE.6488C3D06b0d1db8f99c75d5082dbde7c4e3fa71 (last visited on 27th August 2010)

[45] ETSI TS 102 640-4 V2.1.2, Electronic Signatures and Infrastructures (ESI); Registered Electronic Mail (REM); Part 4: REM_MD Conformance Profiles; http://www.etsi.org/deliver/etsi_ts/102600_102699/10264004/02.01.02_60/ts_10264004v020102p.pdf (last visited on 26th March 2012)

[46] ETSI TS 102 640-6-3 V1.1.1, Electronic Signatures and Infrastructures (ESI); Registered Electronic Mail (REM); Part 6: Interoperability Profile: Sub-part 3: REM-MD SOAP Binding Profile; http://www.etsi.org/deliver/etsi_ts/102600_102699/1026400603/01.01.01_60/ts_1026400603v010101p.pdf (last visited on 26th March 2012)

[47] ETSI TS 102 640-2 V2.2.1, Electronic Signatures and Infrastructures (ESI); Registered Electronic Mail (REM); Part 1: Architecture; http://www.etsi.org/deliver/etsi_ts/102600_102699/10264001/02.02.01_60/ts_10264001v02

0201p.pdf (last visited on 26th March 2012)

[48] Web Services Reliable Messaging (WS-ReliableMessaging) Version 1.2, OASIS Standard, 2 February 2009, http://docs.oasis-open.org/ws-rx/wsrm/200702/wsrm-1.2-spec-os.html (last visited on 11th February 2010)

[49] Web Services Reliable Messaging Policy Assertion (WS-RM Policy) Version 1.2, OASIS Standard, 2 February 2009, http://docs.oasis-open.org/ws-rx/wsrmp/200702/wsrmp-1.2-spec-os.html (last visited on 11th February 2011)

[50] Web Services Policy 1.5 - Framework, W3C Recommendation, September 2007,. http://www.w3.org/TR/2007/REC-ws-policy-20070904 (last visited on 11th February 2011)

[51] WS-SecurityPolicy 1.3, OASIS Standard, 2 February 2009, http://docs.oasis-open.org/ws-sx/ws-securitypolicy/v1.3/os/ws-securitypolicy-1.3-spec-os.html (last visited on 11th February 2011)

[52] World Wide Web Consortium, Web Services Description Language (WSDL) Version 2.0 Part 1: Core Language, http://www.w3.org/TR/wsdl20/

All following SPOCS documents mentioned here are available at: http://www.eu-spocs.eu/index.php?option=com_processes&task=showProcess&id=18&Itemid=61

[53] SPOCS D2.2 Standard Document and Validation Common Specifications"

[54] SPOCS D3.2 Functional Specification, Architecture and Trust Model

[55] Appendix 1: Security Architecture Development Process

[56] Appendix 2: Trust-service Status List Profiling ("SPOCS TSL")

[57] Appendix 3: eDelivery Interconnect Protocol and Gateway Specification

[58] Appendix 4: eSafe – Operations in Detail

[59] Appendix 5: SPOCS TSL Accreditation and Operation Policy

[60] Appendix 6: Security Model