PEPPOL
Pan-European Public Procurement Online

PEPPOL Deliverable D1.1
Requirements for Use of Signatures in Public Procurement Processes

Part 5: XKMS v2 Interface Specification

Profiling and Extensions Specification

Version 1.2

PEPPOL WP1 2009-04-30

Borderless eProcurement

Let's make it happen!

cip

Table of Contents

1 Summary and Structure of Document ... 3
1.1 Scope and Structure of Deliverable D1.1 ... 3
1.2 Scope and Structure of This Document ... 3
1.3 Evolution of This Document ... 5
1.4 Version, List of Contributors ... 5
2 Document Conventions ... 6
2.1 Notational Conventions ... 6
2.2 XML Namespaces ... 7
3 XKMS 2.0 Restrictions ... 8
3.1 General ... 8
3.1.1 Processing Requirements ... 9
3.1.2 XKMS Message Transport ... 9
3.1.3 Message Signing Requirements and Processing Recommendations ... 9
3.1.4 Id Attributes, Identifying Requests and Responses ... 9
3.2 ValidateRequest ... 10
3.3 ValidateResult ... 11
4 Mediating XKMS Requests and Responses ... 13
4.1 Preconditions ... 13
4.2 Request Forwarding ... 13
4.3 Result Delivery ... 13
5 XKMS Extensions defined for PEPPOL ... 14
5.1 Extension for Validate Request ... 14
5.2 Extension for Validate Result ... 14
6 Indices ... 22
6.1 Tables ... 22
6.2 Pictures ... 22
6.3 References ... 22
Appendix A. Extension Schema ... 23

1 Summary and Structure of Document

1.1 Scope and Structure of Deliverable D1.1

This document is a part of the multi-part deliverable D1.1 "Requirements for Use of Signatures in the Procurement Processes" issued by the PEPPOL[1] (Pan-European Public Procurement On-Line) project. PEPPOL is a three-year (May 2008 – May 2011) large scale pilot under the CIP (Competitiveness and Innovation Programme) initiative of the European Commission.

D1.1 consists of the following documents:

Part 1: Background and Scope

Part 2: E-tendering Pilot Specifications

Part 3: Signature Policies

Part 4: Architecture and Trust Models

Part 5: XKMS v2 Interface Specification

Part 6: OASIS DSS Interface Specification

Part 7: eID and eSignature Quality Classification

The D1.1 deliverable is the first version of **functional specifications** for cross-border interoperability of e-signatures in Europe. The specifications are specifically targeted at cross-border public procurement, the topic of PEPPOL. However, if the resulting solution is successful it is believed that it will be applicable also to other application areas in need of e-signature interoperability.

Signature interoperability in PEPPOL focuses on verification of e-signatures and their associated eIDs. Interoperability of signing solutions is not handled as it is assumed that all actors are capable of signing documents within their corporate infrastructure.

The specifications guide the implementation, testing, and piloting of e-signature interoperability solutions to be done by PEPPOL. The specifications are publicly available and comments from any interested party are most welcome. Note that since the specifications of D1.1 by necessity will evolve as a result of further work in PEPPOL, any party using or referring to the specifications must ensure that the latest version is used; contact the PEPPOL project for information.

1.2 Scope and Structure of This Document

Cross-border interoperability for verification of e-signatures requires more information than merely an assessment that the signature is valid. Signature validity is just one aspect of signature acceptance, which is governed by the signature policy in force (see D1.1 part 3).

PEPPOL specifies validation services and their interfaces. A validation service must be able to assess and return information related to signature policy adherence, which necessitates a richer interface than merely OCSP or CRL for revocation checking. Two interfaces are specified:

- XKMS v2 for eID certificate validation (this document);
- OASIS DSS for verification of entire, signed documents (part 6 of D1.1).

[1] http://www.peppol.eu

The W3C "XML Key Management Specification" [XKMS], part "Key Information Service Specification" (X-KISS) has been chosen as standard interface for the validation process of X509-Certificates used for digital signatures and other purposes in the context of PEPPOL.

XKMS defines a service named "XKMS-Responder", which in the case of X-KISS is able the check the validity of X509-Certificates with regard to a given time instant and appropriate operational model – in case of certificates issued by PKI at least following relevant specifications as defined by the IETF PKIX Working Group[2]. For this scenario, a XKMS-Responder is in the role of kind of a relay

- accepting certificate validation requests on base of the XKMS protocol;
- in case of a unknown certificate issuer mediating request to other XKMS responder instances able to serve the request[3];
- checking certificates and certificate chains locally;
- connecting to issuer CAs using the respective served protocols (OCSP, CRL, LDAP…);
- if available at responder instance, including assertions on certificate quality and CSP status as outlined in according Trusted Service List (TSL) entry[4];
- building up and delivering the validation response with detailed information as defined by the XKMS protocol.

For sake of interoperability, this document defines restrictions made by PEPPOL to the relevant parts XKMS specification in chapter [3].

In addition, the XKMS extension mechanism is used to define sets of optional attributes, which seem to be valuable for already existing implementations of XKMS responders/requestors. As these extensions are seen as MS specific requirements, they should optionally be servable on a profile base. Chapter [4] outlines the extensions defined for PEPPOL. MS may define own extensions in coordination with the PEPPOL WP1 technical subgroup.

It is an assumption of PEPPOL that there will be several XKMS responder instances with different sets of CAs that can be connected directly – one imaginable XKMS Responder landscape could be a model where each member state (MS) operates a XKMS Responder instance covering connectivity to the CAs of this MS. In reality, there might be *n* specialized instances per MS or even instances covering connectivity to CAs located in different MS.

Another assumption is, a certificate validating client connects to one standard XKMS responder of his choice with trust established to this instance, which – in case of a here unknown issuer of the certificate to be validated – contacts other instances on behalf of the client. This scenario leads to the requirement that XKMS responders must be able to mediate requests to other appropriate instances. In addition, trust relationships must be federated when mediating. Chapter [4] outlines these additional requirements out of scope of the standard XKMS specification in detail.

Chapter [0] describes conventions and XML namespaces used in this document.

Sufficient knowledge of XKMS and other referenced specifications is assumed for the addressed audience of this document.

---

[2] Public-Key Infrastructure X.509 Working Group (PKIX-WG) of the Internet Engineering Task Force

3 This feature is especially defined by PEPPOL with regard to be able to reach any known CA in the EU over the initially contacted XKMS-Responder instance.

[4] Before TSLs will be available in machine readable format, it is planned to use human readable TSLs as base for according configuration entries of XKMS responders

1.3 Evolution of This Document

The following evolution of this document may be envisaged in future versions:

- Ongoing alignment with D1.1 part 6 (OASIS DSS) in order to optimize structure and semantics of statements about eIDs.
- Further alignment with D1.1 part 7 (quality assessment scheme) should be done in order to incorporate possible quality scheme standardizations in XKMS requests and responses.
- The specification should be promoted as a standard profile. PEPPOL will consider submission and follow up to W3C or OASIS; this process will necessarily lead to changes in specifications.
- Changes due to experience gained in PEPPOL and due to comments from external sources must be expected.

