Draft ETSI TS 119 442 V0.0.9d (2018-07)

Electronic Signatures and Infrastructures (ESI);

Protocol profiles for trust service providers providing AdES digital signature validation services

[Based on OASIS DSS-X core v2.0]

<

**TECHNICAL SPECIFICATION**

Reference

DTS/ESI-0019442

Keywords

<keywords>

***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 only prevailing document is the print of the Portable Document Format (PDF) version kept on a specific network drive within ETSI Secretariat.

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.

© European Telecommunications Standards Institute 2017.

All rights reserved.

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

# Contents

Contents 3

Intellectual Property Rights 8

Foreword 8

Modal verbs terminology 8

Executive summary 8

1 Scope 8

2 References 9

2.1 Normative references 9

2.2 Informative references 10

3 Definitions, symbols and abbreviations 10

3.1 Definitions 10

3.3 Abbreviations 11

4 Technical approach to the specification of the profiles. 11

4.1 Introduction 11

4.2 XML protocol 12

4.3 JSON protocol 13

5 Profile validation of AdES signatures 13

5.1 Request message 13

5.1.1 Component for requesting validation 13

5.1.1.1 Component semantics 13

5.1.1.2 XML component 14

5.1.1.3 JSON component 14

5.1.2 Component for submitting signature to be validated 15

5.1.2.1 Component semantics 15

5.1.2.2 XML components 16

5.1.2.3 JSON component 16

5.1.3 Component for submitting signed documents or representations of the signed documents 16

5.1.3.1 Component semantics 16

5.1.3.2 XML components 18

5.1.3.2.1 General requirements 18

5.1.3.2.2 Additional requirements for contents of dss:InputDocuments 18

5.1.3.2.2.1 Element dss2:Document for sending original documents 18

5.1.3.2.2.2 Element dss2:TransformedData for sending transformed documents 18

5.1.3.2.2.3 Element dss2:DocumentHash for sending digest of documents 19

5.1.3.3 JSON component 19

5.1.3.3.1 General requirements 19

5.1.3.3.2 Additional requirements for contents of inDocs 19

5.1.3.3.2.1 Element doc for sending original documents 19

5.1.3.3.2.2 Element transformed for sending transformed documents 19

5.1.3.3.2.3 Element docHash for sending digest of documents 20

5.1.3.4 Cardinalities for elements used for sending information of documents 20

5.1.4 Optional components 20

5.1.4.1 Container for optional components 20

5.1.4.1.1 Semantics 20

5.1.4.1.2 XML component 21

5.1.4.1.3 JSON component 22

5.1.4.2 Component for identifying the signatures to be validated (signatures-to-process-ids container)Components for managing the individual validation of signatures 23

5.1.4.2.1 Introduction 23

5.1.4.2.2 Semantics 23

5.1.4.2.3 XML component 23

5.1.4.2.4 JSON component 24

5.1.4.2.2.4 Processing model 25

5.1.4.3 Component for identifying under which service policy the validation has to be conducted 25

5.1.4.3.1 Component semantics 25

5.1.4.3.2 XML component 25

5.1.4.3.3 JSON component 25

5.1.4.4 Component for allowing the client to claim for an identity 25

5.1.4.4.1 Component semantics 25

5.1.4.4.2 XML component 25

5.1.4.4.3 JSON component 25

5.1.4.5 Component for requesting notifications in a certain language 26

5.1.4.5.1 Component semantics 26

5.1.4.5.2 XML component 26

5.1.4.5.3 JSON component 26

5.1.4.6 Component for requesting to set the validation time to a certain instant different from current time 26

5.1.4.6.1 Component semantics 26

5.1.4.6.2 XML component 26

5.1.4.6.3 JSON component 26

5.1.4.7 Component for requesting to return the validation time 26

5.1.4.7.1 Component semantics 26

5.1.4.7.2 XML component 26

5.1.4.7.3 JSON component 26

5.1.4.8 Component for passing validation material to the server 27

5.1.4.8.1 Component semantics 27

5.1.4.8.2 XML component 27

5.1.4.8.3 JSON component 27

5.1.4.9 Component for requesting the server to return the identity of the signer 27

5.1.4.9.1 Component semantics 27

5.1.4.9.2 XML component 27

5.1.4.9.3 JSON component 27

5.1.4.10 Component for requesting validation against a certain signature policy 27

5.1.4.10.1 Component semantics 27

5.1.4.10.2 XML component 27

5.1.4.10.3 JSON component 28

5.1.4.10.4 Processing model 28

5.1.4.11 Component for requesting a detailed validation report (as the one to be specified in TS 119 102-2) 29

5.1.4.11.1 Component semantics 29

5.1.4.11.2 XML component 29

5.1.4.11.3 JSON component 29

5.1.4.12 Component for requesting that the validation report is signed by the server a detailed validation report (as the one to be specified in TS 119 102-2) 29

5.1.4.12.1 Component semantics 29

5.1.4.12.2 XML component 29

5.1.4.12.3 JSON component 29

5.1.4.13 Component for requesting the server to return the result of transforming the input document 30

5.1.4.13.1 Component semantics 30

5.1.4.13.2 XML component 30

5.1.4.13.3 JSON component 30

5.1.4.14 Component for requesting to return the validation of signed ds:Manifest in XAdES signatures 30

5.1.4.14.1 Component semantics 30

5.1.4.14.2 XML component 30

5.1.4.14.3 JSON component 30

5.1.4.15 Component for identifying a request 30

5.1.4.15.1 Component semantics 30

5.1.4.15.2 XML component 30

5.1.4.15.3 JSON component 31

5.1.5 Asynchronous processing 31

5.1.5.1 Asynchronous processing protocol 31

5.1.5.2 Component for identifying a request as a subsequent request to an initial request 31

5.1.5.2.1 Semantics 31

5.1.5.2.2 XML component 31

5.1.5.2.3 JSON component 31

5.1.5.3 Component for correlating subsequent requests to the initial response 32

5.1.5.3.1 Semantics 32

5.1.5.3.2 XML component 32

5.1.5.3.3 JSON component 32

5.2 Response message 32

5.2.1 Component for responding to a validation request 32

5.2.1.1 Component semantics 32

5.2.1.2 XML component 32

5.2.1.3 JSON component 33

5.2.2 Component for the global validation result 33

5.2.2.1 Component semantics 33

5.2.2.2 Processing model 33

5.2.2.3 XML component 34

5.2.2.4 JSON component 34

5.2.3 Optional components 34

5.2.3.1 Container for optional components 34

5.2.3.1.1 Semantics 34

5.2.3.1.2 XML component 35

5.2.3.1.3 JSON component 36

5.2.3.2 Identifying the profile used 37

5.2.3.2.1 Semantics 37

5.2.3.2.2 XML components 37

5.2.3.2.3 JSON component 37

5.2.3.3 Component for indicating the service policy 37

5.2.3.3.1 Component semantics 37

5.2.3.3.2 XML component 37

5.2.3.3.3 JSON component 38

5.2.3.4 Component for indicating validation time 38

5.2.3.4.1 Component semantics 38

5.2.3.4.2 XML component 38

5.2.3.4.3 JSON component 38

5.2.3.5 Component for returning signer’s identity 38

5.2.3.5.1 Component semantics 38

5.2.3.5.2 XML component 38

5.2.3.5.3 JSON component 38

5.2.3.6 Component for notifying the signature policy applied during the validation 38

5.2.3.6.1 Component semantics 38

5.2.3.6.2 XML component 38

5.2.3.6.3 JSON component 39

5.2.3.7 Component for notifying the signature policies under which the server can conduct validation 39

5.2.3.7.1 Component semantics 39

5.2.3.7.2 XML component 39

5.2.3.7.3 JSON component 39

5.2.3.8 Component for returning the detailed validation report 40

5.2.3.8.1 Component semantics 40

5.2.3.8.1.1 ETSI XML protocol 40

5.2.3.8.1.2 ETSI JSON protocol 40

5.2.3.9 Component for returned the detailed validation report signed 40

5.2.3.9.1 Component semantics 40

5.2.3.9.2 XML component 40

5.2.3.9.3 JSON component 40

5.2.3.10 Component for returning the result of transforming the input document 40

5.2.3.10.1 Component semantics 40

5.2.3.10.2 XML component 40

5.2.3.10.3 JSON component 41

5.2.3.11 Component for returning the result of validating ds:Manifest elemens in XAdES signatures 41

5.2.3.11.1 Component semantics 41

5.2.3.11.2 XML component 41

5.2.3.11.3 JSON component 41

5.2.3.12 Components for asynchronous processing 41

5.2.3.12.1 Introduction 41

5.2.3.12.2 Component for indicating not completion of signature(s) validation 41

5.2.3.12.2.1 Semantics 41

5.2.3.12.2.2 XML component 41

5.2.3.12.2.3 JSON component 41

5.2.3.12.3 Component for correlating subsequent requests to the initial response 41

5.2.3.12.3.1 Semantics 41

5.2.3.12.3.2 XML component 42

5.2.3.12.3.3 JSON component 42

6 Profile for augmentation of AdES signatures. 42

6.1 Request message 42

6.1.1 Component for requesting augmentation of signatures 42

6.1.1.1 Semantics 42

6.1.1.2 XML component 42

6.1.1.3 JSON component 43

6.1.1.43 Processing model 43

6.1.2 Component for identifying the level the signatures are requested to be augmented to 44

6.1.2.1 Component semantics 44

6.1.2.2 XML component 45

6.1.2.3 JSON component 45

6.2 Response message 45

6.2.1 Component for responding to augmentation request 45

6.2.1.1 Component semantics 45

6.2.1.2 XML component 46

6.2.1.3 JSON component 46

6.2.2 Component for the global augmentation result 47

6.2.2.1 Component semantics 47

6.2.2.2 Processing model 47

6.2.2.2 XML component 47

6.2.2.3 JSON component 47

6.2.3 Signature result container 47

6.2.2.1 Semantics 47

6.2.2.1.2 XML component 48

6.2.2.1.3 JSON component 48

7 Profile for requesting validation and augmentation of signatures. 49

7.1 Request message 49

7.1.1 Component for requesting validation and augmentation 49

7.1.1.1 Component semantics 49

7.1.1.2 XML component 49

7.1.1.3 JSON component 49

7.1.2 Components for submitting signatures and signed documents 49

7.1.3 Optional components 49

7.1.3.1 Container for optional components 49

7.1.3.1.1 Semantics 49

7.1.3.1.2 XML component 50

7.1.3.1.2 JSON component 50

7.1.4 Processing model 50

7.2 Response message 50

7.2.1 Component for responding to validation and augmentation request 50

7.2.1.1 Component semantics 50

7.2.1.2 XML component 50

7.2.1.3 JSON component 51

7.2.2 Component for the global validation and augmentation result 51

7.2.2.1 Semantics 51

7.2.2.2 Processing model 51

7.2.2.3 XML component 51

7.2.2.4 JSON component 51

7.2.3 Optional components 51

7.2.3.1 Container for optional components 51

7.2.3.1.1 Semantics 51

7.2.3.1.2 XML component 51

7.2.3.1.3 JSON component 52

7.2.3.2 Signature result container 52

7.2.3.2.1 Semantics 52

7.2.3.2.2 XML component 52

7.2.3.2.3 JSON component 52

Annex A (normative): XML and JSON Schema files 53

A.1 XML Schema file location for namespace http://uri.etsi.org/19442/v1.1.1# 53

A.2 JSON Schema file location for “$schema” "http://etsi.org/119442/v1.1.1/json#" 53

Annex (informative): Change History 54

History 55

# Intellectual Property Rights

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

# Foreword

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

# 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](https://portal.etsi.org/Services/editHelp%21/Howtostart/ETSIDraftingRules.aspx) (Verbal forms for the expression of provisions).

