



## Electronic Signatures and Infrastructures (ESI); Signature Policies; Part 2: XML format for signature policies

**STABLE DRAFT**

Send comments **ONLY** to [E-SIGNATURES\\_COMMENTS@list.etsi.org](mailto:E-SIGNATURES_COMMENTS@list.etsi.org)

by 31 AUGUST 2019

Download the template for comments:

[https://docbox.etsi.org/ESI/Open/Latest Drafts/Template-for-comments.doc](https://docbox.etsi.org/ESI/Open/Latest_Drafts/Template-for-comments.doc)

CAUTION: This **DRAFT** document is provided for information and is for future development work within the ETSI Technical Committee ESI only. ETSI and its Members accept no liability for any further use/implementation of this Specification.

Approved and published specifications and reports shall be obtained exclusively via the ETSI Documentation Service at

<http://www.etsi.org/standards-search>

---

Reference

DTS/ESI-0019172-2

---

Keywords

XML, signature policy, electronic signature, e-commerce, trust services

**ETSI**

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

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

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

---

**Important notice**

The present document can be downloaded from:  
<http://www.etsi.org/standards-search>

The present document may be made available in electronic versions and/or in print. The content of any electronic and/or print versions of the present document shall not be modified without the prior written authorization of ETSI. In case of any existing or perceived difference in contents between such versions and/or in print, the prevailing version of an ETSI deliverable is the one made publicly available in PDF format at [www.etsi.org/deliver](http://www.etsi.org/deliver).

Users of the present document should be aware that the document may be subject to revision or change of status. Information on the current status of this and other ETSI documents is available at <https://portal.etsi.org/TB/ETSIDeliverableStatus.aspx>.

If you find errors in the present document, please send your comment to one of the following services:  
<https://portal.etsi.org/People/CommiteeSupportStaff.aspx>

---

**Copyright Notification**

No part may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm except as authorized by written permission of ETSI.

The content of the PDF version shall not be modified without the written authorization of ETSI.  
The copyright and the foregoing restriction extend to reproduction in all media.

© ETSI 2019.  
All rights reserved.

**DECT™**, **PLUGTESTS™**, **UMTS™** and the ETSI logo are trademarks of ETSI registered for the benefit of its Members.  
**3GPP™** and **LTE™** are trademarks of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners.  
**oneM2M™** logo is a trademark of ETSI registered for the benefit of its Members and of the oneM2M Partners.  
**GSM®** and the GSM logo are trademarks registered and owned by the GSM Association.

# Contents

Intellectual Property Rights .....	6
Foreword.....	6
Modal verbs terminology .....	6
Executive summary .....	6
1 Scope .....	7
2 References .....	7
2.1 Normative references .....	7
2.2 Informative references .....	7
3 Definition of terms, symbols and abbreviations.....	8
3.1 Terms .....	8
3.2 Symbols .....	8
3.3 Abbreviations.....	8
4 XML syntax for machine processable signature policy document .....	9
4.1 Introduction.....	9
4.1.1 Technical approach .....	9
4.1.2 XML namespaces .....	9
4.2 The <code>SignaturePolicy</code> element.....	9
4.2.1 Semantics .....	9
4.2.2 Syntax.....	10
4.3 The <code>Digest</code> element .....	10
4.3.1 Semantics .....	10
4.3.2 Syntax.....	10
4.4 The <code>PolicyComponents</code> element .....	10
4.4.1 Semantics .....	10
4.4.2 Syntax.....	11
4.5 The <code>GeneralDetails</code> element .....	11
4.5.1 Semantics .....	11
4.5.2 Syntax.....	11
4.6 The <code>SigPolicyDetails</code> element .....	11
4.6.1 Semantics .....	11
4.6.2 Syntax.....	12
4.7 The <code>VersionInfo</code> element .....	12
4.7.1 Semantics .....	12
4.7.2 Syntax.....	12
4.8 The <code>AuthorityDetails</code> element .....	13
4.8.1 Semantics .....	13
4.8.2 Syntax.....	13
4.9 The <code>Name</code> element.....	13
4.9.1 Semantics .....	13
4.9.2 Syntax.....	14
4.10 The <code>TradeName</code> element .....	14
4.10.1 Semantics .....	14
4.10.2 Syntax.....	14
4.11 The <code>PostalAddresses</code> element.....	14
4.11.1 Semantics .....	14
4.11.2 Syntax.....	14
4.12 The <code>ElectronicAddresses</code> element .....	14
4.12.1 Semantics .....	14
4.12.2 Syntax.....	14
4.13 The <code>ContactPersons</code> element .....	14
4.13.1 Semantics .....	14
4.13.2 Syntax.....	15
4.14 The <code>OtherDetails</code> element.....	15

4.14.1	Semantics .....	15
4.14.2	Syntax.....	15
4.15	The OtherPolicies element .....	16
4.15.1	Semantics .....	16
4.15.2	Syntax.....	16
4.16	The PolicyRules element .....	16
4.16.1	Semantics .....	16
4.16.2	Syntax.....	17
4.17	The CommitmentRules element.....	18
4.17.1	Semantics .....	18
4.17.2	Syntax.....	18
4.18	The DataToBeSignedRules element .....	19
4.18.1	Semantics .....	19
4.18.2	Syntax.....	19
4.19	The SigToDTBSRelationRules element .....	19
4.19.1	Semantics .....	19
4.19.2	Syntax.....	20
4.20	The DTBSCardinality element.....	20
4.20.1	Semantics .....	20
4.20.2	Syntax.....	20
4.21	The SigDTBSRelativePosition element.....	21
4.21.1	Semantics .....	21
4.21.2	Syntax.....	21
4.22	The SigFormatsAndLevels element .....	22
4.22.1	Semantics .....	22
4.22.2	Syntax.....	22
4.23	The AugmentationRules element .....	22
4.23.1	Semantics .....	22
4.23.2	Syntax.....	23
4.24	Types for defining constraints on certificates' trust.....	23
4.24.1	Introduction .....	23
4.24.2	TrustAnchorsListType type .....	23
4.24.2.1	Semantics .....	23
4.24.2.2	Syntax.....	24
4.24.3	NameConstraintsType type.....	24
4.24.3.1	Semantics .....	24
4.24.3.2	Syntax.....	25
4.24.4	PolicyConstraintsType type.....	25
4.24.4.1	Semantics .....	25
4.24.4.2	Syntax.....	25
4.24.5	CertificateTrustTreesType type .....	26
4.24.5.1	Semantics .....	26
4.24.5.2	Syntax.....	26
4.25	Types for defining constraints on certificates' revocation status .....	27
4.25.1	Introduction .....	27
4.25.2	CertificateRevReqType type.....	27
4.25.2.1	Semantics .....	27
4.25.2.2	Syntax.....	28
4.25.3	CertificateRevTrustType type.....	28
4.25.3.1	Semantics .....	28
4.25.3.2	Syntax.....	28
4.26	The SigningCertTrustCondition element.....	29
4.26.1	Semantics .....	29
4.26.2	Syntax.....	29
4.27	The TimeEvidencesRules element.....	29
4.27.1	Semantics .....	29
4.27.2	Syntax.....	30
4.28	The RolesConstraints element .....	30
4.28.1	Semantics .....	30
4.28.2	Syntax.....	31

4.29	The QualifyingPropertiesRules element.....	32
4.29.1	Semantics .....	32
4.29.2	Syntax.....	32
4.30	The SCDLoARules element .....	33
4.30.1	Semantics .....	33
4.30.2	Syntax.....	33
4.31	The CryptoSuitesRules element.....	34
4.31.1	Semantics .....	34
4.31.2	Syntax.....	34
4.32	The OtherRules element.....	34
4.32.1	Semantics .....	34
4.32.2	Syntax.....	35
<b>Annex A (normative): URIs for identifying signature formats .....</b>		<b>36</b>
A.1	URIs to signature formats mapping .....	36
<b>Annex B (normative): XML Schema file .....</b>		<b>37</b>
B.1	XML Schema file location for namespace http://uri.etsi.org/01910202/v1.1.1# .....	37
B.1	First clause of the annex .....	<b>Error! Bookmark not defined.</b>
B.1.1	First subdivided clause of the annex .....	<b>Error! Bookmark not defined.</b>
History .....		38

---

## Intellectual Property Rights

### Essential patents

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

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

### Trademarks

The present document may include trademarks and/or tradenames which are asserted and/or registered by their owners. ETSI claims no ownership of these except for any which are indicated as being the property of ETSI, and conveys no right to use or reproduce any trademark and/or tradename. Mention of those trademarks in the present document does not constitute an endorsement by ETSI of products, services or organizations associated with those trademarks.

---

## Foreword

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

The present document is part 2 of a multi-part deliverable. Full details of the entire series can be found in part 1 [i.3].

---

## Modal verbs terminology

In the present document "**shall**", "**shall not**", "**should**", "**should not**", "**may**", "**need not**", "**will**", "**will not**", "**can**" and "**cannot**" are to be interpreted as described in clause 3.2 of the [ETSI Drafting Rules](#) (Verbal forms for the expression of provisions).

"**must**" and "**must not**" are **NOT** allowed in ETSI deliverables except when used in direct citation.

---

## Executive summary

**TBC**

---

# 1 Scope

The present document defines an XML format of machine readable signature policies based on the building blocks and table of contents for human readable signature policy documents as described in ETSI TS 119 172-1 [i.3].

The present document defines elements which can be used to describe signature creation constraints, signature validation constraints and signature augmentation constraints.

For each element of the machine readable signature policy, the present document specifies the semantic and the how to implement it in XML syntax.