1.4 Version, List of Contributors

Version 1.0	2009/02/11	Complete version for internal quality assurance.
Version 1.1	2009/02/27	Submitted to PEPPOL project management, approved with comments at project management meeting 2009/03/27.
Version 1.2	2009/04/30	For publication, updated according to comments.

The following organizations, in alphabetical order, have contributed to Deliverable D1.1.

- **bremen online services, Germany,** http://www.bos-bremen.de
- **CNIPA, Italy** http://www.cnipa.it
- **DGME, French Ministry of Finance** http://www.references.modernisation.gouv.fr/
- **DNV, Norway** http://www.dnv.com

The following persons (alphabetical ordering for each participating organization) have contributed to the work:

Jörg Apitzsch	bos	Uwe Trostheide	bos	Dr. Daniele Tatti	CNIPA
Markus Ernst (co-editor)	bos	Jens Wothe	bos	Mario Terranova	CNIPA
Mark Horstmann	bos	Martine Schiavo	DGME	Anette Andresen	DNV
André Jens	bos	Stefano Arbia	CNIPA	Dr. Leif Buene	DNV
Dr. Jan Pelz	bos	Giovanni Manca	CNIPA	Jon Ølnes (editor)	DNV
Marco von der Pütten	bos	Adriano Rossi	CNIPA		

2 Document Conventions

2.1 Notational Conventions

The keywords "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

This specification uses the following syntax to define normative outlines for messages:

- The syntax appears as an XML instance, but values in italics indicate data types instead of values.
- Characters are appended to elements and attributes to indicate cardinality:
  - "?" (0 or 1)
  - "*" (0 or more)
  - "+" (1 or more)
- The character "|" is used to indicate a choice between alternatives.
- The characters "(" and ")" are used to indicate that contained items are to be treated as a group with respect to cardinality or choice.
- An ellipsis (i.e. "...“) indicates a point of extensibility that allows other child or attributes content specified in this document. Additional children elements and/or attributes MAY be added at the indicated extension points but they MUST NOT contradict the semantics of the parent and/or owner, respectively. If an extension is not recognized it SHOULD be ignored.
- XML namespace prefixes (see chapter 2.2) are used to indicate the namespace of the element being defined.

Elements and Attributes defined by this specification are referred to in the text of this document using [XPATH 1.0] expressions. Extensibility points are referred to using an extended version of this syntax:

- An element extensibility point is referred to using {any} in place of the element name. This indicates that any element name can be used, from any namespace other than the **`xkms:`** or **`xkmsEU:`** namespaces.
- An attribute extensibility point is referred to using @{any} in place of the attribute name. This indicates that any attribute name from any namespace can be used.

For those parts of this specification 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 document, thereby forming a unique requirement identifier.

If needed for clarification, indentation "(gen)" is used, when a software instance is required to support generation of a certain requirement or XML Infoset, indentation "(proc)" if processing is required; "(gen/proc)" if both.

2.2 XML Namespaces

Following XML namespaces are referenced:

Prefix	XML Namespace	Specification
ds	**http://www.w3.org/2000/09/xmldsig#**	[XMLDSIG]
xkms	**http://www.w3.org/2002/03/xkms#**	[XKMS]
xkmsEU	**http://www.lsp.eu/2009/04/xkmsExt#**	This document
xs	**http://www.w3.org/2001/XMLSchema**	[XMLSchema]

Table 1: Referenced Namespaces

The namespace choosen for the XKMS extension outlined in this document is preliminary. It is intended to align details with other large scale pilot projects which may use outcomes of PEPPOL.

3 XKMS 2.0 Restrictions

For XKMS in general and X-KISS in detail, definitions of [XKMS] apply; only deviations from the standard are outlined here.

3.1 General

R0100 - For simplification of processing and implementation, conformant XKMS requestors (gen) and responders (proc) MUST use synchronous request/response processing as defined in ([XKMS], chapter 2.4.1). For the PEPPOL pilot, asynchronous processing MUST NOT be used.[5]

R0110 - For optimization reasons, conformant XKMS requestors (gen/proc) and responders (gen/proc) MUST support compound request/responses as defined in ([XKMS], chapter 3.4).

R0110 applies in conjunction with

R0120 - Conformant XKMS implementations MUST support the validate service on base of the XML infosets **`xkms:ValidateRequest`** and **`xkms:ValidateResult`** ([XKMS], chapters 4.2 and 5.3).

These restrictions lead to the following schemas of XKMS request respective response which MUST be supported:

```
<!-- CompoundRequest -->
   <element name="CompoundRequest" type="xkms:CompoundRequestType"/>
   <complexType name="CompoundRequestType">
      <complexContent>
         <extension base="xkms:RequestAbstractType">
            <choice maxOccurs="unbounded">
               <element ref="xkms:ValidateRequest"/>
            </choice>
         </extension>
      </complexContent>
   </complexType>
<!-- /CompoundRequest -->
```

```
<!-- CompoundResult -->
   <element name="CompoundResult" type="xkms:CompoundResultType"/>
   <complexType name="CompoundResultType">
      <complexContent>
         <extension base="xkms:ResultType">
            <choice minOccurs="0" maxOccurs="unbounded">
               <element ref="xkms:ValidateResult"/>
            </choice>
         </extension>
      </complexContent>
   </complexType>
<!-- /CompoundResult -->
```

[5] Support of asynchronous processing is foreseen for a future version. For the pilot version, XKMS clients should be aware that XKMS responders used in the PEPPOL infrastructure are not obligated to support asynchronous requests.

3.1.1 Processing Requirements

R0130 - XKMS responders conformant to this profiling MUST try to obtain all missing data needed for the validation process from the underlying PKI service and hence MUST provide interfaces to underlying PKIs (both is marked optional in the XKMS specification). The *validation processing* MUST at least follow the PKIX-model as outline in [COMMPKI], Part 5: Certificate Path Validation if not otherwise defined by national regulations of the country of the certificate issuing CA.

For *CA access*, XKMS responders MUST support the interfaces as summarized in [COMMPKI] , Part 4: Operational Protocols.

3.1.2 XKMS Message Transport

R0140 - XKMS MUST be bound to SOAP 1.2 over https as defined as one option in the XKMS bindings specification [XKMSBIND].