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

# 1 Scope

The present document specifies the semantics of a protocol for requesting to a remote server (and for receiving the corresponding response) the validation of AdES digital signatures compliant with the following ETSI deliverables: ETSI EN 319 122 [2], ETSI EN 319 132 [3], ETSI EN 319 142 [4], ETSI TS 101 733 [5], ETSI TS 102 778 [9], ETSI TS 101 903 [7], ETSI TS 103 171 [8], ETSI TS 103 172 [10], and ETSI TS 103 173 [6].

The present document specifies the semantics of a second protocol for requesting the augmentation of AdES digital signatures compliant with the aforementioned ETSI deliverables.

The present document also specifies the semantics of a third protocol for requesting the validation and augmentation of AdES digital signatures compliant with the aforementioned ETSI deliverables.

Finally, the present document specifies two bindings, each one in a different syntax (XML and JSON), for each of the aforementioned protocols.

As far as it has been possible and suitable, the protocols have re-used constructs of OASIS DSS and DSS-X specifications.

NOTE 1: The protocols specified in this document do not include components for submitting to the server ASiC containers compliant with ETSI EN 319 152, ETSI TS 102 912, and ETSI TS 103 174. They do not include either components for reporting on the validation of signatures included within an ASiC container. However, clients may always extract individual signatures and groups of signed documents from ASiC containers and prepare and submit suitable requests to the server for these individual signatures and groups of signed documents.

NOTE 2: The protocols specified in this document do not include components for submitting to the server time-stamp tokens for their verification. They do not include either components for reporting on the verification of time-stamp tokens. Protocols specified by OASIS DSS and OASIS DSS-X Technical Committees include this type of components.

# 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] OASIS Standard: Digital Signature Service Core Protocols, Elements, and Bindings Version 2.0.

[2] ETSI EN 319 122: "CAdES digital signatures. ETSI EN 319 122".

[3]ETSI EN 319 132: "XAdES digital signatures. ETSI EN 319 132".

[4] ETSI EN 319 142: "PAdES digital signatures. ETSI EN 319 142".

[5] ETSI TS 101 733: "CMS Advanced Electronic Signatures", April 2014.

[6] ETSI TS 103 173: "CAdES Baseline Profile", April 2014.

[7] ETSI TS 101 903: "XML Advanced Electronic Signatures", December 2012.

[8] ETSI TS 103 171: "XAdES Baseline Profile", March 2012".

[9] ETSI TS 102 778: "PDF Advanced Electronic Signatures"

[10] ETSI TS 103 172: "PAdES Baseline Profile. ETSI TS 103 172", April 2014.

[11] ETSI TS 119 102-2: "Procedures for Creation and Validation of AdES Digital Signatures; Part 2: Signature Validation Report".

[12] IETF RFC 5646: "Tags for Identifying Languages".

[13] IETF RFC 7515: "JSON Web Signature (JWS)".

[14] IETF IS 5652: "Cryptographic Message Syntax (CMS)". September 2009.

[15] W3C Recommendation (11 April 2013): "XML Signature Syntax and Processing. Version 1.1".

[16] ISO 32000-1: "Document management — Portable document format — Part 1: PDF 1.7".

[17] ETSI EN 319 102-1: "Procedures for Creation and Validation of AdES Digital Signatures; Part 1: Creation and Validation".

[18] RFC 3061: "A URN Namespace of Object Identifiers". February 2011.

[19] ETSI TS 119 441: "Policy and security requirements for trust service providers providing AdES digital signature validation services ".

[20] ETSI TR 119 001: "Electronic Signatures and Infrastructures (ESI); The framework for standardization of signatures; Definitions and abbreviations".

## 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 TS 119 172-4 "Electronic Signatures and Infrastructures (ESI); Signature applicability rules for European qualified electronic signatures/seals using trusted lists"

 *(note: to be completed)*

# 3 Definitions, symbols and abbreviations

## 3.1 Definitions

For the purposes of the present document, the terms and definitions given in ETSI TR 119 001 [20], ETSI TS 119 441 [19], and the following apply:

**attachment reference container**: sub-component of input documents container for transferring a reference to an underlying protocol attachment where either the signed document or the transformed document is placed.

**augmented signature container**: response protocol container for transferring to the client an augmented non-embedded AdES signature.

NOTE 1: document with signature container and augmented signature container are components of response messages for the augmentation protocol and for the validation and augmentation protocol.

**document container**: sub-component of input documents container for transferring to the server one signed document or a reference to one underlying transport protocol attachment where the signed document is placed.

**document digest container**: sub-component of input documents container for transferring to the server the digest of one signed document.

**document with signature container**: response protocol container for transferring to the client one signed document embedding its signature(s) or a reference to one underlying transport protocol attachment where the signed document embedding its signature(s) is placed.

**embedded AdES signature**: is an AdES signature placed within a document that it signs total or partially.

NOTE 2: A XAdES enveloped signature (a XAdES signature that signs a data object that contains the XAdES signature itself) is an example, but there may be other situations where a non enveloped XAdES signature is an embedded XAdES signature, for example a XAdES signature that is a component of a XML file, signs only one specific part of that XML file, and this signed part does not envelope the signature.

NOTE 3: The rationale for this definition is that the placement of the signature to be validated and the signed documents within the protocol messages depends on whether the signature is embedded or not, as specified in clause 5.1.2.

**input documents container**: request protocol component for transferring to the server either the signed documents themselves, or the transformed documents, or the digest of the signed documents, or references to underlying transport protocol attachments where the signed documents or the transformed documents are placed.

NOTE 4: For more information about transformations of signed documents, see "XML Signature Syntax and Processing. Version 1.1" [15].

**representation of a (signed) document:** either the (signed) document itself, its digest, or the result of applying to the (signed) document a certain set of known transformations.

**signature object container**: request protocol component for transferring to the server either one non-embedded signature, or a reference to an embedded signature.

EXAMPLE: For instance, the client can place a reference to the signature instead the signature itself in this component when the signature is embedded within the signed document. In these situations the client can include the signed document (and its embedded signature) within the input documents container and include a reference to the signature within the signature object container.

NOTE 5: From the definitions above, input documents container can contain signatures as long as they are embedded within documents. And signature object container can contain signed documents as long as they are fully enveloped by the signature. The basic principle for placement of signed documents and signatures is the following: an object that essence is a non-embedded signature (even if it envelops a signed document) is placed in the signature object container; and an object that in essence is a document (even if it embeds a signature) is placed in or is referenced from the input documents object container.

NOTE 6: Input documents container and signature object container components are implemented by specific XML and JSON types and elements in the bindings defined by the present document.

**signature results container**: response protocol component including the optional outputs generated by the server when processing (validating, augmenting, or validating and augmenting) one specific signature .

NOTE 7: The requests of the three protocols specified in the present document can contain more than one signature. This container includes all the optional outputs generated by the server when it process one of these signatures. The response message can, consequently, contain one or more signature results containers.

**signatures-to-process-ids container**: request protocol component that includes identifiers of the signatures whose processing the client requests to the server.

NOTE 8: The word “processing” in the former definition means “validation” in the validation protocol, “augmentation” in the augmentation protocol, and “validation and augmentation” in the validation and augmentation protocol.

NOTE 9: A request message can include more than one signature. This component allows the client to instruct the server to process (validate, augment, or validate and augment) a selected subset of them.

**transformed document container**: sub-component of input documents container for transferring to the server the transformed document or a reference to one underlying transport protocol attachment where the transformed document is placed.

## 3.2 Abbreviations

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

DSS-X Digital Signature Services eXtended.

# 4 Technical approach to the specification of the profiles.

## 4.1 Introduction

The protocols specified in the present document take as starting point, whenever it has been possible, and as much as possible, the OASIS DSS-X Technical Committee’s specification "OASIS Standard: Digital Signature Service Core Protocols, Elements, and Bindings Version 2.0 [1]".

Whenever it has been possible the components specified within [1] have been re-used. When this has not been possible the present document specifies new components semantically and also syntactically in the two formats: XML and JSON.

Services implementing the protocols defined in the present document shall support all the components specified in the present document, regardless their incorporation in the messages is mandatory or optional.