Pure signature applicability rules are out of the scope of the present document.

NOTE: The first version of an XML format for signature policies was described in ETSI TR 102 038 [i.1]. That document was written before the framework for standardization of signatures was described. The present document tries to use elements from ETSI TR 102 038 [i.1] where possible to reduce the implementation effort when changing to the present specification.

---

## 2 References

### 2.1 Normative references

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

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

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

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

- [1] ETSI TS 119 612: "Electronic Signatures and Infrastructures (ESI); Trusted Lists".
- [2] IETF RFC 3061: "A URN Namespace of Object Identifiers".
- [3] IETF RFC 5280: "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile".
- [4] Recommendation ITU-T X.509: "Information technology - Open Systems Interconnection - The Directory: Public-key and attribute certificate frameworks".
- [5] ETSI TS 119 442: "Electronic Signatures and Infrastructures (ESI); Protocol profiles for trust service providers providing AdES digital signature validation services".

### 2.2 Informative references

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

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

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

- [i.1] ETSI TR 102 038: " TC Security - Electronic Signatures and Infrastructures (ESI); XML format for signature policies".
- [i.2] ETSI TS 119 102-1 (V1.2.1): "Electronic Signatures and Infrastructures (ESI); Procedures for Creation and Validation of AdES Digital Signatures; Part 1: Creation and Validation".
- [i.3] ETSI TS 119 172-1:"Electronic Signatures and Infrastructures (ESI); Signature Policies; Part 1: Building blocks and table of contents for human readable signature policy documents".
- [i.4] ETSI TS 102 231: "Electronic Signatures and Infrastructures (ESI); Provision of harmonized Trust-service status information".
- [i.5] ETSI TS 119 612: "Electronic Signatures and Infrastructures (ESI); Trusted Lists".

---

## 3 Definition of terms, symbols and abbreviations

### 3.1 Terms

For the purposes of the present document, the terms given in ETSI TS 119 172-1:"Electronic Signatures and Infrastructures (ESI); Signature Policies; Part 1: Building blocks and table of contents for human readable signature policy documents" [i.3] and the following] apply:

**signature applicability rules:** set of rules, applicable to one or more digital signatures, that defines the requirements for determination of whether a signature is fit for a particular business or legal purpose

**signature augmentation constraint:** criteria used when augmenting a digital signature

**signature augmentation policy:** set of signature augmentation constraints

**signature creation application:** application within the signature creation system that creates the AdES digital signature and relies on the signature creation device to create a digital signature value

**signature creation constraint:** criteria used when creating a digital signature

**signature creation policy:** set of **signature creation constraints** processed or to be processed by the signature creation application

**signature validation application:** application that validates a signature against a signature validation policy, and that outputs a status indication (i.e. the signature validation status) and a signature validation report

**signature validation policy:** set of signature validation constraints processed or to be processed by the signature validation applicaiton

**signature validation constraint:** technical criteria against which a digital signature can be validated, e.g. as specified in ETSI TS 119 102-1 [i.2]

### 3.2 Symbols

For the purposes of the present document, the [following] symbols [given in ... and the following] apply:

### 3.3 Abbreviations

For the purposes of the present document, the following abbreviations apply:

BSP                      Business Scoping parameter



## 4 XML syntax for machine processable signature policy document

### 4.1 Introduction

#### 4.1.1 Technical approach

#### 4.1.2 XML namespaces

The present document uses the URI namespaces listed below:

- <http://uri.etsi.org/01917202/v1.1.1#>
- <http://www.w3.org/2000/09/xmldsig#>
- <http://www.w3.org/2001/XMLSchema>

ETSI defines one XML Schema file for the present specification, namely: "XMLSCHEMANAMEFILE.xsd". See Annex B for details on their locations.

Table 1 shows the URIs of the namespaces used in the XML Schema definitions, mapped to their corresponding prefixes.

**Table 1: Namespaces with constant prefixes**

XML Namespace URI	Prefix
<a href="http://www.w3.org/2000/09/xmldsig#">http://www.w3.org/2000/09/xmldsig#</a>	ds
<a href="http://www.w3.org/2001/XMLSchema">http://www.w3.org/2001/XMLSchema</a>	xs
<a href="http://uri.etsi.org/02231/v2#">http://uri.etsi.org/02231/v2#</a>	tsl
<a href="http://uri.etsi.org/2038/v.1.1.1#">http://uri.etsi.org/2038/v.1.1.1#</a>	sptr

Below follows a copy of the `xs:schema` element of the XML Schema file "XMLSCHEMANAMEFILE.xsd", whose location is detailed in clause B.1, and that defines the namespace whose URI is <http://uri.etsi.org/01910202/v1.1.1#>.

```
<xs:schema targetNamespace="http://uri.etsi.org/1917202/v1.1.1#"
xmlns="http://uri.etsi.org/1917202/v1.1.1#" xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:ds="http://www.w3.org/2000/09/xmldsig#" xmlns:sptr="http://uri.etsi.org/2038/v.1.1.1#"
elementFormDefault="qualified" attributeFormDefault="unqualified">

  <xs:import namespace="http://uri.etsi.org/02231/v2#" schemaLocation="ts_119612v020101_xsd.xsd"/>
  <xs:import namespace="http://uri.etsi.org/2038/v.1.1.1#"
schemaLocation="foo_tr_1020380v10101_xsd.xsd"/>
  <xs:import namespace="http://www.w3.org/2000/09/xmldsig#"
schemaLocation="http://www.w3.org/TR/2008/REC-xmldsig-core-20080610/xmldsig-core-schema.xsd"/>
```

**EDITOR'S NOTE: POINTERS TO THE RIGHT LOCATIONS SHALL BE INCLUDED LATER**

## 4.2 The SignaturePolicy element

### 4.2.1 Semantics

This element shall contain

- 1) A digest for securing the contents of the signature policy document, and an identifier of the digest algorithm used for computing it, as specified in clause 4.3.

- 2) An element that contains all the machine-processable components of the signature policy, as specified in clause 4.4.

## 4.2.2 Syntax

The `SignaturePolicy` element shall be as defined in XML Schema file "XMLSCHEMANAMEFILE.xsd", whose location is detailed in clause B.1, and is copied below for information.

```
<xs:element name="SignaturePolicy" type="SignaturePolicyType" />

  <xs:complexType name="SignaturePolicyType">
    <xs:sequence>
      <xs:element ref="Digest" />
      <xs:element ref="PolicyComponents" />
    </xs:sequence>
  </xs:complexType>
```

## 4.3 The Digest element

### 4.3.1 Semantics

This element shall contain:

- 1) One identifier of a digest algorithm.
- 2) One digest value.

In the case that the structured document is an XML document, this element shall also contain:

- 1) A canonicalization algorithm.

### 4.3.2 Syntax

The `Digest` element shall be as defined in XML Schema file "XMLSCHEMANAMEFILE.xsd", whose location is detailed in clause B.1, and is copied below for information.

```
<xs:element name="Digest" type="DigestDetailsType"/>
<xs:complexType name="DigestDetailsType">
  <xs:sequence>
    <xs:element ref="ds:DigestMethod" />
    <xs:element ref="ds:DigestValue" />
    <xs:element ref="ds:CanonicalizationMethod" />
  </xs:sequence>
</xs:complexType>
```

The digest value shall be computed on the output of applying the canonicalization algorithm identified in this component, to the `PolicyComponents` element specified in clause 4.4.

## 4.4 The PolicyComponents element

### 4.4.1 Semantics

This element shall contain:

- 1) An element including general details, as specified in clause 4.5
- 2) An element containing all the rules defined by the signature policy itself, as specified in clause 4.16.

This element may also contain:

- 1) A list of identifiers and names of other signature policies incorporated or referenced by the signature policy that is the subject of the machine processable document, as specified in clause 4.15.

## 4.4.2 Syntax

The `PolicyComponents` element shall be as defined in XML Schema file "XMLSCHEMANAMEFILE.xsd", whose location is detailed in clause B.1, and is copied below for information.

```
<xs:element name="PolicyComponents" type="PolicyComponentsType"/>
<xs:complexType name="PolicyComponentsType">
  <xs:sequence>
    <xs:element ref="GeneralDetails"/>
    <xs:element ref="OtherPolicies" minOccurs="0" />
    <xs:element ref="PolicyRules"/>
  </xs:sequence>
</xs:complexType>
```

## 4.5 The GeneralDetails element

### 4.5.1 Semantics

This element shall contain:

- 1) An element containing details on the signature policy itself, as specified in clause 4.6.
- 2) An element containing details of the responsible authority of the signature policy, as specified in clause 4.8.

This element may also contain:

- 1) An element containing other details not defined within the present document, as specified in clause 4.14.

### 4.5.2 Syntax

The `GeneralDetails` element shall be as defined in XML Schema file "XMLSCHEMANAMEFILE.xsd", whose location is detailed in clause B.1, and is copied below for information.

```
<xs:element name="GeneralDetails" type="GeneralDetailsType"/>
<xs:complexType name="GeneralDetailsType">
  <xs:sequence>
    <xs:element ref="SigPolicyDetails"/>
    <xs:element ref="AuthorityDetails"/>
    <xs:element ref="OtherDetails" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
```

## 4.6 The SigPolicyDetails element

### 4.6.1 Semantics

This element shall contain:

- 1) An element identifying the signature policy itself.
- 2) An element whose value shall be the name of the signature policy itself.

This element may also contain:

- 1) An element containing one or more distribution points of the signature policy itself.
- 2) An element providing versioning information, as specified in clause 4.7.