3.1.3 Message Signing Requirements and Processing Recommendations

R0150 - For integrity protection and authentication reasons, XKMS messages MUST be signed by the respective producer. Implementations MUST ensure that all the bytes in the XKMS messages be included in hashing and in the resulting signature value of the message (see [XKMS], chapter 3.1.1); message consumers MUST validate the signatures. For compound requests and responses, the **`/xkms:CompoundRequest/ds:Signature`** respective **`/xkms:CompoundResponse/ds:Signature`** element MUST be generated, the inner **`…/ds:Signature`** elements of the contained **`…/xkms:ValidateRequest`** respective **`…/xkms:ValidateResult`** containers SHOULD NOT be generated in addition. The latter MUST be generated if simple requests/responses are used, which are not enveloped in a compound request respective response.

R0160 - XKMS signatures MUST be generated using X509 certificates, which MUST be embedded in the **`ds:Signature`** elements according to [XMLDSIG].

XKMS responders MAY decide service processing or denial on base of known the requestor certificates, which in addition may be taken for accounting issues. Responder instances MUST publish their policies concerning the regulations in effect for these issues.

For XKMS requestors, the signing certificate of the used responder is in the role of a trust anchor. Requestors MUST NOT consume response messages, for which untrusted or unknown certificates were used for message signing.

3.1.4 Id Attributes, Identifying Requests and Responses

R0170 - Following [XMLSchema], Id attributes used in a XML Infoset instance MUST have unique values. To fulfil this requirement, Id attribute values SHOULD be generated according to IETF RFC "A Universally Unique Identifier (UUID) URN Namespace" [RFC4122], whereby this value SHOULD be preceded by an underscore ("_") character[6].

R0180 - To enable requestor-side correlation of requests and responses, the values of the request **`@Id`** attributes of elements **`/xkms:CompoundRequest`** and **`/xkms:ValidateRequest`** MUST be copied to the corresponding **`@RequestId`** attributes of the **`/xkms:CompoundResult`** and **`/xkms:ValidateResult.`**[7]

[6] Values generated following [RFC4122] may have leading characters which violate the production rules of the xs:ID type

[7] [XKMS] outlines the `@RequestId` as on optional attribute

3.2 ValidateRequest

R0200 - **xkms:ValidateRequest** is an extension of **xkms:RequestAbstractType**, which itself is an extension of **xkms:MessageAbstractType**. The extensions defined by **xkms:RequestAbstractType** are defined optional. Following elements and attributes of these extensions MUST NOT be used, as they are meaningful only in the context of asynchronous processing:

@OriginalRequestId, @ResponseLimit, xkms:ResponseMechanism, xkms:PendingNotification

R0210 - The **xkms:RespondWith** extension of **xkms:RequestAbstractType** SHOULD be used to indicate the base PKI validation data required in the response. **xkms:RespondWith** is based on the URI enumeration simple type **xkms:RespondWithEnum**. Following table outlines the meaningful choices in this context, which MUST be understood by conformant XKMS responders. Other values MAY be used[8], for which standard XKMS responders are not obliged to support them:

RespondWith URI	Meaning
http://www.w3.org/2002/03/xkms#X509Cert	Return certificate (default behaviour, if no element **xkms:RespondWith** present in the request)
http://www.w3.org/2002/03/xkms#X509Chain	Return certificate chain build by responder
http://www.w3.org/2002/03/xkms#X509CRL	Return CRL acquired by responder
http://www.w3.org/2002/03/xkms#OCSP[9]	Return acquired OCSP response for validated certificate (not multiple OCSPs of the whole chain!)

Table 2: RespondWith URIs of the XKMS standard set to be supported

R0220 - Extended response information can be requested by following additional URIs; XKMS responders used in the PEPPOL context SHOULD support this functionality:

RespondWith URI	Meaning
http://www.lsp.eu/2009/04/xkmsExt#edIDQuality	Return quality of certificate and status of issuing CSP (default behaviour, if no element **xkms:RespondWith** present in the request)
http://www.lsp.eu/2009/04/xkmsExt#OCSPNoCache	Attention: If not provided, XKMS responder MAY use cached OCSP response for validation[10]

[8] This is covered by the XKMS schema, as the underlying type is a `xs:union` of defined URI enumerations and `xs:anyURI`

[9] This enumeration is not defined in [XKMS], but seen as an inevitable extension.

[10] OCSP caching may be an implementation feature to reduce network latencies

RespondWith URI	Meaning
http://www.lsp.eu/2009/04/xkmsExt#ValidationDetails	Details on validation process to be delivered

Table 3: RespondWith URIs that SHOULD be supported for extended responses

If a XKMS responder instance does not understand one of these RespondWith URIs, processing MUST continue and an entry in of <xkmsEU:ErrorExtension> MUST be generated:

```
<xkmsEU:ErrorExtension>
  <xkmsEU:Reason>
     http://lsp.eu/2009/04/reason#NotUnderstood
  </xkmsEU:Reason>
  <xkmsEU:Detail>
     copy of RespondWith URI not understood to be placed here
  </xkmsEU:Detail>
</xkmsEU:ErrorExtension>
```

R0230 - **xkms:ValidateRequest** carries an element **xkms:QueryKeyBinding**, which is an extension of **xkms:KeyBindingAbstractType**, which in case of a **xkms:ValidateRequest** MUST contain at least the **ds:KeyInfo** element.

```
<!-- KeyBindingAbstractType-->
<complexType name="KeyBindingAbstractType" abstract="true">
  <sequence>
     <element ref="ds:KeyInfo" minOccurs="1"/>
     <element ref="xkms:KeyUsage" minOccurs="0" maxOccurs="3"/>
     <element ref="xkms:UseKeyWith" minOccurs="0"
                          maxOccurs="unbounded"/>
  </sequence>
  <attribute name="Id" type="ID" use="optional"/>
</complexType>
<!-- /KeyBindingAbstractType-->
```

R0240 - **ds:KeyInfo** MUST at least carry the certificate to be validated in **ds:X590Data/ds:X509Certificate**. More information – e.g. certificate chains – MAY be supplied by the requestor. One **xkms**:**ValidateRequest** MUST carry only one end user certificate to be validated; multiple **xkms:ValidateRequest** elements SHOULD be grouped in a **xkms:CompoundRequest**, if validation of more then one certificate is required to be done within one request/response sequence (see R0110 above).

R0250 - **xkms:QueryKeyBinding** carries an optional element **xkms**:**TimeInstant**, the value outlined here is the requested time instant for which the requestor wants to check the certificate validity. If available, the requestor SHOULD supply here the time instant the certificate was applied for the cryptographic operation which is about to be verified by the requestor. In case of verifying digital signatures, the value of **xkms**:**TimeInstant** MUST be derived from the signing time instant, if available in the underlying **ds:Signature** element.

If **xkms**:**TimeInstant** is not supplied in the request, according to [XKMS] the responder has to validate the certificate on base of the responders actual server time.

3.3 ValidateResult

For the standard part of **xkms:ValidateResult**, no further detailing is made here. [XKMS] applies here without restrictions.

Following applies concerning message extension:

R0300 - If a request carries an extension with a namespace known by the contacted XKMS responder instance, the request message extension MUST be processed according to the rules defined for this extension set. Processing MAY lead to a corresponding message extension in the response.