The rest of the document is organized as follows:

1. Sub-clauses 4.2 and 4.3 provide general remarks on the XML and JSON protocols relying on OASIS DSS and DSS-X Technical Committees’ protocols.
2. Clause 5 specifies requirements for all the components for the validation protocol in its two bindings (XML and JSON) relying on OASIS DSS and DSS-X Technical Committees’ protocols.
3. Clause 6 specifies requirements for those specific components for the augmentation protocol in its two bindings (XML and JSON) relying on OASIS DSS and DSS-X Technical Committees’ protocols.
4. Clause 7 specifies requirements for those specific components for the validation and augmentation protocol in its two bindings (XML and JSON) relying on OASIS DSS and DSS-X Technical Committees’ protocols

For each component of the aforementioned protocols, the present document:

1. Defines requirements for the semantics of the component (i.e. its mandatory contents, its optional contents, etc). These requirements are defined in clauses "Component semantics".
2. Defines requirements for the XML component of the XML protocol relying on OASIS DSS and DSS-X Technical Committees’ protocols, which is able to fulfil the semantic requirements already defined. These requirements are defined in clauses named "XML component". In case there is no XML component specified within the OASIS DSS and DSS-X Technical Committees’ protocols able to fulfil the aforementioned semantics, the present document specifies a suitable XML component and its requirements within clauses named “XML component”.
3. Defines requirements for the JSON component of the JSON protocol relying on OASIS DSS and DSS-X Technical Committees’ protocols, which is able to fulfil the semantic requirements already defined. These requirements are defined in clauses named "JSON component". In case there is no JSON component specified within the OASIS DSS and DSS-X Technical Committees’ protocols able to fulfil the aforementioned semantics, the present document specifies a suitable JSON component and its requirements within clauses named “JSON component”

## 4.2 XML protocol

The structures described in this specification are contained in the schema files [DSS\_Core\_XSD], [AdES\_XSD], [ASYN\_XSD], [SIG\_POL\_XSD], and [SIGNED\_VAL\_ RESP\_XSD] and the xml schema file "[XSDFILESIGVALPROT]". The new elements and types defined in that schema are defined within the XML namespace whose URI value is shown below:

[http://uri.etsi.org/19442/v1.1.1#](http://uri.etsi.org/19442/v1.1.1)

Table 1 shows the URI values of other XML namespaces and their corresponding prefixes used in the aforementioned schema file and within the present document.

Table

|  |  |
| --- | --- |
| URI value of the XML Namespace | Prefix |
| [http://uri.etsi.org/19442/v1.1.1#](http://uri.etsi.org/19442/v1.1.1) | etsival |
| http://docs.oasis-open.org/dss/ns/core | dss2 |
| http://docs.oasis-open.org/dss/ns/base | dsb |
| urn:oasis:names:tc:dss:1.0:profiles:verificationreport:schema# | dssvalrep |
| [http://www.w3.org/2000/09/xmldsig#](http://www.w3.org/2000/09/xmldsig) | ds |
| <http://uri.etsi.org/01903/v1.3.2> | xades |
| <http://uri.etsi.org/01903/v1.4.1> | xadesv141 |
| urn:oasis:names:tc:SAML:2.0:assertion | saml2 |
| ??URI defined in ETSI TS 119 102-2?? | etsivr |

The present document will reference components in the aforementioned documents and further profiles some of them.

Unless anything said against it, or in the absence of any further requirement defined in the present document, the requirements defined in the aforementioned documents for the of each element present in this profile, shall apply.

Unless anything said against it, or in the absence of a different processing model defined in the present document, the processing model (including results returned by the server) defined in the aforementioned documents for each element present in this profile, shall apply.

NOTE: This document does not specify, for instance, that the presence of a certain optional component in the request message imply the presence of a certain optional component in the response message: this is a requirement inherited from the aforementioned OASIS specifications.

In case that a requirement defined in the present document contradicts any requirement defined in "OASIS Standard: Digital Signature Service Core Protocols, Elements, and Bindings Version 2.0" [1] the requirement defined in the present document shall take precedence.

In case that a certain aspect of the processing model defined in the present document contradicts any aspect of the processing model defined in the aforementioned specifications, the processing model defined in the present document shall take precedence.

The present document also specifies elements that are not specified in the aforementioned documents. For these elements, the present document also defines the processing model that the server shall implement. This processing model is specified below the indication Processing model within each clause that specifies one of these elements.

## 4.3 JSON protocol

The structures described in this specification are contained in the schema files [DSS\_Core\_JSCHEMA] and "[JSONSCHEMAFILESIGVALPROT]".

Unless anything said against it, or in the absence of any further requirement defined in the present document, the requirements defined in the aforementioned documents for the of each element present in this profile, shall apply.

Unless anything said against it, or in the absence of a different processing model defined in the present document, the processing model (including results returned by the server) defined in the aforementioned documents for each element present in this profile, shall apply.

NOTE: This document does not specify, for instance, that the presence of a certain optional component in the request message implies the presence of a certain optional component in the response message: this is a requirement inherited from the aforementioned OASIS specifications.

In case that a requirement defined in the present document contradicts any requirement defined in "OASIS Standard: Digital Signature Service Core Protocols, Elements, and Bindings Version 2.0" [1] the requirement defined in the present document shall take precedence.

In case that a certain aspect of the processing model defined in the present document contradicts any aspect of the processing model defined in the aforementioned specifications, the processing model defined in the present document shall take precedence.

The present document also specifies elements that are not specified in the aforementioned documents. For these elements, the present document also defines the processing model that the server shall implement. This processing model is specified below the indication Processing model within each clause that specifies one of these elements.

# 5 Protocol for validation of AdES signatures

## 5.1 Request message

### 5.1.1 Component for requesting validation

#### 5.1.1.1 Component semantics

The message for requesting the validation of an AdES signature to a remote server shall contain components for:

1. Submitting either the signature to be validated or a reference to the signature to be validated when this signature is enveloped within a signed document. Clause 5.1.2 specifies semantic requirements for this component.
2. Submitting the signed document(s) or representation(s) of these signed document(s). Clause 5.1.3 specifies semantic requirements for this component.

NOTE 1: When the signature to validate is separated from all or part of the documents it signs, the signature is placed in one component and the signed document(s) is (are) placed in another component.

NOTE 2: When the signature to validate envelops the signed document (this is the case of a XAdES enveloping and non-separated signature or a CAdES attached structure) or it is enveloped within the signed document (this is the case of a PDF document with one or more PAdES signatures, or a XAdES enveloped and non-separated signature), signature and signed documents are placed in one component.

1. Identifying the request as a request that has been built according to the profile specified in the present document. The identifier shall be the following URI [http://uri.etsi.org/19442/v1.1.1/validationprofile#](http://uri.etsi.org/19442/v1.1.1/validationprofile).

The message for requesting the validation of an AdES signature to a remote server may contain other components for requesting to the server additional features. Clause 5.1.4 lists these optional components and contain references to clauses that specify semantic requirements for each component.

#### 5.1.1.2 XML component

The element that shall be the component for requesting the validation of AdES signature(s) shall be the root element of the message VerifyRequest as specified in the present clause.

The VerifyRequest element shall be defined as in XML Schema file "[XSDFILESIGVALPROT]", whose location is detailed in clause A.1, and is copied below for information.

<!—targetNamespace="http://uri.etsi.org/19442/v1.1.1#" 🡪

 <xs:element name="VerifyRequest" type="VerifyRequestType"/>

 <xs:complexType name=="VerifyRequestType">

 <xs:extension base="dss2:RequestBaseType">

 <xs:sequence>

 <xs:element ref="dss2:InputDocuments" minOccurs="0"/>

 <xs:element name="OptionalInputs" type="OptionalInputsVerifyType" />

 <xs:element ref="dss2:SignatureObject" minOccurs="0"/>

 </xs:sequence>

 </xs:extension>

 </xs:complexType>

The contents inherited from dss2:verifyRequest shall be as specified in OASIS Standard: Digital Signature Service Core Protocols, Elements, and Bindings Version 2.0 [1] clause (RE QUESTBASETYPE 3.12).

The VerifyRequest element shall have one dss2:Profile child element, and it shall have the value [http://uri.etsi.org/19442/v1.1.1/validationprofile#](http://uri.etsi.org/19442/v1.1.1/validationprofile), identifying the request as a validation request compliant with the validation profile specified in the present document.

The dss2:SignatureObject child element shall not contain any time-stamp token.

Any optional component specified in clause 5.1.4 shall appear as child of the OptionalInputs child element of VerifyRequest element.

#### 5.1.1.3 JSON component

The element that shall be the component for requesting the validation of AdES signature(s) shall be the root element of the message dss-VerifyRequest as specified in the present clause.

The dss-VerifyRequest element shall be defined as in JSON Schema file "[JSONSCHEMAFILESIGVALPROT], whose location is detailed in clause A.2, and is copied below for information.

"dss-VerifyRequest": {

 "$xsd-type": "VerifyRequest",

 "type": "object",

 "properties": {

 "inDocs": {

 "type": "object",

 "$ref": "<DSSXCORESCHEMAFILELOCATION>#/definitions/dss2-InputDocumentsType"

 },

 "reqID": {

 "type": "string"

 },

 "profile": {

 "type": "array",

 "items": {

 "type": "string"

 }

 },

 "optInp": {

 "type": "object",

 "$ref": "#/definitions/OptionalInputsVerifyType"

 },

 "sigObj": {

 "type": "object",

 "$ref": "<DSSXCORESCHEMAFILELOCATION>#/definitions/dss-SignatureObjectType"

 }

 }

 "required": ["optInp"]

 }

The profile array shall have one item, and it shall have the value [http://uri.etsi.org/19442/v1.1.1/validationprofile#](http://uri.etsi.org/19442/v1.1.1/validationprofile), identifying the request as a validation request compliant with the validation profile specified in the present document.

### 5.1.2 Component for submitting signature to be validated

#### 5.1.2.1 Component semantics

The protocol shall allow including the signature in different containers according to the following rules:

1. If the signature is an embedded AdES signature, the embedding document and the signature shall be placed either
* within one sub-component of the input documents container or
* within an underlying protocol attachment. In this case, a sub-component of the input documents container shall include a reference to the aforementioned attachment.

Additionally the signature object container shall contain a reference to the embedded signature.

1. If the signature is a non-embedded AdES signature, it shall be placed within the signature object container.

Table 2 shows the cardinalities of the components required in this profile for incorporating signature(s), for requesting the validation of AdES signatures, depending on its types (CAdES, PAdES or XAdES) and their relative position with respect the signed document(s).

Rows in the table show information corresponding to the different types of AdES signatures whose validation is requested, as well as their relative position to the signed document(s).

The first column shows the different types of AdES signatures that can be submitted to the server.

Column I shows the cardinalities of the signature object container.

Colum II shows the contents of the signature object container. The values appearing in the cells of this column may be the following ones:

* Signature: This value indicates that the signature object container contains the non-embedded signature itself.
* Message reference. This content appears when submitting an embedded signature. It indicates that the signature object container shall contain a reference to the embedded signature.

Column III shows the cardinalities of the input documents container that includes either a document embedding the signature or a reference to an underlying protocol attachment that contains the document embedding the signature.

Each cell in I and III columns indicates the required cardinality of the component shown in the header of the corresponding column, for the type of signature, located in a relative position to the signed document(s) as indicated in the header of the corresponding row. An integer value indicates an exact number of components, "\*" stands for "0 or more", and "0..1" means "0 or 1".