## 4.6.2 Syntax

The `SigPolicyDetails` element and the `SigPolicyDetailsType` type shall be as defined in XML Schema file "XMLSCHEMANAMEFILE.xsd", whose location is detailed in clause B.1, and is copied below for information.

```
<xs:element name="SigPolicyDetails" type="SigPolicyDetailsType"/>

<xs:complexType name="SigPolicyDetailsType">
  <xs:sequence>
    <xs:element name="PolicyIdentifier" type="xs:anyURI"/>
    <xs:element name="PolicyName" type="tsl:InternationalNamesType"/>
    <xs:element ref="tsl:DistributionPoints" minOccurs="0"/>
    <xs:element ref="VersionInfo" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
```

The `PolicyIdentifier` child element shall contain an URI identifying the signature policy. If the signature policy is identified by an OID, the value of `PolicyIdentifier` child element shall be the OID value encoded as an URN as specified by the IETF RFC 3061 [2].

The `PolicyName` child element shall contain one or more names of the signature policy. Each name shall be qualified with an indication of a language.

NOTE 1: See the specification of the semantics and the syntax of `InternationalNamesType` type in ETSI TS 119 612 [1].

If present, the `tsl:DistributionPoints` child element shall contain one or more URIs where the signature policy can be reached.

NOTE 2: See the specification of the semantics and the syntax of `tsl:DistributionPoints` element in ETSI TS 119 612 [1].

If present, the `VersionInfo` child element is specified in clause 4.7.

## 4.7 The `VersionInfo` element

### 4.7.1 Semantics

This element shall contain an element whose value is the version of the signature policy.

This element may also contain sequence of elements each one providing information of one previous version of the signature policy. Each of the elements with information of one previous version:

- 1) shall contain the value of the corresponding previous version, and
- 2) may also contain a non-empty sequence of points where that version of signature policy can be reached.

### 4.7.2 Syntax

The `VersionInfo` element shall be as defined in XML Schema file "XMLSCHEMANAMEFILE.xsd", whose location is detailed in clause B.1, and is copied below for information.

```
<xs:element name="VersionInfo" type="VersionInfoType"/>
<xs:complexType name="VersionInfoType">
  <xs:sequence>
    <xs:element name="ThisVersion" type="xs:string"/>
    <xs:element name="PreviousVersion" type="VersionAndDistributionType" minOccurs="0"
maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="VersionAndDistributionType">
  <xs:sequence>
    <xs:element name="Version" type="xs:string"/>
    <xs:element ref="tsl:DistributionPoints" minOccurs="0" />
  </xs:sequence>
```

```
</xs:complexType>
```

The `ThisVersion` child element shall contain the version of this signature policy.

The `PreviousVersion` children elements shall contain versioning information of previous versions of this signature policy.

The `Version` child element shall contain the version value of the identified previous version.

The `tsl:DistributionPoints` child element shall contain the distribution points of the previous version.

## 4.8 The `AuthorityDetails` element

### 4.8.1 Semantics

This element shall contain details of the authority responsible for the signature policy document that is the subject of the machine processable document.

This element shall contain at least, one of the two following components:

- 1) An element including the name of the authority as specified in clause 4.9.
- 2) An element as specified in clause 4.10, including an official registration identifier, unambiguously identifying the authority, as registered in official records, where such a registered identifier does exist.

This element shall also contain the following components:

- 1) An element including a sequence of one or more postal addresses of the authority as specified in clause 4.11.
- 2) An element including one or more electronic addresses of the authority as specified in clause 4.12.

This element may also contain:

- 1) An element for providing details of one or more natural persons acting as contact as specified in clause 4.13.

### 4.8.2 Syntax

The `Authority` element shall be as defined in XML Schema file "XMLSCHEMANAMEFILE.xsd", whose location is detailed in clause B.1, and is copied below for information.

```
<xs:element name="AuthorityDetails" type="AuthorityDetailsType"/>
<xs:complexType name="AuthorityDetailsType">
  <xs:sequence>
    <xs:element ref="Name" minOccurs="0"/>
    <xs:element ref="TradeName" minOccurs="0"/>
    <xs:element ref="PostalAddresses" />
    <xs:element ref="ElectronicAddresses" />
    <xs:element ref="ContactPersons" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
```

## 4.9 The `Name` element

### 4.9.1 Semantics

This element shall meet the requirements specified in clause 5.3.4 ("Scheme operator name") of ETSI TS 119 612 [1].

## 4.9.2 Syntax

The `Name` element shall be as defined in XML Schema file "XMLSCHEMANAMEFILE.xsd", whose location is detailed in clause B.1, and is copied below for information.

```
<xs:element name="Name" type="tsl:InternationalNamesType" />
```

## 4.10 The `TradeName` element

### 4.10.1 Semantics

This element shall meet the requirements specified in clause 5.4.2 ("TSP trade name") of ETSI TS 119 612 [1] for the contents of TSP trade names for legal persons.

### 4.10.2 Syntax

The `TradeName` element shall be as defined in XML Schema file "XMLSCHEMANAMEFILE.xsd", whose location is detailed in clause B.1, and is copied below for information.

```
<xs:element name="TradeName" type="tsl:InternationalNamesType"/>
```

## 4.11 The `PostalAddresses` element

### 4.11.1 Semantics

This element shall meet the requirements specified in clause 5.3.5.1 ("Scheme operator postal address") of ETSI TS 119 612 [1].

### 4.11.2 Syntax

The `PostalAddresses` element shall be as defined in XML Schema file "XMLSCHEMANAMEFILE.xsd", whose location is detailed in clause B.1, and is copied below for information

```
<xs:element name="PostalAddresses" type="tsl:PostalAddressListType"/>
```

## 4.12 The `ElectronicAddresses` element

### 4.12.1 Semantics

This element shall meet the requirements specified in clause 5.3.5.2 ("Scheme operator electronic address") of ETSI TS 119 612 [1].

### 4.12.2 Syntax

The `ElectronicAddresses` element shall be as defined in XML Schema file "XMLSCHEMANAMEFILE.xsd", whose location is detailed in clause B.1, and is copied below for information.

```
<xs:element name="ElectronicAddresses" type="tsl:ElectronicAddressType"/>
```

## 4.13 The `ContactPersons` element

### 4.13.1 Semantics

This element shall include the details of one or more contact persons.

For each contact person this element shall contain

- 1) one name,
- 2) one or more electronic addresses where to reach that contact person, and
- 3) one or more telephone numbers.

### 4.13.2 Syntax

The `ContactPersons` element shall be as defined in XML Schema file "XMLSCHEMANAMEFILE.xsd", whose location is detailed in clause B.1, and is copied below for information.

```
<xs:element name="ContactPersons" type="ContactPersonsListType"/>
<xs:complexType name="ContactPersonsListType">
  <xs:sequence maxOccurs="unbounded">
    <xs:element ref="ContactPerson"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="ContactPerson" type="ContactPersonType"/>
<xs:complexType name="ContactPersonType">
  <xs:sequence>
    <xs:element name="Name" type="xs:string"/>
    <xs:element ref="ElectronicAddresses"/>
    <xs:element name="PhoneNumbers" type="StringListType"/>
  </xs:sequence>
</xs:complexType>
<xs:simpleType name="StringListType">
  <xs:list itemType="xs:string"/>
</xs:simpleType>
```

Each string value in the `PhoneNumbers` element shall contain exactly one phone number of the contact person.

## 4.14 The `OtherDetails` element

### 4.14.1 Semantics

This element shall contain:

- 1) The date and time of issue of the signature policy.

This element may also contain:

- 1) The signing period, which identifies the date and time before which, and an optional date and time after which,
  - a. The signature policy should not be used for creating new signatures.
  - b. A signature should not be considered valid if validating a signature against the signature policy

**NOTE:** While this component impacts the creation and validation of a digital signature, it does not impact the augmentation of a digital signature, as this augmentation aims at preserving the status of the signature regardless when the signature was created.

- 2) A list of other general details, not defined in the present document

### 4.14.2 Syntax

The `OtherDetails` element shall be as defined in XML Schema file "XMLSCHEMANAMEFILE.xsd", whose location is detailed in clause B.1, and is copied below for information.

```
<xs:element name="OtherDetails" type="OtherDetailsType"/>
<xs:complexType name="OtherDetailsType">
  <xs:sequence>
    <xs:element name="DateOfIssue" type="xs:dateTime"/>
    <xs:element name="SigningPeriod" type="TimePeriodType" minOccurs="0"/>
    <xs:element name="Others" type="AnyType" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
```

```

<xs:complexType name="TimePeriodType">
  <xs:sequence>
    <xs:element name="NotBefore" type="xs:dateTime"/>
    <xs:element name="NotAfter" type="xs:dateTime" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="AnyType" mixed="true">
  <xs:sequence minOccurs="0" maxOccurs="unbounded">
    <xs:any namespace="##any" processContents="lax"/>
  </xs:sequence>
  <xs:anyAttribute namespace="##any"/>
</xs:complexType>

```

## 4.15 The OtherPolicies element

### 4.15.1 Semantics

This element shall contain a sequence of elements, each one providing details of a externally defined signature policy.

The details provided for each externally defined signature policy shall be as specified in clause 4.6.

If a referred signature policy contradicts the present signature policy, the rules in present document take precedence.

If the present document refers to other policies which have contradicting rules, then the rules appearing in the first of the aforementioned policies in the list take precedence.

### 4.15.2 Syntax

The OtherPolicies element shall be as defined in XML Schema file "XMLSCHEMANAMEFILE.xsd", whose location is detailed in clause B.1, and is copied below for information.

```

<xs:element name="OtherPolicies" type="SigPoliciesIdsListType"/>
<xs:complexType name="SigPoliciesIdsListType">
  <xs:sequence>
    <xs:element name="PolicyId" type="SigPolicyDetailsType" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>

```

## 4.16 The PolicyRules element

### 4.16.1 Semantics

This element shall contain all the rules defined within the signature policy subject of the machine processable document.