If an extension contained in the request is bound to a namespace not known by the responder instance, processing MUST proceed ignoring this request extension; the generated response MUST outline this fact by setting the **`@ResultMinor`** attribute value of the response to **`"http://www.w3.org/2002/03/xkms#OptionalElementNotSupported"`**, even if the **`@ResultMinor`** attribute value may be set to **`"http://www.w3.org/2002/03/xkms#Success".`** In case different values for these attributes should be generated during processing covered by the XKMS standard part, these values dominate.

4 Mediating XKMS Requests and Responses

4.1 Preconditions

R1000 - If a XKMS responder instance to process a validate request because the issuer of the certificate to be validated is not known here, it MUST be able to forward the validate request to another instance able to process the request. It is an implementation detail how the appropriate routing information is made available to the forwarding responder. This information SHOULD be gathered on base of Trusted Service Lists (TSL) (D1.1 part 4).

R1010 - Trust MUST been established between the forwarding XKMS responder and validate request destination on base of known signature certificates used for message signing by the involved XKMS responder instances. Again, TSLs SHOULD serve as the anchor to establish trust.

R1020 - For the synchronous processing as restricted for this version (see R0100), all instances involved in a mediation scenario MUST NOT close network connections on application level until response delivery is acknowledged by the respective requesting instance.

4.2 Request Forwarding

R1030 - Before request forwarding, the original request has to be modified:

The **`@service`** attribute of the request message MUST set to the value of the URI to which the XKMS request is directed now.

The **`@Id`** attribute of the request message MUST reset to a newly generated value according to chapter [3.1.4]; the original value MUST be retained for further processing.

A new **`…/ds:Signature`** element MUST be provided¸ the forwarding instance MUST resign the request message after eliminating the existing **`…/ds:Signature`** element.

4.3 Result Delivery

R1040 - The responder instance the XKMS request has been directed to MUST deliver the result message to the mediating responder instance.

R1050 - The mediating responder instance MUST verify the result message signature. In case of fault or missing trust to the result messages signature, this message MUST be discarded and a new result messages MUST be generated with following fault information attributes:

`@ResultMajor=http://www.w3.org/2002/03/xkms#Receiver`

`@ResultMinor=http://www.lsp.eu/2009/04/xkms#TrustViolation`

`@Service` MUST carry the URI of the corresponding request message was directed to.

R1060 - Before the mediating responder is re-signing the result message (see R1060) and forwarding it to the initial requestor, the result message attribute

`@RequestId` MUST be set to the value of the initial request

(which MUST have been retained by the mediating instance, see R1030).

R1070 - To provide trust establishment for the initial requestor, a new **`…/ds:Signature`** element MUST be provided¸ mediating instance MUST resign the result message after eliminating the existing **`…/ds:Signature`** element.

R1080 - The mediating responder MUST NOT apply any other changes on the result message.

5 XKMS Extensions defined for PEPPOL

For XKMS messages an abstract extension point **`xkms:MessageExtension`** is foreseen to carry additional information. German regulations require detailed information on certificate quality and validity status as well as the validation process itself. Thus, a **`/xkms:ValidateResult`** SHOULD contain an extension block **`/xkmsEU:ValidateResultExtLSP`** as defined here if requested by a message extension in the respective validate request.

5.1 Extension for Validate Request

No special **`xkms:MessageExtension`** is defined; the only extensions going beyond the standard **`xkms:ValidateRequest`** are defined above with R0220 for **`xkms:RespondWith`** URIs.

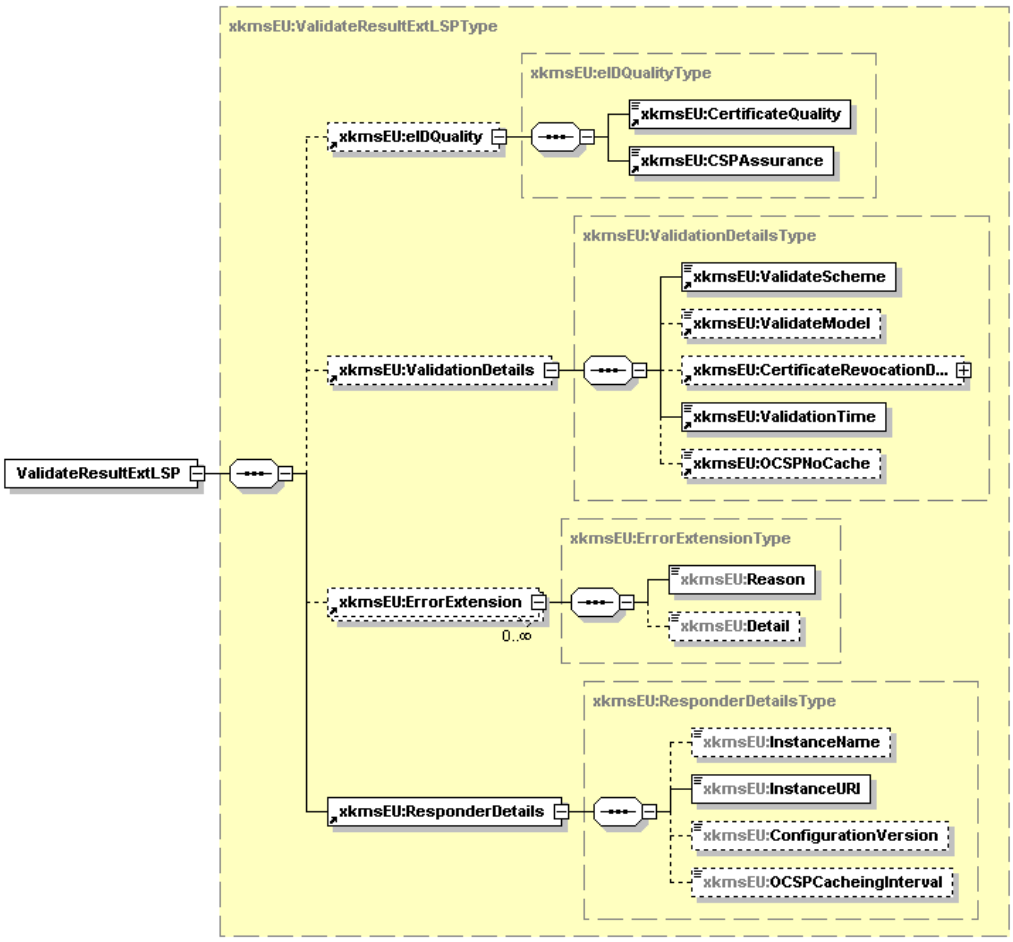
5.2 Extension for Validate Result

Extended validation information is defined for

- the quality of a certificate and the issuing CSP
- details for the validation processing done by a XKMS Responder instance
- details about the Responder itself

complemented by possible fault information concerning the processing of the extensions.