Table . Placement of signatures.

|  |  |  |  |
| --- | --- | --- | --- |
| ComponentType of signature | ISignature object container | IIContents of signature object container | IIIInput documents container containing embedded signatures or references to attachments with signatures |
| CAdES attached | 1 | Signature  | 0 |
| CAdES detached | 1 | Signature  | 0 |
| Non-embedded XAdES  | 1 | Signature  | 0 |
| Embedded XAdES | 1 | Message reference | 1 |
| PAdES enveloped within the PDF document (embedded signature) | 1 | Message reference | 1 |

#### 5.1.2.2 XML components

The element that shall be the component for submitting the signature(s) to be validated shall be either:

1. the dss2:SignatureObject child element of VerifyRequest root element if the signature is not enveloped within the signed document(s) or
2. the dss2:InputDocument child element of VerifyRequest element if the signature(s) is enveloped within the signed document.

NOTE: dss2:SignatureObject is the XML implementation of the signature object container and that dss2:InputDocument is the XML implementation of the input documents container within the XML binding of the protocol.

The requirements governing the presence, cardinalities, and contents of the aforementioned elements are given in clause 5.1.3.4.

#### 5.1.2.3 JSON component

 The element that shall be the component for submitting the signature(s) to be validated shall be either:

1. the sigObj child element of dss-VerifyRequest root element if the signature is not enveloped within the signed document(s)or
2. the inDocs child element of dss-VerifyRequest element if the signature(s) is enveloped within the signed document.

NOTE: sigObj is the JSON implementation of the signature object container and that inDocs is the JSON implementation of the input documents container within the JSON binding of the protocol.

The requirements governing the presence, cardinalities, and contents of the aforementioned elements are given in clause 5.1.3.4.

### 5.1.3 Component for submitting signed documents or representations of the signed documents

#### 5.1.3.1 Component semantics

The protocol shall allow including the document in several containers depending on its relative position with regards the signature that signs it, and whether it is the actual document what is submitted to the server, or its digest, or a transformed version of the original message:

1. If a non-embedded signature envelops the document, then the client shall place the enveloping signature and the enveloped signed document within the signature object container.
2. If the signature does not accomplish with the requirements in item 1), then the client shall place the signed document either directly within the input documents container or within an underlying protocol attachment. Additionally the input documents container shall contain one sub-component containing either the document or a reference to the attachment where the document is placed.

The current profile shall also allows submitting to the server different representations of each one of the signed documents. Each type of representation shall be placed in a different sub-component of the input documents container, as indicated below.

1. If a client wants to submit the actual signed document it shall place it either:
* within the document container OR
* within an underlying transport protocol attachment. In this case, the document container shall include a reference to this attachment.
1. If a client wants to submit the digest of the actual signed document is sent to the server, then the client shall place this digest within the document digest container.
2. If a client wants to submit the result of transforming a document, being this result what has actually been signed, it shall place this transformed document either:
* within the transformed document container OR
* within an underlying transport protocol attachment. In this case, the transformed document container shall include a reference to this attachment

Table 3 shows the cardinalities of the components required in this profile for incorporating signed documents, or signed documents representations (transformed documents and documents digests) for requesting the validation of AdES signatures, depending on the types of the signatures and their relative position with the signed document(s).

Rows in the table show information corresponding to the different types of AdES signatures whose validation is requested, as well as their relative position to the signed document(s).

The first column shows the different types of AdES signatures that can be submitted to the server.

Column **Document container** shows the cardinalities of the document container.

Column **Transformed document container** shows the cardinalities of transformed document container.

Column **Document digest container** shows the cardinalities of document digest container.

Conventions for indicating cardinalities are as in Table 2. In addition to that, the column “Notes” contains integers identifying additional notes present below the Table 2.

Table . Components containing either documents or representations of documents

in validation requests messages.

|  |  |
| --- | --- |
| Component and subcomponentsType of signature | Input documents container containing documents or documents representations |
| Document container | Transformed document container | Document digest container |
| CAdES attached | 0 | 0 | 0 |
| CAdES detached | 0..1 | 0 | 0..1 |
| Non-embedded XAdES | \* | \* | \* |
| Embedded XAdES | 1..\* | \* | \* |
| PAdES enveloped within the PDF document | 1 | 0 | 0 |

NOTE: In the case of embedded XAdES document container has a minimum cardinality of 1, as it is the only way of submitting to the server the embedded signature: within the signed document that embeds it. If the signature is not embedded, then either the document or a representation of the document can be submitted as the signature is submitted within the signature object container.

#### 5.1.3.2 XML components

##### 5.1.3.2.1 General requirements

The dss2:SignatureObject element shall implement the signature object container for the XML binding of the protocol.

The dss2:InputDocuments element shall implement the input documents container for the XML binding of the protocol.

The requirements governing the presence, cardinalities, and contents of the XML components are given in clause 5.1.3.4.

##### 5.1.3.2.2 Additional requirements for contents of dss:InputDocuments

###### 5.1.3.2.2.1 Element dss2:Document for sending original documents

The dss2:Document element shall implement the document container for the XML binding of the protocol.

NOTE: dss2:Base64Data child element of dss2:Document can contain either the signed document in its dsb:Value child element, or a reference to an underlying protocol attachment (where the signed document is placed) in its dsb:AttRef child element.

###### 5.1.3.2.2.2 Element dss2:TransformedData for sending transformed documents

The dss2:TransformedData element shall implement the transformed document container for the XML binding of the protocol.

If the signature(s) to be validated is(are) XAdES signature(s), and if the client wants to submit to the server not the original document, but the result of the set of transformations applied to it before computing the digest that appears within the ds:DigestManifest child element of the corresponding ds:Reference in the XAdES signature, then the client shall incorporate the base-64 encoding of the binary representation of the result of applying the same sequence of transformations to the original document either into the dss2:Base64Data child element of dss2:TransformedData child element of dss2:InputDocuments or within an underlaying protocol attachment.

The client shall submit one dss2:TransformedData element for each result of applying a sequence of transformations to one of the original documents.

The dss2:TransformedData element shall incorporate the WhichReference attribute.

NOTE: dss2:Base64Data child element of dss2: TransformedData can contain either the transformed document in its dsb:Value child element, or a reference to an underlying protocol attachment (where the transformed document is placed) in its dsb:AttRef child element.

###### 5.1.3.2.2.3 Element dss2:DocumentHash for sending digest of documents

The dss2:DocumentHash element shall implement the digest document container for the XML binding of the protocol.

If the signature(s) to be validated is(are) XAdES or CAdES signature(s), and if the client wants to submit to the server not the signed document, but the digest of the document, then the client shall incorporate the information corresponding to the digest in one dss2:DocumentHash child element of dss2:InputDocuments. The client shall incorporate one dss2:DocumentHash element for each digest of one of the signed documents.

The client shall incorporate the base-64 encoding of this digest value into the dsb:DigestValue child element of the dss2:DigestInfo child element of dss2:DocumentHash.

The client shall incorporate the algorithm identifier of the digest method into the dsb:DigestMethod child element of dss2:DigestInfo child element of dss2:DocumentHash

If the signature(s) to be validated is(are) XAdES then the dss2:DocumentHash element shall incorporate the WhichReference attribute.

 If the signature(s) to be validated is(are) CAdES then the dss2:DocumentHash element shall not incorporate the WhichReference attribute.

#### 5.1.3.3 JSON component

##### 5.1.3.3.1 General requirements

The sigObj element shall implement the signature object container for the JSON binding of the protocol.

The inDocs element shall implement the input documents container for the JSON binding of the protocol.

The requirements governing the presence, cardinalities, and contents of the JSON elements are given in clause 5.1.3.4.

##### 5.1.3.3.2 Additional requirements for contents of inDocs

###### 5.1.3.3.2.1 Element doc for sending original documents

Each item within the doc array shall implement a document container for the JSON binding of the protocol.

NOTE: b64Data child element of one item within the doc array can contain either the signed document in its value child element, or a reference to an underlying protocol attachment (where the signed document is placed) in its attRef child element.

###### 5.1.3.3.2.2 Element transformed for sending transformed documents

Each item within the transformed array shall implement a transformed document container for the JSON binding of the protocol.

If the signature(s) to be validated is(are) XAdES signature(s), and if the client wants to submit to the server not the original document, but the result of the set of transformations applied to it before computing the digest that appears within the ds:DigestManifest child element of the corresponding ds:Reference in the XAdES signature, then the client shall incorporate the base-64 encoding of the binary representation of the result of applying the same sequence of transformations to the original document either into the base64Data child element of transformed child element of inDocs or in an underlying protocol attachment.

The client shall submit one transformed element for each result of applying a sequence of transformations to one of the original documents.

The transformed element shall incorporate the whichRef element child.

NOTE: b64Data component of an item within the transformed array can contain either the transformed document in its value child, or a reference to an underlying protocol attachment (where the transformed document is placed) in its attRef child element.

###### 5.1.3.3.2.3 Element docHash for sending digest of documents

Each item within the docHash array shall implement a transformed document container for the JSON binding of the protocol.

If the signature(s) to be validated is(are) XAdES or CAdES signature(s), and if the client wants to submit to the server not the signed document, but the digest of the document, then the client shall incorporate the information corresponding to the digest in one component of the di array child element of docHash child element of inDocs.

The client shall incorporate the base-64 encoding of this digest value into the value child element of the aforementioned component of di array element.

The client shall incorporate the algorithm identifier of the digest method into the alg child element of the aforementioned component of di array element.

If the signature(s) to be validated is(are) XAdES then the docHash element shall incorporate the whichRef element.

 If the signature(s) to be validated is(are) CAdES then the docHash element shall not incorporate the whichRef element.

#### 5.1.3.4 Cardinalities for elements used for sending signatures, signed documents and representations of signed documents

Table 4 shows the cardinalities of the XML and JSON elements required in this profile for incorporating signature(s), signed documents, signed documents representations (transformed documents and documents digests) for requesting the validation of AdES signatures, depending on the types of the signatures and their relative position with the signed document(s).

Columns in the table show information corresponding to the different types of AdES signatures whose validation is requested, as well as their relative position to the signed document(s).

Rows in the table show different XML and JSON elements, which in the validation request message may appear for incorporating signature(s), signed documents, or representations of the signed documents.