This component shall not be empty. All the components of this element shall be optional.

This element shall contain at least one of the following elements:

- 1) One element containing rules on commitments, corresponding to technical counterpart of BSP (g) in ETSI TS 119 172-1 [i.3]. This element shall be as specified in clause 4.17.
- 2) One element containing rules that apply to the data to be signed, corresponding to technical counterpart of BSP (b) in ETSI TS 119 172-1 [i.3]. This element shall be as specified in clause 4.18.
- 3) One element containing rules on the relationship between the signature and the data to be signed, corresponding to technical counterpart of BSP (c) in ETSI TS 119 172-1 [i.3]. This element shall be as specified in clause 4.19.
- 4) One element containing rules about the responsibility of validating and/or augmenting the signature, corresponding to technical counterpart of BSP (e) in ETSI TS 119 172-1 [i.3]. This element shall be as specified in clause 4.23.



- 5) One element containing rules on the level of assurance required for timing evidence, corresponding to technical counterpart of BSP (h) in ETSI TS 119 172-1 [i.3]. This element shall be as specified in clause 4.27.
- 6) An element containing rules that apply to the identity and the role of the signer, corresponding to technical counterpart of BSP (l) in ETSI TS 119 172-1 [i.3]. This element shall be as specified in clause 4.26.
- 7) An element containing rules that apply to the required level of assurance on signer authentication, including trust conditions on X509 certificates, and trust conditions on revocation status information data objects, corresponding to technical counterpart of BSP (m) in ETSI TS 119 172-1 [i.3]. This element shall be as specified in clause 4.28.
- 8) An element containing rules on mandated signed and unsigned qualifying properties/attributes incorporated to the signature, corresponding to technical counterpart of BSP (o) in ETSI TS 119 172-1 [i.3]. This element shall be as specified in clause 4.29.
- 9) An element containing rules on the required level of assurance of the signature creation device, corresponding to technical counterpart of BSP (n) in ETSI TS 119 172-1 [i.3]. This element shall be as specified in clause 4.30.
- 10) An element containing rules on the cryptographic suites, corresponding to technical counterpart of BSP (p) in ETSI TS 119 172-1 [i.3]. This element shall be as specified in clause 4.31.
- 11) An element acting as placeholder for rules that have not been defined in ETSI TS 119 172-1 [i.3]. This element shall be as specified in clause 4.32.

NOTE: The set of rules defined in this document builds a framework for defining policies, as users can select those rules that better fit the purposes of the signature policies that they are defining.

Each element in the bulleted list above may contain a component indicating a recommended scope for the rules contained in that element (recommended scope component hereinafter). The scope may be any combination of: generation, validation, and augmentation of digital signatures. Users of the signature policy should use the policy rule in the scopes identified in this component.

Absence of the recommended scope component shall mean that users of the signature policy can use the policy rules in any scope.

## 4.16.2 Syntax

The PolicyRules element shall be as defined in XML Schema file "XMLSCHEMANAMEFILE.xsd", whose location is detailed in clause B.1, and is copied below for information.

```
<xs:element name="PolicyRules" type="PolicyRulesListType"/>
<xs:complexType name="PolicyRulesListType">
  <xs:sequence>
    <xs:element ref="PolicyRule" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>

<xs:element name="PolicyRule" type="PolicyRuleType"/>

<xs:group name="CommitmentRulesGroup">
  <xs:choice>
    <xs:element ref="DataToBeSignedRules"/>
    <xs:element ref="SigToDTBSRelationRules"/>
    <xs:element ref="SigFormatAndLevel"/>
    <xs:element ref="AugmentationRules"/>
    <xs:element ref="SigningCertTrustConditions"/>
    <xs:element ref="TimeEvidencesRules"/>
    <xs:element ref="RoleTrustCondition"/>
    <xs:element ref="QualifyingPropertiesRules"/>
    <xs:element ref="SCDLoARules"/>
    <xs:element ref="CryptoSuitesRules"/>
    <xs:element ref="OtherRules"/>
  </xs:choice>
</xs:group>

<xs:complexType name="PolicyRuleType">
  <xs:choice>
```

```

        <xs:element ref="CommitmentRules"/>
        <xs:group ref="CommitmentRulesGroup"/>
    </xs:choice>
    <xs:attribute name="RecommendedScope" type="ScopeListType" use="optional"/>
</xs:complexType>

<xs:simpleType name="ScopeListType">
    <xs:list itemType="ScopeType"/>
</xs:simpleType>

<xs:simpleType name="ScopeType">
    <xs:restriction base="xs:string">
        <xs:enumeration value="Generation"/>
        <xs:enumeration value="Validation"/>
        <xs:enumeration value="Augmentation"/>
    </xs:restriction>
</xs:simpleType>

```

NOTE 1: A `PolicyRule` element can have as child either one of the elements referenced in `CommitmentRulesGroup` or the `CommitmentRules` element.

NOTE 2: The group of elements `CommitmentRulesGroup` groups all elements that can appear as children of the `CommitmentRules` element. It will also be referenced in the definition of the `CommitmentRulesType` in clause 4.17.2.

## 4.17 The `CommitmentRules` element

### 4.17.1 Semantics

This element shall contain a sequence of components. Each one shall contain:

- 1) A non-empty list of commitments identifiers.
- 2) On or more instance of the rules contained in the `PolicyRules` element as specified in clause 4.16, with the only exception that it shall not include the `CommitmentRules` child element.
- 3) A matching value indicator ("all", "atLeastOne", or "none") that identifies to which commitments the rules in the previous bullet apply.

### 4.17.2 Syntax

The `CommitmentRules` element shall be as defined in XML Schema file "XMLSCHEMANAMEFILE.xsd", whose location is detailed in clause B.1, and is copied below for information.

```

<xs:element name="CommitmentRules" type="CommitmentRulesType"/>

<xs:complexType name="CommitmentRulesType">
    <xs:sequence>
        <xs:element ref="CommitmentRulesComponent" maxOccurs="unbounded"/>
    </xs:sequence>
</xs:complexType>

<xs:element name="CommitmentRulesComponent" type="CommitmentRulesComponentType"/>

<xs:complexType name="CommitmentRulesComponentType">
    <xs:sequence>
        <xs:element name="CommitmentIdentifier" type="xs:anyURI" maxOccurs="unbounded"/>
        <xs:sequence maxOccurs="unbounded">
            <xs:group ref="CommitmentRulesGroup"/>
        </xs:sequence>
    </xs:sequence>
    <xs:attribute name="MatchingIndicator" type="MatchingIndicatorType" use="required"/>
</xs:complexType>

<xs:simpleType name="MatchingIndicatorType">

```

```

<xs:restriction base="xs:string">
  <xs:enumeration value="All"/>
  <xs:enumeration value="None"/>
  <xs:enumeration value="AtLeastOne"/>
</xs:restriction>
</xs:simpleType>

```

## 4.18 The DataToBeSignedRules element

### 4.18.1 Semantics

This element shall contain a sequence of 1 or 2 components. Each component:

- 1) Shall have as value a list of mime types, identifying one or more formats.
- 2) Shall have a qualifying component whose value is either "AnyOf" or "NoneOf". Value "AnyOf" indicates that the format(s) of the signed data object(s) shall be present in the aforementioned sequence. Value "NoneOf" indicates that the format(s) of the signed data object(s) shall be absent in the aforementioned sequence.

If this element contains a sequence of 2 components, one of them shall have the qualifying sub-component to "AnyOf" and the other shall have the qualifying sub-component to "NoneOf".

### 4.18.2 Syntax

The DataToBeSignedRules element shall be as defined in XML Schema file "XMLSCHEMANAMEFILE.xsd", whose location is detailed in clause B.1, and is copied below for information.

```

<xs:element name="DataToBeSignedRules" type="DataToBeSignedRulesType"/>
<xs:complexType name="DataToBeSignedRulesType">
  <xs:sequence minOccurs="1" maxOccurs="2">
    <xs:element name="MimeTypes" type="StringListType" maxOccurs="unbounded"/>
    <xs:element name="Qualifier">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:enumeration value="AnyOf"/>
          <xs:enumeration value="NoneOf"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
  </xs:sequence>
</xs:complexType>

```

## 4.19 The SigToDTBSRelationRules element

### 4.19.1 Semantics

This element shall contain:

- 1) An element providing information on the allowed number ranges for the data objects to be signed, as specified in clause 4.20.

This element may also contain:

- 2) A component indicating the relative position of the signature and the signed data object(s), as specified in clause 4.21.
- 3)
- 4) A component indicating the format and the level of the signature as specified in clause 4.22.