An overview is given in the following picture:



Picture 1: Extension scheme overview

Syntax for the **`xkmsEU:ValidateResultExtLSP`** element:

```
<xkmsEU:ValidateResultExtLSP>

   <xkmsEU:eIDQuality>
      <xkmsEU:CertificateQuality>
         http://lsp.eu/2009/04/certquality#unknown |
         http://lsp.eu/2009/04/certquality#low |
         http://lsp.eu/2009/04/certquality#lcp |
         http://lsp.eu/2009/04/certquality#ncp |
         http://lsp.eu/2009/04/certquality#ncpplus |
         http://lsp.eu/2009/04/certquality#qcp |
         http://lsp.eu/2009/04/certquality#qcpplus
      </xkmsEU:CertificateQuality>

      <xkmsEU:CSPAssurance>
      http://lsp.eu/2009/04/CSPAssurance#none |
      http://lsp.eu/2009/04/CSPAssurance#IndependentDocumentReview |
      http://lsp.eu/2009/04/CSPAssurance#InternalComplianceAudit |
      http://lsp.eu/2009/04/CSPAssurance#SupervisionWithComplianceAudit |
      http://lsp.eu/2009/04/CSPAssurance#ExternalComplianceAudit |
      http://lsp.eu/2009/04/CSPAssurance#ExternalComplianceAuditCertified |
      http://lsp.eu/2009/04/CSPAssurance#SupervisionWithExternalCompliance
                                                      Audit |
      http://lsp.eu/2009/04/CSPAssurance#AccreditationWithExternal
                                                      ComplianceAudit
      </xkmsEU:CSPAssurance>
   </xkmsEU:eIDQuality> ?
```

```
   <xkmsEU:ValidationDetails>
      <xkmsEU:ValidateScheme>
         http://www.lsp.eu/2009/04/valScheme#LOCAL |
         http://www.lsp.eu/2009/04/valScheme#OCSP |
         http://www.lsp.eu/2009/04/valScheme#CRL |
         http://www.lsp.eu/2009/04/valScheme#CRL_LDAP |
         http://www.lsp.eu/2009/04/valScheme#LDAP
      </xkmsEU:ValidateScheme>

      <xkmsEU:ValidateModel>
         http://www.lsp.eu/2009/04/valModel#PKIX |
         http://www.lsp.eu/2009/04/valModel#chain |
         http://www.lsp.eu/2009/04/valModel#escapeRoute |
      </xkmsEU:ValidateModel>     ?

      <xkmsEU:CertificateRevocationDetails>
         <xkmsEU:RevocationTimeInstant> xs:dateTime
         </xkmsEU:RevocationTimeInstant>
         <xkmsEU:RevocationReason>
               http://www.lsp.eu/2009/04/reason#unspecified |
               http://www.lsp.eu/2009/04/reason#KeyCompromise |
               http://www.lsp.eu/2009/04/reason#CACompromise |
               http://www.lsp.eu/2009/04/reason#AffiliationChanged |
               http://www.lsp.eu/2009/04/reason#Superseded |
               http://www.lsp.eu/2009/04/reason#CessationOfOperation |
               http://www.lsp.eu/2009/04/reason#CertificateHold |
               http://www.lsp.eu/2009/04/reason#RemoveFromCRL |
               http://www.lsp.eu/2009/04/reason#PrivilegeWithdrawn |
               http://www.lsp.eu/2009/04/reason#AACompromise |
               http://www.lsp.eu/2009/04/reason#none
         </xkmsEU:RevocationReason>
      </xkmsEU:CertificateRevocationDetails>     ?
      <xkmsEU:ValidationTime> xs:dateTime </xkmsEU:ValidationTime>
   </xkmsEI:ValidationDetails> ?

   <xkmsEU:ResponderDetails>
      <xkmsEU:InstanceName> xs:string </xkmsEU:InstanceName> ?
      <xkmsEU:InstanceUri> xs:anyUri </xkmsEU:InstanceUri>
      <xkmsEU:ConfigurationVersion> xs:string
               </xkmsEU:ConfigurationVersion> ?
      <xkmsEU:OCSPCacheInterval> xs:duration
               </xkmsEU:OCSPCacheInterval> ?
      <xkmsEU:OCSPNoCache> xs:boolean </xkmsEU:OCSPNoCache> ?
   </xkmsEU:ResponderDetails>

   <xkmsEU:ErrorExtension
      <xkmsEU:Reason=
         http://www.lsp.eu/2009/04/reason#OpaqueClientDataTooLong |
         http://www.lsp.eu/2009/04/reason#TrustCenterNotReachable |
         http://www.lsp.eu/2009/04/reason#WrongCertificateFormat |
         http://www.lsp.eu/2009/04/reason#WrongTimeInstant |
         http://www.lsp.eu/2009/04/reason#UnkownCA |
         http://www.lsp.eu/2009/04/reason#SignatureKeyTooShort
         http://www.lsp.eu/2009/04/reason#Unknown
         http://www.lsp.eu/2009/04/reason#NotUnderstood
      </xkmsEU:Reason>
      <xkmsEU:Detail> xs:string </xkmsEU:Details>
   </xkmsEU:ErrorExtension>    *

</xkmsEU:ValidateResultExtLSP> ?
```

Description of elements and attributes in the schema overview above:

/xkmsEU:ValidateResultExtLSP ?

Container element carrying all items explained below.

.../xkmsEU:eIDQuality ?

Optional container element carrying assurances on certificate quality and issuing CSP status. MUST be present if certificate validation could be processed. Explicitly requested by a **xkms:RespondWith** value of **http://www.lsp.eu/2009/04/xkmsExt#edIDQuality**

.../xkmsEU:eIDQuality/xkmsEU:CertificateQuality

Element of type **xs:anyURI** indicating the certificate quality. All values in the table below carry the prefix **http://lsp.eu/2009/04/certquality#**, which is omitted here for readability. This table corresponds to D1.1 Part 7, "eID and eSignature Quality Classification", chapter 3.2.1. For further details, see ETSI specification [ETSI101456], [ETSI102042] referenced in this table.

CertificateQuality URI ending	Meaning
unknown	Certificate quality can't be determined
low	Low confidence in certificate but certificate policy exists or quality assessment is possible by other means
lcp	Certificate governed by a Certificate Policy in compliance with the ETSI TS 102 042 standard for LCP or a similar standard
ncp	Certificate governed by a Certificate Policy in compliance with the ETSI TS 102 042 standard for NCP or a similar standard
ncpplus	Certificates governed by a Certificate Policy in compliance with the ETSI TS 102 042 standard for NCP+ or a similar standard (Use of a SSCD is mandated in the CP)
qcp	Certificates governed by a Certificate Policy in compliance with the ETSI TS 101 456 standard for QCP or a similar standard
qcpplus	Certificates governed by a Certificate Policy in compliance with the ETSI TS 101 456 standard for QCP+ or a similar standard. (Use of a SSCD is mandated in the CP)

Table 4: Quality of Certificate

.../xkmsEU:eIDQuality/xkmsEU:CSPAssurance

Element of type **xs:anyURI** indicating the certificate issuing CSP status according to D1.1 Part 7, "eID and eSignature Quality Classification", chapter 3.2.3. All values in the table below carry the prefix **http://lsp.eu/2009/04/CSPAssurance#**, which is omitted here for readability.