Each cell in the table indicates the required cardinality of the XML element and JSON element (or the number of items within the array if the JSON element is an array) shown in the header of the corresponding row, for the type of signature, located in a relative position to the signed document(s) as indicated in the header of the corresponding column.

Numbers within round brackets identify additional explanatory notes added after the table.

Table . Components for signatures and documents in validation requests messages

for XML and JSON profiles based on OASIS specifications.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  Type of signature and relative positionXML elementsJSON elements | CAdES atached | CAdES detached | Non-embedded XAdES | Embedded XAdES | PAdES |
| dss:SignatureObject | 1 | 1 | 1 | 1 | 1 |
| sigObject |
| dss:Document  | 0 | 0..1 | \* | 1..\* | 1 |
| Items in doc array |
| dss:DocumentHash  | 0 | 0..1 | \* | \* | 0 |
| Items in docHash array  |
| dss:TransformedData  | 0 | 0 | \* | \* | 0 |
| Items in transformed array |

### 5.1.4 Optional components

#### 5.1.4.1 Container for optional components

##### 5.1.4.1.1 Semantics

The validation request message may also contain optional components.

1. One component for identifying the signatures that the server is requested to validate in addition to the signature present or referenced within the signature object container. Clause 5.1.4.2.1 specifies semantic requirements for this component.

NOTE: Each signed document submitted to the server can be signed by more than one signature. This component allows the client to request to the server the validation of some of them.

1. One component for identifying one or more service policies under which the validation shall be conducted. Clause 5.1.4.3.1 specifies semantic requirements for this component.
2. One component for claiming the client’s identity. Clause 5.1.4.4.1 specifies semantic requirements for this component.
3. One component for requesting the server to generate notifications using a certain language. Clause 5.1.4.5.1 specifies semantic requirements for this component.
4. One component for requesting to set the validation time to a certain instant different from the current time. Clause 5.1.4.6.1 specifies semantic requirements for this component.
5. One component for requesting the server to return information on the validation time. Clause 5.1.4.7.1 specifies semantic requirements for this component.
6. One component for requesting the server to return information on the signing time(s). Clause 5.1.4.16.1 specifies semantic requirements for this component.
7. One component for passing to the server validation material in case this is not present within the signature(s) to be validated. Clause 5.1.4.8.1 specifies semantic requirements for this component.
8. Component for requesting the server to return the identity of the signer(s). Clause 5.1.4.9.1 specifies semantic requirements for this component.
9. Component for requesting validation of the signature(s) against a certain signature policy. Clause 5.1.4.10.1specifies semantic requirements for this component.
10. Component for requesting detailed validation report(s) of the validation of the signature(s). Clause 5.1.4.11.1 specifies semantic requirements for this component.
11. Component for requesting that the server signs the validation report(s). Clause 5.1.4.12.1 specifies semantic requirements for this component.
12. In the case of requesting validation of XAdES signatures, one component requesting to the server to return the result of transforming the signed documents, if any transformation is applied. Clause 5.1.4.13.1 specifies semantic requirements for this component.
13. In the case of requesting validation of XAdES signatures, request to return the results of the validation of any signed ds:Manifest present in these signatures. Clause 5.1.4.14.1 specifies semantic requirements for this component.
14. One component for passing to the server one or more time values, each one being according to client’s claim, a Proof of Existence of one signature present within the request.

##### 5.1.4.1.2 XML component

The OptionalInputs child element of VerifyRequest shall be an instance of OptionalInputsVerifyType defined as in XML Schema file "[XSDFILESIGVALPROT]", whose location is detailed in clause A.1, and is copied below for information.

<!—targetNamespace="http://uri.etsi.org/19442/v1.1.1#" -->

<xs:complexType name="OptionalInputsVerifyType">

 <xs:complexContent>

 <xs:extension base="dss:OptionalInputsBaseType">

 <xs:sequence>

 <xs:element ref="ProcessSignatures" minOccurs="0" />

 <xs:element minOccurs="0" name="UseVerificationTime" type="UseVerificationTimeType"/>

 <xs:element default="false" minOccurs="0" name="ReturnVerificationTimeInfo" type="boolean"/>

 <xs:element default="false" minOccurs="0" name="ReturnSigningTimeInfo" type="boolean"/>

 <xs:element maxOccurs="1" minOccurs="0" name="AdditionalKeyInfo" type="dss:AdditionalKeyInfoType"/>

 <xs:element default="false" minOccurs="0" name="ReturnSignerIdentity"" type="boolean"/>

 <xs:element maxOccurs="unbounded" minOccurs="0" name="ReturnTransformedDocument" type="dss:ReturnTransformedDocumentType" />

 <xs:element minOccurs="0" ref="UseSignatureValidationPolicy" />

 <xs:element minOccurs="0" ref="ReturnValidationReport"/>

 <xs:element minOccurs="0" ref="dss:VerifyManifests"/>

 <xs:element minOccurs="0" ref="SignVerificationReport"/>

 <xs:element minOccurs="0" ref="ReturnAugmentedSignature"/>

 <xs:element minOccurs="0" ref="ProofsOfExistence"/>

 </xs:sequence>

 </xs:extension>

 </xs:complexContent>

</xs:complexType>

The dss:OptionalInputs child element of VerifyRequest element shall have at least one dss:Profile child element.

The following children elements of instances of dss:OptionalInputsBaseType type shall not be used: dss:AddTimeStamp and dss:SignatureForm.

In addition to that, the ReturnAugmentedSignature element shall not be present either.

NOTE: The optional child element ReturnAugmentedSignature is the component for requesting augmenting of the signature. The protocol specified in this clause is the “validation protocol”. The ReturnAugmentedSignature will certainly be used within the “augmentation protocol” and the “validation and augmentation protocol”. This element is present in the definition of the OptionalInputsVerifyType type for re-using this XML schema definition in the “validation and augmentation” protocol.

Any other child element of instances of dss:OptionalInputsBaseType type not mentioned and profiled in the present document may be present. Their semantics and syntax shall be as specified in [1].

Children elements xs:any are placeholders for optional inputs that are not defined in the present document.

##### 5.1.4.1.3 JSON component

The oplInp child element of dss-VerifyRequest shall be an instance of OptionalInputsVerifyType defined as in JSONSchema file "[JSONSCHEMAFILESIGVALPROT]", whose location is detailed in clause A.2, and is copied below for information.

 "OptionalInputsVerifyType": {

 "$xsd-type": "OptionalInputsVerifyType",

 "type": "object",

 "properties": {

 "profiles": {

 "type": "array",

 "items": {

 "type": "string"

 }

 },

 "policy": {

 "type": "array",

 "items": {

 "type": "string"

 }

 },

 "claimedIdentity": {

 "type": "object",

 "$ref": "<DSSXCORESCHEMAFILELOCATION>#/definitions/dss-ClaimedIdentityType"

 },

 "lang": {

 "type": "string"

 },

 "useVerificationTime": {

 "type": "object",

 "$ref": "<DSSXCORESCHEMAFILELOCATION>#/definitions/dss-UseVerificationTimeType"

 },

 "returnVerificationTime": {

 "type": "boolean"

 },

 "returnSigningTime": {

 "type": "boolean"

 },

 "addKeyInfo": {

 "type": "object",

 "$ref": "<DSSXCORESCHEMAFILELOCATION>#/definitions/dss-AdditionalKeyInfoType"

 },

 "returnSigner": {

 "type": "boolean"

 },

 "useSigValPol":{

 "type": "object",

 "$ref":"#/definitions/SigValPolicyType "

 },

 "returnAugmentedSig": {

 "type": "string"

 },

 "returnValReport": {

 "type": "boolean"

 },

 "signValReport": {

 "type": "boolean"

 },

 "returnTransformed": {

 "type": "array",

 "items": {

 "type": "object",

 "$ref": "<DSSXCORESCHEMAFILELOCATION>#/definitions/dss-ReturnTransformedDocumentType"

 }

 },

 "retResValManifests": {

 "type": "boolean"

 },

 "requestID": {

 "type": "string"

 },

 "processSigs": {

 "type": "object",

 "$ref": "#/definitions/SigsRefsType"

 },

 "proofsOfExist": {

 "type": "object",

 "$ref": "#/definitions/ProofsOfExistenceType"

 }

 }

 "required": ["profiles"]

 }

The returnAugmentedSign propery shall not be present.

#### 5.1.4.2 Component for identifying the signatures to be validated (signatures-to-process-ids container)

##### 5.1.4.2.1 Introduction

A validation request message may contain more than one signature; however, it may well happen that, under certain use cases, a certain business process does not require all of them to be validated. Consequently validation requests messages need a mechanism for allowing the client to enumerate to the server the signatures that the client requests to validate. This mechanism is provided by the optional input that is fully specified in clause 5.1.4.2.2.

##### 5.1.4.2.2 Semantics

The validate signatures component shall contain one identifier per each signature that the server is requested to validate.

The validate signatures component shall allow reference either any signature present in the validation request.

This component may identify the signature, either:

1. Using the digest value of the signature computed using a specific digest algorithm. This mechanism may be used for identifying any type of signature. In this case, this component:
* shall include one or more digest values (each one computed on one digital signature),
* shall include the identification of one digest algorithm: the algorithm used for computing all the digest values and
* shall this component shall include the identifier of a canonicalization algorithm, if some of the signatures to validate are XAdES signatures.
1. Using a pointer to the embedding document and a XPath expression pointing to that specific embedded XAdES signature within the embedding document. This mechanism may only be used if the signature is an embedded XAdES signature.
2. Using the name of the field where the PAdES signature is embedded within the PDF document. This mechanism may only be used if the signature is a PAdES signature

##### 5.1.4.2.3 XML component

The ProcessSignatures optional input shall be an instance of SignatureIdentifiersType defined as in XML Schema file "[XSDFILESIGVALPROT]", whose location is detailed in clause A.1, and is copied below for information.

NOTE 23/5/2018. ADD THE OTHER ALTERNATIVES FOR IDENTIFYING THE SIGNATURES (XADESPTR AND PADESPTR). CHECK IF ADDING A SIGNING CERTIFICATE WITH THE SEMANTICS OF “VALIDATE ONLY THE SIGNATURES SIGNED WITH THIS SIGNING CERTIFICATE” , OR THE IDENTIFIER OF A CERTAIN CERTIFICATION SERVICE WITH THE SEMANTICS OF “VALIDATING ONLY SIGNATURES SIGNED WITH A CERTIFICATE WHOSE ISSUER IS THAT SERVICE”, WOULD ALSO BE WORTHWHILE.