## 4.19.2 Syntax

The `SigToDTBSRelationRules` element shall be as defined in XML Schema file "XMLSCHEMANAMEFILE.xsd", whose location is detailed in clause B.1, and is copied below for information.

```
<xs:element name="SigToDTBSRelationRules" type="SigToDTBSRelationRulesType"/>

<xs:complexType name="SigToDTBSRelationRulesType">
  <xs:sequence>
    <xs:element ref="DTBSCardinality"/>
    <xs:element ref="SigDTBSRelativePosition" minOccurs="0"/>
    <xs:element ref="SigFormatsAndLevels" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
```

## 4.20 The DTBSCardinality element

### 4.20.1 Semantics

This element shall contain at least one of the two following components:

- 1) One component whose value is an integer value, indicating a maximum value of data objects to be signed. This component shall also include a qualifier that shall take one of the following values: "LessThan", "LessOrEqualTo", or "Equal". If the qualifier has the value "LessThan", the number of data objects signed by the signature shall be less than the integer value in the component. If the qualifier has the value "LessOrEqualTo", the number of data objects signed by the signature shall be less than or equal to the integer value in the component. If the qualifier has the value "Equal", the number of data objects signed by the signature shall equal to the integer value in the component.
- 1) One component whose value is an integer value, indicating a minimum value of data objects to be signed. This component shall also include a qualifier that shall take one of the following values: "HigherThan", "HigherOrEqualTo", or "Equal". If the qualifier has the value "HigherThan", the number of data objects signed by the signature shall be higher than the integer value in the component. If the qualifier has the value "HigherOrEqualTo", the number of data objects signed by the signature shall be higher than or equal to the integer value in the component. If the qualifier has the value "Equal", the number of data objects signed by the signature shall equal to the integer value in the component.

### 4.20.2 Syntax

The `DTBSCardinality` element shall be as defined in XML Schema file "XMLSCHEMANAMEFILE.xsd", whose location is detailed in clause B.1, and is copied below for information.

```
<xs:element name="DTBSCardinality" type="CardinalityType"/>

<xs:complexType name="CardinalityType">
  <xs:sequence>
    <xs:element name="MaximumValue" minOccurs="0">
      <xs:complexType>
        <xs:simpleContent>
          <xs:extension base="xs:int">
            <xs:attribute name="qualifier" type="MaxValueQualEnumeratedType"
use="required"/>
          </xs:extension>
        </xs:simpleContent>
      </xs:complexType>
    </xs:element>
    <xs:element name="MinimumValue" minOccurs="0">
      <xs:complexType>
        <xs:simpleContent>
          <xs:extension base="xs:int">
            <xs:attribute name="qualifier" type="MinValueQualEnumeratedType"
use="required"/>
          </xs:extension>
        </xs:simpleContent>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
```

```

</xs:complexType>

<xs:simpleType name="MaxValueQualEnumeratedType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="LessThan"/>
    <xs:enumeration value="LessOrEqualTo"/>
    <xs:enumeration value="Equal"/>
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="MinValueQualEnumeratedType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="HigherThan"/>
    <xs:enumeration value="HigherOrEqualTo"/>
    <xs:enumeration value="Equal"/>
  </xs:restriction>
</xs:simpleType>

```

## 4.21 The SigDTBSRelativePosition element

### 4.21.1 Semantics

This component shall be able to indicate either:

- a) that the signature shall be enclosed within an ASiC container (which also encloses the signed data objects), or
- b) that the signature shall be enveloping, enveloped, or detached from the signed data objects, or
- c) that the relative position between the signature and the signed data objects shall be a combination of two or three of enveloping, enveloped, and detached.
- d) That the relative position between the signature and the signed data objects may be any combination of the aforementioned ones.
- e) That the relative position between the signature and the signed data objects shall be all of them.

NOTE: This last situation can only happen in XML and XAdES digital signatures (an XML or a XAdES digital signature can be enveloped in one of the document it signs, enveloping a second signed data object, and detached from a third signed data object).

Absence of this component shall mean that any relative position is valid for the signature policy.

### 4.21.2 Syntax

The DTBSCardinality element shall be as defined in XML Schema file "XMLSCHEMANAMEFILE.xsd", whose location is detailed in clause B.1, and is copied below for information.

```

<xs:element name="SigDTBSRelativePosition" type="SigDTBSRelativePositionType"/>

<xs:simpleType name="SigDTBSRelativePositionType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="EnvelopingSig"/>
    <xs:enumeration value="EnvelopedSig"/>
    <xs:enumeration value="DetachedSig"/>
    <xs:enumeration value="EnvelopingAndEnvelopedSig"/>
    <xs:enumeration value="EnvelopingAndDetachedSig"/>
    <xs:enumeration value="EnvelopedAndDetachedSig"/>
    <xs:enumeration value="ASiC"/>
    <xs:enumeration value="All"/>
  </xs:restriction>
</xs:simpleType>

```

## 4.22 The SigFormatsAndLevels element

### 4.22.1 Semantics

This component shall include:

- 1) One component for identifying the admitted signature formats, or for indicating that any format is admitted.
- 2) One component for indicating the admitted levels of the signature, or for indicating that any level is admitted. This component may also indicate the versions of the specifications where the levels are defined, that are admitted. Absence of this component shall mean that the levels of any version are admitted.

The component indicating the formats shall indicate each format (or any format) using one of the values listed in annex A.1 URIs to signature formats mapping of ETSI TS 119 442: "Electronic Signatures and Infrastructures (ESI); Protocol profiles for trust service providers providing AdES digital signature validation services" [5].

### 4.22.2 Syntax

The SigFormatsAndLevels element shall be as defined in XML Schema file "XMLSCHEMANAMEFILE.xsd", whose location is detailed in clause B.1, and is copied below for information.

```
<xs:element name="SigFormatsAndLevels" type="SigFormatsAndLevelsType"/>
<xs:complexType name="SigFormatsAndLevelsType">
  <xs:sequence>
    <xs:element ref="SigFormats"/>
    <xs:element ref="SigLevels" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="SigFormats" type="SigFormatsType"/>
<xs:complexType name="SigFormatsType">
  <xs:sequence>
    <xs:element name="Format" maxOccurs="unbounded" type="xs:anyURI" />
  </xs:sequence>
</xs:complexType>
<xs:element name="SigLevels" type="SigLevelsType"/>
<xs:complexType name="SigLevelsType">
  <xs:sequence>
    <xs:element name="SigLevel" type="xs:anyURI" maxOccurs="unbounded"/>
    <xs:element name="Version" type="xs:string" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
```

NOTE 1: URIs starting with "<http://uri.etsi.org/ades/191x2/level/>" identify levels defined in ETSI ENs 319 1x2, with x=2, 3, and 4, specifying CAAdES, XAdES, and PAdES digital signatures respectively.

NOTE 2: URIs starting with "<http://uri.etsi.org/ades/etsits/level/>" identify levels defined in the different ETSI TSs specifying CAAdES, XAdES, and PAdES signatures.

## 4.23 The AugmentationRules element

### 4.23.1 Semantics

This element shall contain:

- 1) A component indicating whether validation of the signature is required before proceeding to augment it
- 2) A component identifying the constraints for the augmentation of the signature. This component shall include:
  - a. The identifier of one level or an identifier indicating that there are no constraints on the augmentation levels.

- b. A qualifier indicating whether the indicated level is the exact level, a minimum level, or the maximum level the signature has to be augmented to.

## 4.23.2 Syntax

The `AugmentationRules` element shall be as defined in XML Schema file "XMLSCHEMANAMEFILE.xsd", whose location is detailed in clause B.1, and is copied below for information.

```
<xs:element name="AugmentationRules" type="AugmentationRulesType"/>
<xs:complexType name="AugmentationRulesType">
  <xs:sequence>
    <xs:element name="PreviousValidationRequired" type="xs:boolean"/>
    <xs:element name="LevelId">
      <xs:complexType>
        <xs:simpleContent>
          <xs:extension base="xs:anyURI">
            <xs:attribute name="Qualifier">
              <xs:simpleType>
                <xs:restriction base="xs:string">
                  <xs:enumeration value="ThisLevel"/>
                  <xs:enumeration value="MinimumLevel"/>
                  <xs:enumeration value="MaximumLevel"/>
                </xs:restriction>
              </xs:simpleType>
            </xs:attribute>
          </xs:extension>
        </xs:simpleContent>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
</xs:complexType>
```

## 4.24 Types for defining constraints on certificates' trust

### 4.24.1 Introduction

The present clause defines four types:

- 1) `TrustAnchorsListType` which defines the trust anchors.
- 2) `NameConstraintsType` which defines constraints on the names of entities.
- 3) `PolicyConstraintsType` for defining constraints on certificate policies.
- 4) `CertificateTrustTreesType`, which defines constraints on the trust conditions required to certificates.

### 4.24.2 `TrustAnchorsListType` type

#### 4.24.2.1 Semantics

Instances of this type shall contain a list of trust anchors. Each trust anchor may be provided in the different ways indicated below:

- 1) As a tuple formed by a self-signed X.509 certificate and an optional time instant indicating until when the trust anchor is reliable, its absence meaning that the trust anchor is reliable.
- 2) As a tuple formed by a reference to a Trusted List as specified in ETSI TS 119 612: "Electronic Signatures and Infrastructures (ESI); Trusted Lists" [i.5] and an optional time instant indicating until when the trust anchor is reliable, its absence meaning that the usage of this trust anchor is reliable as long as the Trusted List itself is reliable.
- 3) As a reference to a Trust Service Status List, as specified in ETSI TS 102 231: "Electronic Signatures and Infrastructures (ESI); Provision of harmonized Trust-service status information" [i.4] and an optional time

instant indicating until when the trust anchor is reliable, its absence meaning that the usage of this trust anchor is reliable as long as the Trust Status List itself is reliable

## 4.24.2.2 Syntax

The `TrustAnchors` element and the `TrustAnchorsType` type shall be as defined in XML Schema file "XMLSCHEMANAMEFILE.xsd", whose location is detailed in clause B.1, and is copied below for information.

```
<xs:element name="TrustAnchors" type="TrustAnchorListType"/>

<xs:complexType name="TrustAnchorListType">
  <xs:sequence maxOccurs="unbounded">
    <xs:choice>
      <xs:element name="X509CertificateBased" type="Base64BinaryAndTimeType"/>
      <xs:element name="TrustedList" type="URIAndReliableTimeType"/>
      <xs:element name="TrustStatusList" type="URIAndReliableTimeType"/>
    </xs:choice>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="Base64BinaryAndTimeType">
  <xs:simpleContent>
    <xs:extension base="xs:base64Binary">
      <xs:attribute name="reliableUntil" type="xs:dateTime" use="optional"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>

<xs:complexType name="URIAndReliableTimeType">
  <xs:simpleContent>
    <xs:extension base="xs:anyURI">
      <xs:attribute name="reliableUntil" type="xs:dateTime" use="optional"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>
```

## 4.24.3 NameConstraintsType type

### 4.24.3.1 Semantics

Instances of this type shall specify constraints on the names within the certificates (constraints on the subtrees).