CSPAssurance URI ending	Meaning
none	Self assessment only

CSPAssurance URI ending	Meaning
IndependentDocument Review	Statement of compliance issued by an independent, external unit based on document review only
InternalCompliance Audit	Internal audit carried out periodically concludes compliance to applicable requirements
SupervisionWithout ComplianceAudit	CA is supervised by a public, national or international authority according to applicable law to the CA
ExternalCompliance Audit	Audit carried out periodically by external, independent auditor concludes compliance to applicable requirements
ExternalCompliance AuditCertified	Audit carried out periodically by external, independent auditor concludes compliance to applicable requirements. CA operations are certified in accordance with a relevant standard; OR cross certification with a relevant bridge CA has been made; OR the CA has obtained membership in a PKI hierarchy as a result of appropriate assessment
SupervisionWith ExternalCompliance Audit	Audit carried out periodically by external, independent auditor concludes compliance to applicable requirements. CA is supervised by a public, national or international authority according to applicable law to the CA
AccreditationWith ExternalCompliance Audit	Audit carried out periodically by external, independent auditor concludes compliance to applicable requirements. **CA is accredited** by a public, national or international authority according to applicable law to the CA

Table 5: CA Independent Assurance

.../xkmsEU:ValidationDetails ?

Optional container element carrying details on the certificate validation. MUST be present if certificate validation could be processed. Explicitely requested by a **xkms:RespondWith** value of **http://www.lsp.eu/2009/04/xkmsExt#ValidationDetails.**

.../xkmsEU:ValidationDetails/xkmsEU:ValidateScheme

Element of type **xs:anyURI** indicating the mechanism respective the protocol a certificate was validated. All values in the table below carry the prefix **http://lsp.eu/2009/04/valScheme#**, which is omitted here for readability.

ValidateScheme URI ending	Meaning

ValidateScheme URI ending	Meaning
`LOCAL`	Only local checked by responder instance
`OCSP`	Request to CA OCSP responder
`CRL`	CRL used
`CRL_LDAP`	CRL and LDAP used
`LDAP`	Request to CA LDAP certificate directory

Table 6: Certificate Validation Schemes

`.../xkmsEU:ValidationDetails/xkmsEU:ValidateModel ?`

Element of type **`xs:anyURI`** indicating the validation scheme used. All values in the table below carry the prefix **`http://lsp.eu/2009/04/valModel#`**, which is omitted here for readability.

ValidateModel URI ending	Validation Process
`PKIX`	Validation PKIX-conformant (shell-model)
`chain`	Strict certificate chain validation processing
`escapeRoute`	Mix of both above as described in [COMMPKI], part 9

Table 7: Certificate Validation Models

`.../xkmsEU:ValidationDetails/xkmsEU:CertificateRevocationDetails ?`

Container holding details in case of a certificate revoked status.

`.../xkmsEU:ValidationDetails/xkmsEU:CertificateRevocationDetails/xkmsEU:RevocationTimeInstant`

Time of revocation; type is **`xs:dateTime`**.

`.../xkmsEU:ValidationDetails/xkmsEU:CertificateRevocationDetails/xkmsEU:RevocationReason`

Element of type **`xs:anyURI`** indicating one of the following revocation reasons outlines in the table below. All values carry the prefix **`http://lsp.eu/2009/04/reason#`**, which is omitted here for readability.

RevocationReason URI ending	Meaning
`unspecific`	No specific revocation reason specified
`KeyCompromise`	User certificate is compromised

RevocationReason URI ending	Meaning
CACompromise	Issuer certificate is compromised
AffiliationChanged	Name or other attributes of certificate owner changed; certificate is not compromised
Superseded	Certificate marked as superseded; certificate is not compromised
CessationOfOperation	Certificate marked as no longer needed; certificate is not compromised
CertificateHold	Certificate withdrawn temporarily; certificate is not compromised
RemoveFromCRL	Certificate is withdrawn form CRL, reusable again
PrivilegeWithdrawn	A privilege documented in certificate is withdrawn
AACompromise	The private key of an Attribute Authority could be or is compromised
None	No revocation reason available

Table 8: Certificate Revocation Reasons

.../xkmsEU:ValidationDetails/xkmsEU:ValidationTime

Time of validation processing; element of type **xs:dateTime**.

.../xkmsEU:ValidationDetails/xkmsEU:OCSPNoCache ?

Optional element of type **xs:boolean.** MUST be reported as true, if the OSCP response was not taken from the cache.

.../xkmsEU:ResponderDetails

This container MUST be present, indicating details to the XKMS responder used, otherwise corresponding attributes of the node generating this validation result.

.../xkmsEU:ResponderDetails/xkmsEU:InstanceName ?

Optional element of type **xs:string** carrying a responder name.

.../xkmsEU:ResponderDetails/xkmsEU:InstanceUri

Mandatory element of type **xs:anyURI** carrying the responder URI.

.../xkmsEU:ResponderDetails/xkmsEU:ConfigurationVersion ?

Optional element of type **xs:string** carrying information about the responders configuration version.[11]

.../xkmsEU:ResponderDetails/xkmsEU:OCSPCacheInterval ?

Optional element of type **xs:duration.** If a responder uses cacheing for OSCP responses, the cacheing interval time SHOULD be reported here.

.../xkmsEU:ErrorExtension *

This optional element is used to report errors concerning the validation process in the attribute:

.../xkmsEU:ErrorExtension/Reason

Element of type **xs:anyURI** with following possible values; all values carry the prefix **http://lsp.eu/2009/04/reason#**, which is omitted here for readability.

ErrorExtension/Reason URI ending	Semantics
OpaqueClientData TooLong	Length of value of **/xkms:OpaqueClientData** exceeds 256 byte
TrustCenter NotReachable	Responder of certificate issuer CA not reached - time-out limit reached or other technical reasons
WrongCertificateFormat	Certificate defect or wrong coded
WrongTimeInstant	Validation time instant not recognizable or in future
UnknownCA	Certificate issuer not known
SignatureKeyTooShort	Key length of signature certificate is too short
Unknown	Error reason could not be determined
NotUnderstood	A request parameter could not be understood, but processing was (partially) possible. The indicated parameter SHOULD be outlined in the **xkmsEU:Details** element of this **xkmsEU:ErrorExtension** entry.