<!—targetNamespace="http://uri.etsi.org/19442/v1.1.1#" 🡪

<xs:element name="ProcessSignatures" type="SignaturesReferencesType"/>

<xs:complexType name="SignaturesReferencesType">

 <xs:sequence >

 <xs:element ref="DigestReferences" minOccurs="0"/>

 <xs:element ref="XAdESSignaturePtr" minOccurs="0" maxOccurs="unbounded"/>

 <xs:element name="PAdESFieldName" type="xs:string" minOccurs="0" maxOccurs="unbounded"/>

 </xs:sequence>

</xs:complexType>

<xs:element name="DigestReferences" type="DigestReferencesType"/>

<xs:complexType name="DigestReferencesType">

 <xs:sequence >

 <xs:element name="CanonicalizationMethod" type="xs:anyURI" minOccurs="0"/>

 <xs:element name="DigestMethod" type="xs:anyURI"/>

 <xs:element name="DigestValue" type="xs:base64Binary" maxOccurs="unbounded"/>

 </xs:sequence>

</xs:complexType>

<xs:element name="XAdESSignaturePtr" type="XAdESSignaturePtrType"/>

<xs:complexType name="XAdESSignaturePtrType">

 <xs:sequence>

 <xs:element maxOccurs="unbounded" minOccurs="0" name="NsPrefixMapping" type="dsb:NsPrefixMappingType"/>

 </xs:sequence>

 <xs:attribute name="WhichDocument" type="xs:IDREF" use="optional"/>

 <xs:attribute name="XPath" type="xs:string" use="optional"/>

 <xs:attribute name="SchemaRefs" type="xs:IDREFS" use="optional"/>

</xs:complexType>

DigestReferences child element may be used for referencing any type of signature within the request.

CanonicalizationMethod child element’s value shall be an URI identifying a canonicalization algorithm

DigestMethod child element’s value shall be an URI identifying a digest algorithm.

DigestValue child element’s value shall be the base-64 encoded value of the digest of the referenced digital signature computed using the digest algorithm identified in DigestMethod child element’s value.

The actual computation of the digest value shall be dependant on the type of signature and shall be performed as follows:

1. In case of CAdES signatures, the input to the digest value computation shall be one of the DER-encoded instances of SignedInfo type present within the CMS structure.

NOTE: A CMS structure may enclose several parallel CAdES signatures (each instance of SignedInfo type contain the digital signature value generated by a different private key).

1. In case of XAdES signatures, the input of the digest value computation shall be the result of applying the canonicalization algorithm identified within the CanonicalizationMethod child element’s value to the corresponding ds:Signature element and its contents. The canonicalization shall be computed keeping this ds:Signature element as a descendant of the XML root element, without detaching it .
2. In case of PAdES signatures, the input of the digest value computation shall be the result of decoding the hexadecimal string present within the Contents field of the Signature PDF dictionary enclosing one PAdES digital signature.

XAdESSignaturePtr child element may only be used for referencing XAdES signatures.

XAdESSignaturePtr child element is an instance of XAdESSignaturePtrType type, whose requirements are the same as the requirements for dss2:SignaturePtrType specified in clause 4.2.9 of DSS-X core v2.0 [1], with the following exceptions:

1. The attribute WhichDocument may be absent. If this attribute is present, its value shall be as specified in 4.2.9 of DSS-X core v2.0 [1] and the XAdESSignaturePtr element shall reference a signature embedded in some input document. If this attribute is absent, the XAdESSignaturePtr element shall reference one of the signatures placed within the signature object container.
2. The SchemaRefs attribute shall have the same semantics and usage as SchemaRefs attribute in DocumentBaseType of OASIS Standard: Digital Signature Service Core Protocols, Elements, and Bindings Version 2.0 [1].

PAdESFieldName child element may only be used for referencing PAdES signatures.

The value of PAdESFieldName child element shall be the name of a PDF field where a PAdES signature is present within the PDF signed document.

##### 5.1.4.2.4 JSON component

The processSigs element shall be an instance of SigsRefsType defined as in JSON Schema file "[JSONSCHEMAFILESIGVALPROT]", whose location is detailed in clause A.2, and is copied below for information

 "SigsRefsType": {

 "$xsd-type": "SigsRefsType",

 "type": "object",

 "properties": {

 "digRefs": {

 "type": "object",

 "properties": {

 "digVals": {

 "type": "array",

 "items": {

 "type": "string"

 }

 },

 "digAlg": {

 "type": "string"

 },

 "canAlg": {

 "type": "string"

 }

 }

 "required": ["digVals", "digAlg"]

 },

 "padesFieldN": {

 "type": "stromg"

 },

 "xadesSigPtrs": {

 "type": "array",

 "items": {

 "type": "object",

 "properties": {

 "nsDecl": {

 "type": "array",

 "items": {

 "type": "object",

 "$ref": "<DSSXCORESCHEMAFILELOCATION>#/definitions/dsb-NsPrefixMappingType"

 }

 },

 "whichDoc": {

 "$ref": "#/definitions/dss2-DocumentBaseType"

 },

 "xPath": {

 "type": "string"

 },

 "SchemaRefs": {

 "type": "string"

 }

 }

 }

 }

 }

 }

Below follow the requirements that apply to this element:

1. Each item within digVals array shall have the same requirements as the DigestValue element in the previous clause, including those ones that apply to the computation of their values.
2. The property digAlg shall have the same requirements as the DigestMethod element in the previous clause.
3. The property canAlg shall have the same requirements as the CanonicalizationMethod element in the previous clause.

xadesSigPtrs element may only be used for referencing XAdES signatures.

xadesSigPtrs element is an array of instances of XAdESSignaturePtrType type, whose requirements are the same as the requirements for dss2-SignaturePtrType specified in clause 4.2.9.3 of DSS-X core v2.0 [1], with the following exceptions:

1. The element whichDoc may be absent. If this element is present, its value shall be as specified in 4.2.9 of DSS-X core v2.0 [1] and the corresponding item of the array shall reference a signature embedded in some input document. If this element is absent, the corresponding item of the array shall reference one of the signatures placed within the signature object container.
2. The SchemaRefs element shall have the same semantics and usage as SchemaRefs attribute in clause 4.2.2.3 of OASIS Standard: Digital Signature Service Core Protocols, Elements, and Bindings Version 2.0 [1].

pAdESFieldN element may only be used for referencing PAdES signatures.

The value of pAdESFieldN element shall be the name of a PDF field where a PAdES signature is present within the PDF signed document.

##### 5.1.4.2.5 Processing model

If this component is present, the server shall validate the signatures referenced by this component. If some of the digest values within this component cannot be matched to a digest of any of the signatures present within the request message, the server shall generate an IndividualResult element with its major result set to urn:oasis:names:tc:dss:1.0:resultmajor:RequesterError, its minor result set to [http://uri.etsi.org/19442/v1.1.1#](http://uri.etsi.org/19442/v1.1.1)SignatureNotLocated.

If this component is not present, the server shall validate all the signatures found within the validation request message, as specified in DSS-X core v2.0 [1].

#### 5.1.4.3 Component for identifying under which service policy the validation has to be conducted

##### 5.1.4.3.1 Component semantics

This component shall contain a non-ambiguous identifier of the service policy under which the client requests the server to validate the signature.

##### 5.1.4.3.2 XML component

The element that shall identify under which service policy the validation has to be conducted shall be the dss:ServicePolicy element specified in clause [OPTIONALINPUTSBASETYPE\_XML ] of [1].

NOTE: The service policy is not the same as signature policy. The server defines the service policy, and one server may have different service policies offering different features to its clients.

##### 5.1.4.3.3 JSON component

The element that shall identify under which service policy the validation has to be conducted shall be the policy element. The contents of this element shall be as specified in clause [SERVICEPOLICY\_JSON] of [1].

#### 5.1.4.4 Component for allowing the client to claim its identity

##### 5.1.4.4.1 Component semantics

This component shall provide the identity of the client as a string-valued name.

This component may include sub-components for supporting names federation.

This component may include one sub-component for identifying the format of the string-valued name representing the identity of the client.

This component may include one sub-component for integrating to the string-valued name representing the identity of the client, a different name identifier that has been established by the validation service itself for the client.

This component may also include one sub-component for incorporating any type of additional supporting information for the string-valued name representing the identity of the client.

##### 5.1.4.4.2 XML component

The element that shall allow the client claiming for an identity shall be the dss:ClaimedIdentity element specified in clause [OPTIONALINPUTSBASETYPE\_XML ] of [1].

JC REMARK: IT IS FORESEEN THAT IN VERSION 2.0 OF THE CORE, THE claimedIdentity is built around saml2:NameID element.

NOTE: The dss:ClaimedIdentity element builds on saml2:NameID element and it incorporates all the XML descendant elements and attributes for matching the semantic requirements in 5.1.4.4.

##### 5.1.4.4.3 JSON component

The element that shall allow the client claiming for an identity shall be claimedIdentity , an instance of dss-ClaimedIdentityType. This type is specified in clause [CLAIMEDIDENTITYTYPE\_JSON] of [1].

#### 5.1.4.5 Component for requesting notifications in a certain language

##### 5.1.4.5.1 Component semantics

This component shall identify a language by a string-valued identifier, whose value shall be one of the identifiers built and registered as specified in RFC 5646 [12].

##### 5.1.4.5.2 XML component

The element that shall request the server to return notifications in a certain language shall be the dss:Language element specified in clause [OPTIONALINPUTSBASETYPE\_XML ] of [1].

The value of this element shall be string compliant with the values defined in RFC 5646 [12].

##### 5.1.4.5.3 JSON component

The element that shall request the server to return notifications in a certain language shall be the lang element. The contents of this element shall be as specified in clause [NOTIFICATIONSLANGUAGE\_JSON] of [1].

The value of this element shall be string compliant with the values defined in RFC 5646 [12].

#### 5.1.4.6 Component for requesting to set the validation time to a certain instant different from current time

##### 5.1.4.6.1 Component semantics

This component shall provide means for indicating to the server that the validation time is either the current time (the time when the server performs the signature validation) or a certain time in the past.

##### 5.1.4.6.2 XML component

The element that shall allow to set the validation time to a certain instant different from current time shall be the UseVerificationTime element instance of dss:UseVerificationTimeType specified in clause [USEVALIDATIONTIME] of [1].

The value of dss:SpecificTime child element shall be expressed as Coordinated Universal Time (UTC): its value shall contain year with four digits, month, day, hour, minute, second (without decimal fraction) and the UTC designator "Z". The time scale shall be based on the second.