Instances of this type shall define a name space within which all subject names in subsequent certificates in a certification path shall be located. The restrictions may apply either to the subject distinguished name or subject alternative names. The restrictions shall apply only when the specified name form is present. If no name of the type is present in the certificate, the certificate shall be acceptable.

Instances of this type shall have two sub-components, namely:

- 1) One subcomponent containing the list of permitted subtrees.
- 2) One subcomponent containing the list of excluded subtrees. Any name matching a restriction in this sub-component shall be considered invalid regardless of information appearing in the subcomponent that contains the list of permitted subtrees.

One names subtree shall be defined by:

- One instance of the type `GeneralName` type defined in IETF RFC 5280: "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile" [3] (the base of the subtree).
- One integer that shall indicate the minimum distance of the names in the subtree to the name acting as base. The default value of this integer shall be 0.
- One optional integer that shall indicate the maximum distance of the names in the subtree to the name acting as base.



### 4.24.3.2 Syntax

The `NameConstraintsType` type shall be as defined in XML Schema file "XMLSCHEMANAMEFILE.xsd", whose location is detailed in clause B.1, and is copied below for information.

```
<xs:complexType name="NameConstraintsType">
  <xs:sequence>
    <xs:element ref="PermittedSubtrees" minOccurs="0"/>
    <xs:element ref="ExcludedSubtrees" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>

<xs:element name="PermittedSubtrees" type="GeneralSubTreesListType" />
<xs:element name="ExcludedSubtrees" type="GeneralSubTreesListType" />

<xs:complexType name="GeneralSubTreesListType">
  <xs:sequence maxOccurs="unbounded">
    <xs:element ref="GeneralSubTree" />
  </xs:sequence>
</xs:complexType>

<xs:element name="GeneralSubTree" type="GeneralSubTreeType"/>

</xsd:complexType>
<xsd:complexType name="GeneralSubTreeType">
  <xsd:sequence>
    <xsd:element name="Base" type="xsd:base64Binary"/>
    <xsd:element name="Minimum" type="xsd:integer" default="0"/>
    <xsd:element name="Maximum" type="xsd:integer" minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>
```

The `Base` child element of `GeneralSubTreeType` shall contain the the base-64 encoding of one DER-encoded instance of type `GeneralName` type defined in IETF RFC 5280 [3].

## 4.24.4 PolicyConstraintsType type

### 4.24.4.1 Semantics

Instances of this type shall constrain path processing in two ways. They can be used to prohibit policy mapping, or to require that each certificate in a path contain an acceptable policy identifier.

Instances of this type shall contain at least one of the following two components:

- 1) One component for restricting policy mapping. It shall have an integer value, which shall indicate the number of additional certificates that may appear in the path (including the trustpoint's self certificate) before policy mapping is no longer permitted.

**EXAMPLE:** A value of one indicates that policy mapping may be processed in certificates issued by the subject of this certificate, but not in additional certificates in the path.

- 2) One component for requiring the presence of an acceptable policy identifier within a set of certificates. It shall have an integer value, which shall indicate the number of additional certificates that may appear in the path (including the trustpoint's self certificate) before an explicit policy is required.

An acceptable policy identifier shall be either the identifier of a policy required by the user of the certification path or the identifier of a policy that has been declared equivalent through policy mapping

### 4.24.4.2 Syntax

The `PolicyConstraintsType` type shall be as defined in XML Schema file "XMLSCHEMANAMEFILE.xsd", whose location is detailed in clause B.1, and is copied below for information.

```
<xs:complexType name="PolicyConstraintsType">
  <xs:sequence>
    <xs:element name="RequireExplicitPolicy" type="xs:integer" minOccurs="0"/>
    <xs:element name="InhibitExplicitPolicy" type="xs:integer" minOccurs="0"/>
  </xs:sequence>
```

```
</xs:complexType>
```

## 4.24.5 CertificateTrustTreesType type

### 4.24.5.1 Semantics

Instances of this type shall contain an instance of `TrustAnchorsListType` as specified in clause 4.24.2 of the present document.

Instances of this type may also contain the following components:

- 1) A component for specifying constraints on the certification path. Its value shall be an integer value greater than or equal to zero. If greater than zero, it shall indicate the maximum number of CA certificates that may be in a certification path following the trustpoint. A value of zero indicates that only the given trustpoint certificate and an end-entity certificate may be used. Absence of this component indicates that there is no limit to the allowed length of the certification path.
- 2) The initial set of acceptable certificate policies: a sequence of identifiers of certificate policies, each of which are acceptable under the signature policy.
- 3) One instance of `PolicyConstraintsType` type as specified in clause 4.24.4 of the present document.
- 4) One instance of `NameConstraintsType` type as specified in clause 4.24.3 of the present document.
- 5) One component that shall indicate the certification path that shall be used. This may be given explicitly, or implicitly, by means of an indication that the certification path provided in the signature shall be used. If this component is present, and its content contradicts requirements in any of the previous components listed in the bullets above, the requirements in the present component shall have preference.

### 4.24.5.2 Syntax

The `CertificateTrustTreesType` type shall be as defined in XML Schema file "XMLSCHEMANAMEFILE.xsd", whose location is detailed in clause B.1, and is copied below for information.

```
<xs:complexType name="CertificateTrustTreesType">
  <xs:sequence maxOccurs="unbounded">
    <xs:element ref="CertificateTrustPoint"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="CertificateTrustPoint" type="CertificateTrustPointType"/>

<xs:complexType name="CertificateTrustPointType">
  <xs:sequence>
    <xs:element ref="TrustAnchors"/>
    <xs:element name="PathLenConstraint" type="xs:integer" minOccurs="0"/>
    <xs:element name="AcceptablePolicySet" type="AcceptablePoliciesListType" minOccurs="0"/>
    <xs:element name="NameConstraints" type="NameConstraintsType" minOccurs="0"/>
    <xs:element name="PolicyConstraints" type="PolicyConstraintsType" minOccurs="0"/>
    <xs:element ref="UseCertPath" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="AcceptablePoliciesListType">
  <xs:sequence maxOccurs="unbounded">
    <xs:element name="AcceptablePolicy" type="xs:anyURI"/>
  </xs:sequence>
</xs:complexType>

<xs:element name="UseCertPath">
  <xs:complexType>
    <xs:choice>
      <xs:element name="AsInSignature"/>
      <xs:sequence>
        <xs:element name="X509Certificate" type="xs:base64Binary" minOccurs="0"
maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:choice>
  </xs:complexType>
</xs:element>
```

The `TrustPoint` child element of `CertificateTrustPointType` and the `X509Certificate` child element of `UseCertPath` shall contain one DER-encoded X509 certificate conformant to Recommendation ITU-T X.509: "Information technology - Open Systems Interconnection - The Directory: Public-key and attribute certificate frameworks" [4].

## 4.25 Types for defining constraints on certificates' revocation status

### 4.25.1 Introduction

The present clause defines two types:

- 1) `CertificateRevReqType` which defines constraints on the certificate revocation checks procedures.
- 2) `CertificateRevStatusType` which defines constraints on the trust conditions required on the certificates' revocation data.

### 4.25.2 `CertificateRevReqType` type

#### 4.25.2.1 Semantics

Instances of this type shall contain rules that specify requirements for the using certificate revocation status information (for instance CRLs, OCSP responses) to check the validity of a certificate.

The entity that validates the signature may take into account information in the certificate for deciding how best to check the revocation status but if the information in the certificate contradicts the requirements expressed in the present component, the requirements of the present document shall take precedence.

The rules specified in this component may apply to different types of certificates (e.g. the signing certificate, CA certificates, OCSP responders' certificates, CRLs issuers' certificates, Time Stamping Units' certificates, etc).

Instances of this type shall have the following components:

- 1) One component for specifying requirements on the revocation checking procedures for end entity certificates.
- 2) One component for specifying requirements on the revocation checking procedures for CA certificates.

Each component shall indicate whether the revocation status of the certificate shall be verified or not, and, in case yes, whether the check shall take place using OCSP, CRL, both of them, either of them, or using other means.

Revocation requirements shall be specified in terms of:

- **clrcheck:** Checks shall be made against current CRLs (or authority revocation lists);
- **ocspcheck:** The revocation status shall be checked using the Online Certificate Status Protocol (RFC 2450 [17]);
- **bothcheck:** Both OCSP and CRL checks shall be carried out;
- **eithercheck:** Either OCSP or CRL checks shall be carried out;
- **nocheck:** No check is mandated.
- **Other:** Other mechanism as defined by signature policy extension

### 4.25.2.2 Syntax

The `CertificateRevReqType` type shall be as defined in XML Schema file "XMLSCHEMANAMEFILE.xsd", whose location is detailed in clause B.1, and is copied below for information.

```
<xs:element name="CertificateRevReq" type="CertificateRevReqType"/>

<xs:complexType name="CertificateRevReqType">
  <xs:sequence>
    <xs:element name="EndRevReq" type="RevocationReqType"/>
    <xs:element name="CACerts" type="RevocationReqType"/>
  </xs:sequence>
</xs:complexType>

<xs:simpleType name="RevocationReqType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="clrcheck"/>
    <xs:enumeration value="ocspcheck"/>
    <xs:enumeration value="bothcheck"/>
    <xs:enumeration value="eithercheck"/>
    <xs:enumeration value="nocheck"/>
    <xs:enumeration value="other"/>
  </xs:restriction>
</xs:simpleType>
```

### 4.25.3 CertificateRevTrustType type

#### 4.25.3.1 Semantics

Instances of this type shall contain rules that specify requirements for the using certificate revocation status information (for instance CRLs, OCSP responses) to check the validity of a certificate.