Table 9: XKMS Error Extension: Reasons

[11] Capabilities of a XKMS responder – i.e. OSCP-responders known by a responder instance - may depend on a concrete configuration version; this information may be helpful when checking for reasons of errors reported by a XKMS responder.

6 Indices

6.1 Tables

Table 1: Referenced Namespaces 7
Table 2: RespondWith URIs of the XKMS standard set to be supported 10
Table 3: RespondWith URIs that SHOULD be supported for extended responses 11
Table 4: Quality of Certificate 17
Table 5: CA Independent Assurance 18
Table 6: Certificate Validation Schemes 19
Table 7: Certificate Validation Models 19
Table 8: Certificate Revocation Reasons 20
Table 9: XKMS Error Extension: Reasons 21

6.2 Pictures

Picture 1: Extension scheme overview 15

6.3 References

[COMMPKI] Common PKI Specifications for interoperable Applications, Version 2.0, 20 January 2009; http://www.common-pki.org/uploads/media/Common-PKI_v2.0.pdf

[ETSI101456] ETSI: Electronic Signatures and Infrastructures (ESI); Policy Requirements for Certification Authorities issuing Qualified Certificates. ETSI TS 101 456 v1.4.1, 2006.

[ETSI102042] ETSI: Electronic Signatures and Infrastructures (ESI); Policy Requirements for Certification Authorities issuing Public Key Certificates. ETSI TS 102 042 v1.2.2, 2005

[RFC2119] S. Bradner, Key words for use in RFCs to Indicate Requirement Levels, RFC 2119, Harvard University, March 1997, http://www.ietf.org/rfc/rfc2119.txt

[RFC4122] A Universally Unique Identifier (UUID) URN Namespace, The Internet Engineering Task Force July 2005, http://www.ietf.org/rfc/rfc4122.txt

[XKMS] XML Key Management Specification (XKMS 2.0) Version 2.0, W3C Recommendation, 28 June 2005, http://www.w3.org/TR/2005/REC-xkms2-20050628/

[XKMSBIND] XML Key Management Specification (XKMS 2.0) Bindings Version 2.0, W3C Recommendation, 28 June 2005, http://www.w3.org/TR/2005/REC-xkms2-bindings-20050628/

[XMLDSIG] World Wide Web Consortium. XML-Signature Syntax and Processing (Second Edition), W3C Recommendation, 10 June 2008; http://www.w3.org/TR/xmldsig-core/

[XMLSchema] World Wide Web Consortium. XML Schema, Parts 0, 1, and 2 (Second Edition). W3C Recommendation, 28 October 2004; http://www.w3.org/TR/xmlschema-0/, http://www.w3.org/TR/xmlschema-1/, and http://www.w3.org/TR/xmlschema-2/

[XML 1.0] 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/

[XPATH 1.0] W3C Recommendation, "XML Path Language (XPath) Version 1.0," 16 November 1999; http://www.w3.org/TR/xpath

Appendix A. Extension Schema

Schema of PEPPOL XKMS Extensions

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:xkms="http://www.w3.org/2002/03/xkms#"
xmlns:xkmsEU="http://www.lsp.eu/2009/04/xkmsExt#"
targetNamespace="http://www.lsp.eu/2009/04/xkmsExt#" elementFormDefault="qualified"
attributeFormDefault="unqualified" xml:lang="EN">
   <xs:annotation>
      <xs:documentation xml:lang="en">This schema serves the reqirements of EU
Large Scale Pilot Projects regarding certificate validation as an extension to
XKMS2 XKISS ValidateResult</xs:documentation>
      <xs:documentation xml:lang="en">1.0 by Apitzsch/bos as of 2009-04-
28</xs:documentation>
   </xs:annotation>
   <xs:import namespace="http://www.w3.org/2002/03/xkms#" schemaLocation=
         "http://www.w3.org/TR/2005/REC-xkms2-20050628/Schemas/xkms.xsd"/>
   <!--ValidateResult EU LSP Extension-->
   <xs:element name="ValidateResultExtLSP" substitutionGroup=
         "xkms:MessageExtension" type="xkmsEU:ValidateResultExtLSPType"/>
   <xs:complexType name="ValidateResultExtLSPType">
      <xs:complexContent>
         <xs:extension base="xkms:MessageExtensionAbstractType">
            <xs:sequence>
               <xs:element ref="xkmsEU:eIDQuality" minOccurs="0"/>
               <xs:element ref="xkmsEU:ValidationDetails" minOccurs="0"/>
               <xs:element ref="xkmsEU:ErrorExtension"  minOccurs="0"
                                                        maxOccurs="unbounded"/>
               <xs:element ref="xkmsEU:ResponderDetails"/>
            </xs:sequence>
         </xs:extension>
      </xs:complexContent>
   </xs:complexType>
   <!-- /ValidateResultext EU LSP Extension-->
   <!-- ValidationDetails -->
   <xs:element name="ValidationDetails" type="xkmsEU:ValidationDetailsType"/>
   <xs:complexType name="ValidationDetailsType">
      <xs:sequence>
         <xs:element ref="xkmsEU:ValidateScheme"/>
         <xs:element ref="xkmsEU:ValidateModel" minOccurs="0"/>
         <xs:element ref="xkmsEU:CertificateRevocationDetails" minOccurs="0"/>
         <xs:element ref="xkmsEU:ValidationTime"/>
         <xs:element ref="xkmsEU:OCSPNoCache" minOccurs="0"/>
      </xs:sequence>
   </xs:complexType>
   <!-- /ValidationDetails -->
   <!-- ValidateScheme -->
   <xs:element name="ValidateScheme" type="xkmsEU:ValidateSchemeType"/>
   <xs:simpleType name="ValidateSchemeType">
      <xs:restriction base="xs:anyURI">
         <xs:enumeration value="http://lsp.eu/2009/04/valScheme#LOCAL"/>
         <xs:enumeration value="http://lsp.eu/2009/04/valScheme#OCSP"/>
         <xs:enumeration value="http://lsp.eu/2009/04/valScheme#CRL"/>
         <xs:enumeration value="http://lsp.eu/2009/04/valScheme#CRL_LDAP"/>
```