EDITOR’S REMARK: THIS IS THE TEXT IN TRUSTED LIST CLAUSE 5.1.3 OF 119 612.

##### 5.1.4.6.3 JSON component

The element that shall allow to set the validation time to a certain instant different from current time shall be the useVerificationTime element, instance of dss-UseVerificationTimeType. This type is specified in clause [ USEVALIDATIONTIMETYPE\_JSON] of [1].

The value of specTime child element shall be expressed as Coordinated Universal Time (UTC): its value shall contain year with four digits, month, day, hour, minute, second (without decimal fraction) and the UTC designator "Z". The time scale shall be based on the second

#### 5.1.4.7 Component for requesting to return the validation time

##### 5.1.4.7.1 Component semantics

This component shall provide means for requesting to the server to return within the response an indication of the validation time.

##### 5.1.4.7.2 XML component

The element that shall allow to request to the server to return the validation time shall be the ReturnVerificationTime element instance of dss:ReturnVerificationTimeType specified in clause [RETURNVALIDATIONTIME] of [1]

##### 5.1.4.7.3 JSON component

The element that shall allow to set the validation time to a certain instant different from current time shall be the returnVerificationTime element. The contents of this element shall be as specified in clause [RETURNVALIDATIONTIMETYPE\_JSON] of [1].

#### 5.1.4.8 Component for passing validation material to the server

##### 5.1.4.8.1 Component semantics

This element shall convey any type of validation material that the client decides to pass to the server. Specifically this component shall provide mechanisms for passing to the server X509 certificates, Attribute certificates, CRLs, OCSP responses, or other type of validation data.

##### 5.1.4.8.2 XML component

The element that shall allow the client to pass validation material to the server shall be the AdditionalKeyInfo element instance of dss:AdditionalKeyInfoType specified in clause [ADDITIONALKEYINFO 5.5.4] of [1].

##### 5.1.4.8.3 JSON component

The element that shall allow the client to pass validation material to the server shall be the addKeyInfo element, instance of dss:AdditionalKeyInfoType. This type specified in clause [ADDITIONALKEYINFOTYPE\_JSON] of [1].

#### 5.1.4.9 Component for requesting the server to return the identity of the signer

##### 5.1.4.9.1 Component semantics

This element shall convey an indication to the server for returning the identity of the signer(s).

##### 5.1.4.9.2 XML component

The element that shall request the server to return the identity of the signer shall be the ReturnSignerIdentity element. This element shall have the same contents and requirements as the dss:ReturnSignerIdentity element specified in clause [RETURNSIGNEDIDENTITY 5.5.7] of [1].

##### 5.1.4.9.3 JSON component

The element that shall request the server to return the identity of the signer shall be the returnSigner element. The contents of this element shall be as specified in clause [RETURNSIGNEDIDENTITY\_JSON 3.1.15.2] of [1].

#### 5.1.4.10 Component for requesting validation against a certain signature policy

##### 5.1.4.10.1 Component semantics

This component shall provide means for unambiguously identifying the signature validation policy against the client requests to validate the digital signature(s).

This component shall also allow the client indicate locations where the signature validation policy may be downloaded from, in case the client wants to indicate them to the server.

##### 5.1.4.10.2 XML component

The element that shall request the server to validate the signature(s) against a certain signature validation policy shall be the UseSignatureValidationPolicy element.

The UseSignatureValidationPolicy element shall be defined as in XML Schema file "[XSDFILESIGVALPROT]", whose location is detailed in clause A.1, and is copied below for information.

<!—targetNamespace="http://uri.etsi.org/19442/v1.1.1#" -->

 <xs:element name="UseSignatureValidationPolicy" type="UseSignatureValidationPolicyType"/>

 <xs:complexType name=="UseSignatureValidationPolicyType">

 <xs:sequence>

 <xs:element name="SignatureValidationPolicyID" type="xs:anyURI"/>

 <xs:element name="SignaturePolicyLocation" type="xs:anyURI" minOccurs="0" maxOccurs="unbounded"/>

 </xs:sequence>

 </xs:complexType>

The SignatureValidationPolicyID child element shall have as value the unique identifier of the signature validation policy as an URI. If the identifier of the signature validation policy is an OID, then the value of this element shall be an URN indicating the value of the aforementioned OID as specified in RFC 3061 [18].

Every SignatureValidationPolicyLocation child element shall have as value one location where the signature validation policy document can be accessed, as an URI value.

##### 5.1.4.10.3 JSON component

The element that shall request the server to validate the signature(s) against a certain signature validation policy shall be the useSigValPol element.

The useSigValPol element shall be an instance of SigValPolicyType defined as in JSON Schema file "[JSONSCHEMAFILESIGVALPROT], whose location is detailed in clause A.2, and is copied below for information.

 "SigValPolicyType": {

 "type": "object",

 "properties": {

 "sigValPolID": {

 "type": "string"

 },

 "sigValPolLocs": {

 "type": "array",

 "items": {

 "type": "string"

 }

 }

 }

 }

The SigValPolID child element shall have as value the unique identifier of the signature validation policy as an URI. If the identifier of the signature validation policy is an OID, then the value of this element shall be an URN indicating the value of the aforementioned OID as specified in RFC 3061 [18].

Every SigValPolLos child element shall have as value one location where the signature validation policy document can be accessed, as an URI value.

##### 5.1.4.10.4 Processing model

When UseSignatureValidationPolicy/useSigValPol is present in the request the server shall process this element as follows:

1. If the server is able to validate signatures against this signature validation policy it shall proceed to the validation. Once finalized the validation, if the server has been able to check all the constraints defined within the signature validation policy on all the validated signatures, it shall incorporate into the response to the client the AppliedSignatureValidationPolicy/appliedSigValPolicy specified in clause 5.2.3.6.2/5.2.3.6.3.

NOTE: The way used by the server to conclude whether it is able to validate signatures against a certain signature validation policy is out of the scope of the present document. Access to the signature validation policy document using SignatureValidationPolicyLocation/sigValPolLocs child elements can help in this process.

1. If the server is not able to validate signatures against this signature validation policy it may decide either:
* not to proceed to validate the signature. Under these circumstances it:
	+ shall incorporate in the response to the client a dss:Result/result element with the dss:ResultMajor/maj set to "urn:oasis:names:tc:dss:1.0:resultmajor:Success", and a dss:ResultMinor/min set to <http://uri.etsi.org/19442/v1.1.1/notFeasibleSignatureValidationPolicy>, and
	+ shall incorporate in the response to the client the AvailableSignatureValidationPolicies/availableSigValPols child element as specified in clause 5.2.3.7.2/5.2.3.7.3. The server may include in this element a list of the identifiers of the signature validation policies against which it is able to validate digital signatures.
* Proceed to validate the signature against a different signature policy. Once finalized the validation, if the server has been able to check all the constraints defined in this other signature validation policy on all the validated signatures, the server shall incorporate in the response the AppliedSignatureValidationPolicy/appliedSigValPol element identifying the signature policy used by the server for validating the signature.

#### 5.1.4.11 Component for requesting a detailed validation report (as the one to be specified in TS 119 102-2)

##### 5.1.4.11.1 Component semantics

This component shall provide means for requesting to the server the generation and return of a detailed validation report. If these mechanisms are used within this component, then the component specified in clause 5.1.4.9 should not be present. If the component specified in clause 5.1.4.9 is present within the request, then these mechanisms should not be used in the component being specified in the present clause.

TOBEDISCUSSED

This component shall provide means for requesting to the server to convey in the detailed validation report the binary values, base 64-encoded, of the X509 certificates, and Attribute certificates, verified during the signature validation.

This component shall provide means for requesting to the server to convey in the detailed validation report the binary values, base 64-encoded, of the revocation values (CRLs, OCSP responses or other), used during the signature validation.

This component shall provide mechanisms for requesting to the server to convey in the detailed validation report textual relevant information on the validation material used during the signature validation.

This component shall provide mechanisms for identifying one level of details among a set of possible different levels of details for the validation report.

##### 5.1.4.11.2 XML component

The element that shall request the server to return a detailed validation report on each signature validated shall be the ReturnValidationReport element.

The ReturnValidationReport element shall be defined as in XML Schema file "[XSDFILESIGVALPROT]", whose location is detailed in clause A.1, and is copied below for information

<!—targetNamespace="http://uri.etsi.org/19442/v1.1.1#" 🡪

 <xs:element name="ReturnValidationReport" type="boolean" default="false"/>

##### 5.1.4.11.3 JSON component

The element that shall request the server to return a detailed validation report shall be the returnValReport element set to value “true”.

The server shall interpret the absence of this element or its presence set to value “false” as absence of this request.

#### 5.1.4.12 Component for requesting that the validation report is signed by the server a detailed validation report (as the one to be specified in TS 119 102-2)

##### 5.1.4.12.1 Component semantics

The presence of this element shall indicate that the client is requesting that the server signs the validation report generated after validating the signature(s) in the validation request.

Absence of this element shall indicate that the client requests no signature of the validation report.

##### 5.1.4.12.2 XML component

The element that shall request the server to sign the detailed validation report shall be the SignVerificationReport.

The SignVerificationReport element shall be defined as in XML Schema file "[XSDFILESIGVALPROT]", whose location is detailed in clause A.1, and is copied below for information.

<!—targetNamespace="http://uri.etsi.org/19442/v1.1.1#" 🡪

 <xs:element minOccurs="0" name="SignVerificationReport"/>

##### 5.1.4.12.3 JSON component

The element that shall request the server to return a detailed validation report shall be the signValReport element set to value “true”.

The server shall interpret the absence of this element or its presence set to value “false” as absence of this request.

#### 5.1.4.13 Component for requesting the server to return the result of transforming the input document

##### 5.1.4.13.1 Component semantics

This component shall not be present when requesting validation of CAdES or PAdES signatures.

This component shall provide means for identifying one or more signed documents whose transformed versions the client requests that the server incorporates into the validation response.

The server shall interpret the absence of this component or its presence set to value "false" as absence of this request.

##### 5.1.4.13.2 XML component

The element that shall request the server to return the result of transforming one input document shall be the ReturnTransformedDocument element instance of dss-ReturnTransformedDocumentType specified in clause [RETURNTRANSFORMEDDOCUMENT 5.5.9] of [1].

The server shall interpret the absence of this element or its presence set to value "false" as absence of this request.