The entity that validates the signature may take into account information in the certificate for deciding how best to check the revocation status but if the information in the certificate contradicts the requirements expressed in the present component, the requirements of the present document shall take precedence.

The rules specified in this component may apply to different types of certificates (e.g. the signing certificate, CA certificates, OCSP responders' certificates, CRLs issuers' certificates, Time Stamping Units' certificates, etc).

Instances of this type shall have an instance of type `CertificateRevReqType` as specified in clause 4.25.2 of the present document.

Instances of this type may also have the following components:

- 1) A component defining a constraint for the freshness of the revocation status data that shall be used in the validation. This component shall either:
  - indicate the maximum accepted difference between the issuance date of the revocation status data of a certificate and the time of validation or
  - require to accept only revocation status data issued a certain time after the signature was created.

For the special case when an instance of this type defines rules for certificate revocation status of the signing certificate used for validating the signature, it may also have the following component:

- 1) A component mandating that the signing certificate has been issued by a certification authority that keeps revocation notices for revoked certificates even after they have expired, for a period exceeding a lower bound indicated in this component.

#### 4.25.3.2 Syntax

The `CertificateRevTrustType` type shall be as defined in XML Schema file "XMLSCHEMANAMEFILE.xsd", whose location is detailed in clause B.1, and is copied below for information.

```

<xs:complexType name="CertificateRevTrustType">
  <xs:sequence>
    <xs:element ref="CertificateRevReq"/>
    <xs:element name="Freshness" minOccurs="0">
      <xs:complexType>
        <xs:choice>
          <xs:element name="MaxDifferenceRevocationAndValidation" type="xs:time"/>
          <xs:element name="TimeAferSignature" type="xs:time"/>
        </xs:choice>
      </xs:complexType>
    </xs:element>
    <xs:element name="SigCertIssuedByCAKeepsExpiredRevokedCertsInfo"/>
  </xs:sequence>
</xs:complexType>

```

## 4.26 The SigningCertTrustCondition element

### 4.26.1 Semantics

This element shall contain:

- 1) a subcomponent instance of `CertificateTrustTreesType` type as specified in 4.24.5 clause of the present document, and
- 2) a subcomponent instance of `CertificateRevReqType` type as specified in clause 4.25.3 of the present document.

### 4.26.2 Syntax

The `SigningCertTrustCondition` element shall be as defined in XML Schema file "XMLSCHEMANAMEFILE.xsd", whose location is detailed in clause B.1, and is copied below for information.

```

<xs:element name="SigningCertTrustCondition" type="SigningCertTrustConditionType"/>
<xs:complexType name="SigningCertTrustConditionType">
  <xs:sequence>
    <xs:element name="SignerTrustTrees" type="CertificateTrustTreesType"/>
    <xs:element name="SignerRevTrust" type="CertificateRevTrustType"/>
  </xs:sequence>
</xs:complexType>

```

## 4.27 The TimeEvidencesRules element

### 4.27.1 Semantics

This element shall include a sequence of tuples. Each tuple:

- 1) Shall have a component that identifies one or more specific time evidences.
- 2) Shall have a component indicating the Level of Assurance on the time evidences identified in the first component of the tuple.
- 3) May also have a component identifying trust conditions for the certificate path processing used for authenticate the time-stamping authority(ies) that generate the time evidences, and constraints on its name. This component may contain:
  - a. One indication of the caution period. This sub-component indicates a delta time with capability for indicating deltas for days, hours, minutes, and seconds.
  - b. One indication of the maximum elapsed time period permitted between the signature generation instant and the signature time-stamp token generation. As above this sub-component indicates a delta time with capability for indicating deltas for days, hours, minutes, and seconds.
  - c. One instance of `CertificateTrustTreesType` type as specified in clause 4.24.5 of the present document.

- d. One instance of `CertificateRevTrustType` type as specified in clause 4.25.3 of the present document.
- e. One instance of `NameConstraintsType` type as specified in clause 4.24.3 of the present document.

## 4.27.2 Syntax

The `TimeEvidencesRules` element shall be as defined in XML Schema file "XMLSCHEMANAMEFILE.xsd", whose location is detailed in clause B.1, and is copied below for information.

```

<xs:element name="TimeEvidencesRules" type="TimeEvidencesRulesType"/>

<xs:complexType name="TimeEvidencesRulesType">
  <xs:sequence>
    <xs:element ref="RulesForSetOfEvidences" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>

<xs:element name="RulesForSetOfEvidences" type="RulesForSetOfEvidencesType"/>

<xs:complexType name="RulesForSetOfEvidencesType">
  <xs:sequence>
    <xs:element name="EvidenceIdentifiers">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="Identifier" type="xs:anyURI" maxOccurs="unbounded"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="LevelOfAssurance" type="xs:anyURI"/>
    <xs:element ref="TimeStampTrustCondition" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>

<xs:element name="TimeStampTrustCondition" type="TimeStampTrustConditionType"/>

<xs:complexType name="TimeStampTrustConditionType">
  <xs:sequence>
    <xs:element name="TstCertificateTrustTrees" type="CertificateTrustTreesType"
minOccurs="0"/>
    <xs:element name="TstRevocationTrust" type="CertificateRevTrustType" minOccurs="0"/>
    <xs:element name="TstNameConstraints" type="NameConstraintsType" minOccurs="0"/>
    <xs:element name="CautionPeriod" type="DeltaTimeType" minOccurs="0"/>
    <xs:element name="SignatureTimeStampDelay" type="DeltaTimeType" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="DeltaTimeType">
  <xs:sequence>
    <xs:element name="DeltaSeconds" type="xs:integer"/>
    <xs:element name="DeltaMinutes" type="xs:integer"/>
    <xs:element name="DeltaHours" type="xs:integer"/>
    <xs:element name="DeltaDays" type="xs:integer"/>
  </xs:sequence>
</xs:complexType>

```

## 4.28 The RolesConstraints element

### 4.28.1 Semantics

This component defines constrains on the signer attributes signed qualifying property, if the signature incorporate such property.

If this component is not present, then any signer attribute enclosed within an attribute certificate or within a signed assertion shall be considered valid under the validation policy.

This component shall be a choice between:

- 1) an indication that signatures according to the policy shall not include signer attributes and

- 2) a list of components defining constraints for signer attributes.

If the first component is present, signatures conformant to the signature policy shall not incorporate any signer attributes.

Each component in the aforementioned list:

- 1) Shall identify the level of assurance of the signer attribute, namely: claimed, certified, a signed assertion, or any of them.
- 2) May include, in the case the signer attribute is certified or a signed assertion, an instance of `CertificateTrustTreesType` type as specified in clause of the present document.
- 3) May include, in the case the signer attribute is certified or a signed assertion, an instance of `CertificateRevTrustType` type as specified in clause of the present document.
- 4) May include one or more components identifying one signer attribute and providing more specific constraints for it. More specifically each component:
  - Shall include a component identifying the mandated signer attribute ("signer attribute identifier" hereinafter).
  - May include a component defining constraints on the value of the signer attribute. If the "signer attribute identifier" component is not present, then this component shall not be present.

If the list of components defining constraints for the signer attributes is present, then signatures compliant with the signature policy shall incorporate the signer attributes identified in the "signer attribute identifier" components, and every signer attribute in the signature shall meet the requirements specified by the components in the numbered bulleted list above.

If this component does not have any "signer attribute identifier" component, then the requirements specified by the rest of the components in the numbered bulleted list above shall apply to any signer attribute incorporated to the signature.

## 4.28.2 Syntax

The `RoleConstraints` element shall be as defined in XML Schema file "XMLSCHEMANAMEFILE.xsd", whose location is detailed in clause B.1, and is copied below for information.

```

<xs:element name="RolesConstraints" type="RolesConstraintsType"/>

<xs:complexType name="RolesConstraintsType">
  <xs:choice>
    <xs:element name="NoSignerAttributesAllowed"/>
    <xs:element name="ConstraintsOnOneSetOfRoles" type="RolesSetConstraintsType"
maxOccurs="unbounded"/>
  </xs:choice>
</xs:complexType>

<xs:complexType name="RolesSetConstraintsType">
  <xs:sequence>
    <xs:element name="HowCertRole" type="HowCertRoleType"/>
    <xs:element name="AttrCertTrustTrees" type="CertificateTrustTreesType" minOccurs="0"/>
    <xs:element name="RoleRevocationTrust" type="CertificateRevTrustType" minOccurs="0"/>
    <xs:element name="RoleConstraints" type="RoleConstraintsType" minOccurs="0"
maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>

<xs:simpleType name="HowCertRoleType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="ClaimedRole"/>
    <xs:enumeration value="CertifiedRole"/>
    <xs:enumeration value="SignedAssertion"/>
    <xs:enumeration value="Any"/>
  </xs:restriction>
</xs:simpleType>

<xs:complexType name="RoleConstraintsType">
  <xs:sequence >
    <xs:element name="RoleIdMustBePresent" type="xs:anyURI"/>

```

```

      <xs:element name="RoleValueConstraint" type="AnyType" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>

```

If the signer attribute type is identified by an OID, the value of the `RoleIdeMustBePresent` child element shall be the OID value encoded as an URN as specified by the IETF RFC 3061 [2].