```
            <xs:enumeration value="http://lsp.eu/2009/04/valScheme#LDAP"/>
        </xs:restriction>
    </xs:simpleType>
    <!-- /ValidateScheme -->
    <!--ValidateModel-->
    <xs:element name="ValidateModel" type="xkmsEU:ValidateModelType"/>
    <xs:simpleType name="ValidateModelType">
        <xs:restriction base="xs:anyURI">
            <xs:enumeration value="http://lsp.eu/2009/04/valModel#PKIX"/>
            <xs:enumeration value="http://lsp.eu/2009/04/valModel#chain"/>
            <xs:enumeration value="http://lsp.eu/2009/04/valModel#escapeRoute"/>
        </xs:restriction>
    </xs:simpleType>
    <!--/ValidateModel-->
    <!--ValidationTime-->
    <xs:element name="ValidationTime" type="xs:dateTime"/>
    <!--/ValidationTime-->
    <!-- OCSPNoCache -->
    <xs:element name="OCSPNoCache" type="xs:boolean"/>
    <!-- /OCSPNoCache -->
    <!-- CertificateRevocationDetail -->
    <xs:element name="CertificateRevocationDetails">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="RevocationTimeInstant" type="xs:dateTime"/>
                <xs:element ref="xkmsEU:RevocationReason"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="RevocationReason" type="xkmsEU:RevocationReasonType"/>
    <xs:simpleType name="RevocationReasonType">
        <xs:restriction base="xs:anyURI">
            <xs:enumeration value="http://lsp.eu/2009/04/xkmsextLSP#unspecified"/>
            <xs:enumeration value="http://lsp.eu/2009/04/xkmsextLSP#KeyCompromise"/>
            <xs:enumeration value="http://lsp.eu/2009/04/xkmsextLSP#CACompromise"/>
            <xs:enumeration value=
                "http://lsp.eu/2009/04/xkmsextLSP#AffiliationChanged"/>
            <xs:enumeration value="http://lsp.eu/2009/04/xkmsextLSP#Superseded"/>
            <xs:enumeration value=
                "http://lsp.eu/2009/04/xkmsextLSP#CessationOfOperation"/>
            <xs:enumeration value="http://lsp.eu/2009/04/xkmsextLSP#CertificateHold"/>
            <xs:enumeration value="http://lsp.eu/2009/04/xkmsextLSP#RremoveFromCRL"/>
            <xs:enumeration value=
                "http://lsp.eu/2009/04/xkmsextLSP#PrivilegeWithdrawn"/>
            <xs:enumeration value="http://lsp.eu/2009/04/xkmsextLSP#AACompromise"/>
            <xs:enumeration value="http://lsp.eu/2009/04/xkmsextLSP#none"/>
        </xs:restriction>
    </xs:simpleType>
    <!-- /CertificateRevocationDetail -->
    <!-- CertificateQuality -->
    <xs:element name="CertificateQuality" type="xkmsEU:CertificateQualityType"/>
    <xs:simpleType name="CertificateQualityType">
        <xs:restriction base="xs:anyURI">
            <xs:enumeration value="http://lsp.eu/2009/04/certquality#unknown"/>
            <xs:enumeration value="http://lsp.eu/2009/04/certquality#low"/>
            <xs:enumeration value="http://lsp.eu/2009/04/certquality#lcp"/>
            <xs:enumeration value="http://lsp.eu/2009/04/certquality#ncp"/>
            <xs:enumeration value="http://lsp.eu/2009/04/certquality#ncpplus"/>
            <xs:enumeration value="http://lsp.eu/2009/04/certquality#qcp"/>
            <xs:enumeration value="http://lsp.eu/2009/04/certquality#qcpplus"/>
        </xs:restriction>
    </xs:simpleType>
    <!-- /CertificateQuality -->
    <!-- CSP independent assurance -->
    <xs:element name="CSPAssurance" type="xkmsEU:CSPAssuranceType"/>
    <xs:simpleType name="CSPAssuranceType">
```

```
      <xs:restriction base="xs:anyURI">
         <xs:enumeration value="http://lsp.eu/2009/04/CSPAssurance#none"/>
         <xs:enumeration value=
         "http://lsp.eu/2009/04/CSPAssurance#IndependentDocument Review"/>
         <xs:enumeration value=
         "http://lsp.eu/2009/04/CSPAssurance#lInternalComplianceAudit"/>
         <xs:enumeration value=
         "http://lsp.eu/2009/04/CSPAssurance#SupervisionWithoutComplianceAudit:"/>
         <xs:enumeration value=
         "http://lsp.eu/2009/04/CSPAssurance#ExternalComplianceAudit"/>
         <xs:enumeration value=
         "http://lsp.eu/2009/04/CSPAssurance#ExternalComplianceAuditCertified"/>
         <xs:enumeration value=
      "http://lsp.eu/2009/04/CSPAssurance#SupervisionWithExternalComplianceAudit"/>
         <xs:enumeration value=
   "http://lsp.eu/2009/04/CSPAssurance#AccreditationWithExternalComplianceAudit"/>
      </xs:restriction>
   </xs:simpleType>
   <!-- /CertificateQuality -->
   <xs:element name="eIDQuality" type="xkmsEU:eIDQualityType"/>
   <xs:complexType name="eIDQualityType">
      <xs:sequence>
         <xs:element ref="xkmsEU:CertificateQuality"/>
         <xs:element ref="xkmsEU:CSPAssurance"/>
      </xs:sequence>
   </xs:complexType>
   <!--ResponderDetails-->
   <xs:element name="ResponderDetails" type="xkmsEU:ResponderDetailsType"/>
   <xs:complexType name="ResponderDetailsType">
      <xs:sequence>
         <xs:element name="InstanceName" type="xs:string" minOccurs="0"/>
         <xs:element name="InstanceURI" type="xs:anyURI"/>
         <xs:element name="ConfigurationVersion" type="xs:string" minOccurs="0"/>
         <xs:element name="OCSPCacheingInterval" type="xs:duration" minOccurs="0"/>
      </xs:sequence>
   </xs:complexType>
   <!--/ResponderDetails-->
   <!-- ErrorExtension -->
   <xs:element name="ErrorExtension" type="xkmsEU:ErrorExtensionType"/>
   <xs:complexType name="ErrorExtensionType">
      <xs:sequence>
         <xs:element name="Reason" type="xkmsEU:ReasonType"/>
         <xs:element name="Detail" type="xs:string" minOccurs="0"/>
      </xs:sequence>
   </xs:complexType>
   <xs:simpleType name="ReasonType">
      <xs:restriction base="xs:anyURI">
         <xs:enumeration value=
         "http://lsp.eu/2009/04/reason#OpaqueClientDataTooLong"/>
         <xs:enumeration value=
         "http://lsp.eu/2009/04/reason#TrustCenterNotReachable"/>
         <xs:enumeration value=
         "http://lsp.eu/2009/04/reason#WrongCertificateFormat"/>
         <xs:enumeration value="http://lsp.eu/2009/04/reason#UnknownCA"/>
         <xs:enumeration value="http://lsp.eu/2009/04/reason#WrongTimeInstant"/>
         <xs:enumeration value=
         "http://lsp.eu/2009/04/reason#SignatureKeyTooShort"/>
         <xs:enumeration value="http://lsp.eu/2009/04/reason#Unknown"/>
         <xs:enumeration value="http://lsp.eu/2009/04/reason#NotUnderstood"/>
         <xs:enumeration value=""/>
      </xs:restriction>
   </xs:simpleType>
   <!-- /errorExtension -->
   <!-- /XKISS EU LSP Extension  End Schema -->
</xs:schema>
```