## 4.29 The `QualifyingPropertiesRules` element

### 4.29.1 Semantics

This component shall include a sequence of tuple. Each tuple:

- 1) May include an identifier of a signature level, as specified in clause 0 of the present document.
- 2) Shall include one component defining requirements for signed properties.
- 3) Shall include one component defining requirements for unsigned properties.

The components defining requirements for signed and unsigned properties:

- 1) Shall provide means for individually and completely identifying the qualifying property affected by the constraints, using an URI.
- 2) Shall provide means for indicating a choice between several properties.
- 3) Shall provide means for indicating the level of the presence requirement, namely: "mandatory" or "optional". Shall provide means for applying the level of presence requirement to both individual properties and choices among several properties.

Absence of the present component indicates that no requirements are imposed on qualifying properties, and signatures compliant with this signature policy may have signed/unsigned qualifying properties or not, and if they have all of them shall be considered valid against the policy.

The URIs for identifying qualifying properties shall be as follows:

- 1) For CADES attributes, the URI for one attribute shall be the OID value encoded as an URN as specified by the IETF RFC 3061 [2].

EXAMPLE 1: The URI identifying the `commitment-type-indication` attribute is  
`urn:oid:1.2.840.113549.1.9.16.2.16.`

- 2) For XAdES attributes the URI for one qualifying property shall be the URI of the namespace where the property has been defined plus a fragment whose value shall be the local name of the property.

EXAMPLE 2: The URI identifying the `CommitmentTypeIndication` qualifying property is  
<http://uri.etsi.org/01903/v1.3.2#CommitmentTypeIndication>.

- 3) For PAdES signatures, the URI for one of the fields in one of the PDF dictionaries shall be the URI corresponding to the ETSI document where the field is defined, with the name of the dictionary added to the path, and with a fragment whose value is the name of the field in the dictionary.

EXAMPLE 3: The URI identifying the `Certs` field of the DSS PDF dictionary defined in ETSI EN 319 162-1 version v1.1.1 is <http://uri.etsi.org/1916201/v1.1.1/DSS#Certs>.

NOTE: The part of the path identifying the ETSI document is formed from the number of deliverable (319 162 in this case), dropping the first digit and adding the part, subpart, and so on number with two digits.

### 4.29.2 Syntax

The `QualifyingPropertiesRules` element shall be as defined in XML Schema file "XMLSCHEMANAMEFILE.xsd", whose location is detailed in clause B.1, and is copied below for information.



```

<xs:element name="QualifyingPropertiesRules" type="QualifyingPropertiesRulesType"/>

<xs:complexType name="QualifyingPropertiesRulesType">
  <xs:sequence>
    <xs:element name="LevelRules" type="LevelQualifyingPropertiesRulesType"
maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="LevelQualifyingPropertiesRulesType">
  <xs:sequence>
    <xs:element name="LevelIdentifier" type="xs:anyURI" minOccurs="0"/>
    <xs:element name="SignedQualifyingProperties" type="QualifyingPropertiesListType"
minOccurs="0"/>
    <xs:element name="UnsignedQualifyingProperties" type="QualifyingPropertiesListType"
minOccurs="0"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="QualifyingPropertiesListType">
  <xs:sequence maxOccurs="unbounded">
    <xs:choice>
      <xs:element name="Choice">
        <xs:complexType>
          <xs:sequence maxOccurs="unbounded">
            <xs:element name="PropertyId" type="xs:anyURI"/>
          </xs:sequence>
          <xs:attribute name="presenceLevel" type="PresenceLevelType" use="required"/>
        </xs:complexType>
      </xs:element>
      <xs:sequence>
        <xs:element name="PropertyId" maxOccurs="unbounded">
          <xs:complexType>
            <xs:simpleContent>
              <xs:extension base="xs:anyURI">
                <xs:attribute name="presenceLevel" type="PresenceLevelType"
use="required"/>
              </xs:extension>
            </xs:simpleContent>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:choice>
  </xs:sequence>
</xs:complexType>

<xs:simpleType name="PresenceLevelType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="Mandatory"/>
    <xs:enumeration value="Optional"/>
  </xs:restriction>
</xs:simpleType>

```

## 4.30 The SCDDLoARules element

### 4.30.1 Semantics

This component shall indicate the Level of Assurance of the signature creation device where resides the private key corresponding to the public key within the certificates validated during the certificate validation process.

### 4.30.2 Syntax

The SCDDLoARules element shall be as defined in XML Schema file "XMLSCHEMANAMEFILE.xsd", whose location is detailed in clause B.1, and is copied below for information.

```
<xs:element name="SCDDLoARules" type="xs:anyURI"/>
```

The SCDDLoARules shall be an URI value indicating the Level of Assurance of the signature creation device.

## 4.31 The `CryptoSuitesRules` element

### 4.31.1 Semantics

This component shall contain:

- 1) One component specifying constraints for the algorithms and parameters used for generating the digital signature.
- 2) One component specifying constraints for the algorithms and parameters used for generating the signing certificate.
- 3) One component specifying constraints for the algorithms and parameters used for generating the certificates in the certification paths.
- 4) One component specifying constraints for the algorithms and parameters used for generating the certificate revocation status data.
- 5) One component specifying constraints for the algorithms and parameters used for generating the time-stamp tokens, attribute certificates, and signed assertions.

Each of the aforementioned components:

- 1) Shall contain the identifier of an algorithm.
- 2) May contain an indication of the minimum length of a key.
- 3) May contain an indication of the minimum length of a hash value.
- 4) Shall contain an indication of the expiration date.

### 4.31.2 Syntax

The `CryptoSuitesRules` element shall be as defined in XML Schema file "XMLSCHEMANAMEFILE.xsd", whose location is detailed in clause B.1, and is copied below for information.

```
<xs:element name="CryptoSuitesRules" type="CryptoSuiteRulesType"/>
<xs:complexType name="CryptoSuiteRulesType">
  <xs:sequence>
    <xs:element name="SignerAlgConstraints" type="AlgConstraintsListType" minOccurs="0"/>
    <xs:element name="EeCertAlgConstraints" type="AlgConstraintsListType" minOccurs="0"/>
    <xs:element name="CACertAlgConstraints" type="AlgConstraintsListType" minOccurs="0"/>
    <xs:element name="AaCertAlgConstraints" type="AlgConstraintsListType" minOccurs="0"/>
    <xs:element name="TSACertAlgConstraints" type="AlgConstraintsListType" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="AlgConstraintsListType">
  <xs:sequence maxOccurs="unbounded">
    <xs:element name="AlgAndLength" type="AlgAndLengthType"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="AlgAndLengthType">
  <xs:sequence>
    <xs:element name="AlgId" type="xs:anyURI"/>
    <xs:element name="MinKeyLength" type="xs:integer" minOccurs="0"/>
    <xs:element name="MinHashLength" type="xs:integer" minOccurs="0"/>
    <xs:element name="Other" type="AnyListType" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
```

## 4.32 The `OtherRules` element

### 4.32.1 Semantics

This component acts a container for components not specified yet in the present document.

It shall be able to contain one or more components each one being of different types.

## 4.32.2 Syntax

The `OtherRules` element shall be as defined in XML Schema file "XMLSCHEMANAMEFILE.xsd", whose location is detailed in clause B.1, and is copied below for information.

```
<xs:element name="OtherRules" type="AnyListType"/>

<xs:complexType name="AnyListType">
  <xs:sequence>
    <xs:element name="Rule" type="AnyType" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
```

---

## Annex A (normative): URIs for identifying signature formats

### A.1 URIs to signature formats mapping

Table 2 lists the URIs used for identifying the digital signature formats.

**Table 2: Namespaces for identifying digital signature formats**

URI	Format identified
<a href="http://uri.etsi.org/ades/format/cades">http://uri.etsi.org/ades/format/cades</a>	CAdES
<a href="http://uri.etsi.org/ades/format/pades">http://uri.etsi.org/ades/format/pades</a>	PAdES
<a href="http://uri.etsi.org/ades/format/xades">http://uri.etsi.org/ades/format/xades</a>	XAdES
<a href="http://uri.etsi.org/ades/format/asic">http://uri.etsi.org/ades/format/asic</a>	ASiC
<a href="http://uri.etsi.org/ades/format/any">http://uri.etsi.org/ades/format/any</a>	Any format

---

## Annex B (normative): XML Schema file

### B.1 XML Schema file location for namespace <http://uri.etsi.org/01910202/v1.1.1#>

The file at (URI\_FOR\_XML\_SCHEMA\_FILE)\_(XMLSCHEMANAMEFILE.xsd) contains the definitions of qualifying properties defined within the namespace whose URI value is <http://uri.etsi.org/01910202/v1.1.1#>.

*Editor's note: the draft xml schema is temporarily available here  
[https://docbox.etsi.org/ESI/Open/Latest\\_Drafts/ts\\_119172\\_2v000a.xsd](https://docbox.etsi.org/ESI/Open/Latest_Drafts/ts_119172_2v000a.xsd)*

*anyone intending to validate the xml schema file itself will be reported with some errors due to the fact that the `xs:import` for TL and for TR 102 038 do not point to places globally accessible in internet*

---

# History

---

<b>Document history</b>		
V0.0.3	2019-05	Stable draft for public availability