##### 5.1.4.13.3 JSON component

The element that shall request the server to return the result of transforming one input document shall be the returnTransformed element, instance of dss-ReturnTransformedDocumentType. This type is specified in clause [RETURNTRANSFORMEDDOCUMENTTYPE\_JSON 3.1.45.2] of [1].

#### 5.1.4.14 Component for requesting to return the validation of signed ds:Manifest in XAdES signatures

##### 5.1.4.14.1 Component semantics

This component shall not be present when requesting validation of CAdES or PAdES signatures.

This component shall notify that the client requests that the server verifies all the ds:Manifest elements present within the XAdES signature.

The server shall interpret the absence of this component or its presence set to value "false" as absence of this request.

##### 5.1.4.14.2 XML component

The element that shall request the server to return the result of transforming one input document shall be the dss:VerifyManifests element specified in clause [VERIFYMANIFESTS 5.5.1.1] of [1].

##### 5.1.4.14.3 JSON component

The element that shall request the server to return the result of transforming one input document shall be retResValManifests, set to value “true”.

#### 5.1.4.15 Component for identifying a request

##### 5.1.4.15.1 Component semantics

This component shall assign an identifier to the request. It allows correlating a certain response with the request with this identifier.

##### 5.1.4.15.2 XML component

The element that shall request the server to return the result of transforming one input document shall be the RequestID attribute specified in clause [REQUESTBASETYPE 3.1.21.1] of [1].

##### 5.1.4.15.3 JSON component

The element that shall identify the request shall be requestID, whose value shall be a string.

#### 5.1.4.16 Component for requesting return of the signing time

##### 5.1.4.16.1 Component semantics

This component shall provide means for requesting to the server to return within the response an indication of the signing time.

##### 5.1.4.16.2 XML component

The element that shall allow to request to the server to return an indication of the signing time shall be the ReturnSigningTimeInfo element specified in clause [RETURNIGNINGTIONTIME] of [1]

##### 5.1.4.16.3 JSON component

The element that shall allow to request to the server to return an indication of the signing time shall be the returnSigningTime element. The contents of this element shall be as specified in clause [RETURNSIGNINGTIMETYPE\_JSON] of [1].

#### 5.1.4.17 Component for passing proofs of existence of one or more signatures

##### 5.1.4.17.1 Component semantics

This component shall have one or more tuples. Each tuple shall contain two components, namely:

1. One component whose value is a time instant value. This time instant shall be considered by the server as a proof of existence of the signature referenced in the other component of the tuple.
2. One component referencing one signature.

##### 5.1.4.17.2 XML component

The ProofsOfExistence optional input shall be an instance of ProofsOfExistenceType defined as in XML Schema file "[XSDFILESIGVALPROT]", whose location is detailed in clause A.1, and is copied below for information.

<!—targetNamespace="http://uri.etsi.org/19442/v1.1.1#" -->

<xs:element name="ProofsOfExistence" type="ProofsOfExistenceType">

<xs:complexType name="ProofsOfExistenceType">

 <xs:sequence>

 <xs:element ref="ProofOfExistence" maxOccurs="unbounded" />

 </xs:sequence>

</xs:complexType>

<xs:element name="ProofOfExistence" type="ProofOfExistenceType">

<xs:complexType name="ProofOfExistenceType">

 <xs:sequence>

 <xs:element name="Time" type="xs:dateTime"/>

 <xs:element ref="SignatureReference"/>

 </xs:sequence>

</xs:complexType>

<xs:element name="SignatureReference" type="SignatureReferenceType">

<xs:complexType name="SignatureReferenceType">

 <xs:choice>

 <xs:sequence>

 <xs:element name="CanonicalizationMethod" type="xs:anyURI" minOccurs="0"/>

 <xs:element name="DigestMethod" type="xs:anyURI"/>

 <xs:element name="DigestValue" type="xs:base64Binary"/>

 </xs:sequence>

 <xs:element ref="XAdESSignaturePtr" />

 <xs:element name="PAdESFieldName" type="xs:string" />

 </xs:choice>

</xs:complexType>

ProofsOfExistence element shall be a sequence of ProofOfExistence children elements.

SignatureReference child element of ProofOfExistence shall be a reference to the signature whose Proof of Existence the client sends to the server. Any of the three mechanisms specified in clause 5.1.4.2.3 may be used for referencing the signature, depending of the type of the referenced signature.

SignatureReference child element of ProofOfExistence shall indicate the time and date when the client claims the referenced signature existed (Proof of Existence).

##### 5.1.4.17.3 JSON component

The proofsOfExist element shall be an instance of ProofsOfExistenceType defined as in JSON Schema file "[JSONSCHEMAFILESIGVALPROT]", whose location is detailed in clause A.2, and is copied below for information

 "ProofsOfExistenceType": {

 "$xsd-type": "ProofsOfExistenceType",

 "type": "array",

 "items": {

 "type": "object",

 "properties": {

 "time": {

 "type": "string"

 },

 "sigRef": {

 "type": "object",

 "$ref": "<DSSXCORESCHEMAFILELOCATION>#/definitions/SignatureReferenceType"

 }

 }

 }

 }

 "SignatureReferenceType": {

 "type": "object",

 "properties": {

 "digRefs": {

 "type": "object",

 "properties": {

 "digVals": {

 "type": "array",

 "items": {

 "type": "string"

 }

 },

 "digAlg": {

 "type": "string"

 },

 "canAlg": {

 "type": "string"

 }

 }

 "required": ["digVals", "digAlg"]

 },

 "padesFieldN": {

 "type": "stromg"

 },

 "xadesSigPtrs": {

 "type": "array",

 "items": {

 "type": "object",

 "properties": {

 "nsDecl": {

 "type": "array",

 "items": {

 "type": "object",

 "$ref": "<DSSXCORESCHEMAFILELOCATION>#/definitions/dsb-NsPrefixMappingType"

 }

 },

 "whichDoc": {

 "$ref": "#/definitions/dss2-DocumentBaseType"

 },

 "xPath": {

 "type": "string"

 },

 "SchemaRefs": {

 "type": "string"

 }

 }

 }

 }

 }

 }

Instances of ProofsOfExistenceType type shall be an array of items. Each item shall be a tuple of two components.

sigRef component of the tuple shall be a reference to the signature whose Proof of Existence the client sends to the server. Any of the three mechanisms specified in clause 5.1.4.2.4 may be used for referencing the signature, depending of the type of the referenced signature.

Time component of the tuple shall indicate the time and date when the client claims the referenced signature existed (Proof of Existence). Its value shall be an string representing an UTCTime as represented in instances of xs:dateTime type in XML schema.

##### 5.1.4.17.4 Processing model

If the validation request incorporates this optional input, the server shall consider the time indicated in one tuple as a Proof of Existence for signature referenced within the same tuple. It shall use this Proof of Existence during the validation of that signature.

The client is fully responsible of submitting accurate Proofs of Existence to the server.

## 5.1.5 Asynchronous processing

#### 5.1.5.1 Asynchronous processing protocol

In asynchronous processing one client usually sends an initial request to the server.

The initial request shall contain, among other things, a request identifier generated by the client, as specified in clause 5.1.4.15.

The server can return a response indicating that the validation has not yet been finished. Within this initial response, the server shall convey a response identifier, as specified in clause 5.2.3.12. Both client and server can correlate the response identifier to the request identifier.

Under this processing mode the client, after a certain time, can send a pending-request to the server. This pending-request shall include the response identifier previously returned by the server. This response identifier allows the server to correlate this pending-request to the initial request, and can return the validation result or return again an indication of “not yet finished”.

If this is the case, the client can send subsequent requests until the server returns a response with the validation result. Each subsequent request shall include the response identifier returned by the server in the response to the initial request.

For managing asynchronous processing, the following components are required:

1. One component for the identifying a request as a pending-request associated to an initial request.
2. One component that allows correlating the subsequent requests with the response to the initial requests, which included a notification of “not yet finished”.

#### 5.1.5.2 Component for identifying a request as a subsequent request to an initial request

##### 5.1.5.2.1 Semantics

This component shall be a request that shall notify to the server to return the response corresponding to a previously sent initial request to the server. Requests of this type are named pending-request hereinafter.

This component shall include an identifier, generated by the server and returned by the server in the response to the initial request, which shall allow correlating the subsequent requests to the initial request and to the subsequent responses.

##### 5.1.5.2.2 XML component

The element that shall indicate to the server that the client is requesting the response corresponding to a previously sent initial request (as part of an asynchronous protocol) shall be the dss2:PendingRequest element specified in clause [PENDINGREQUEST 3.1] of OASIS Standard: Digital Signature Service Core Protocols, Elements, and Bindings Version 2.0 [1].

##### 5.1.5.2.3 JSON component

The element that shall indicate to the server that the client is requesting the response corresponding to a previously sent initial request (as part of an asynchronous protocol) shall be the PendingRequest element.

 The PendingRequest element shall be defined as in JSON Schema file "[JSONSCHEMAFILESIGVALPROT], whose location is detailed in clause A.2, and is copied below for information.

"PendingRequest":{

 "type": "object",

 "properties": {

 "inDocs": {

 "type": "object",

 "$ref": "<DSSXCORESCHEMAFILELOCATION>#/definitions/dss-InputDocumentsType"

 },

 "reqID": {

 "type": "string"

 },

 "optInp": {

 "type": "object",

 "$ref": "#/definitions/OptionalInputsVerifyType"

 },

 "sigObj": {

 "type": "object",

 "$ref": "<DSSXCORESCHEMAFILELOCATION>#/definitions/dss-SignatureObjectType"

 }

 }

 }

Components inDocs, optInp, and sigObj, shall have the same semantics and syntaxes as the components of dss-VerifyRequest with the same names.

#### 5.1.5.3 Component for correlating subsequent requests to the initial response

##### 5.1.5.3.1 Semantics

This component of pending request shall contain an identifier for correlating it to a certain initial request.

##### 5.1.5.3.2 XML component

The element that shall contain an identifier for correlating subsequent pending-requests associated to a certain initial request shall be the dss2:ResponseID as specified in clause [ASYNCORRELATINGID\_INRESP 5.2.2] of OASIS Standard: Digital Signature Service Core Protocols, Elements, and Bindings Version 2.0 [1].

##### 5.1.5.3.3 JSON component

The element that shall contain an identifier for correlating subsequent pending-requests associated to a certain initial request shall be the respID whose value shall be as specified in clause [ASYNCORRELATINGID\_INRESP 5.2.2] of OASIS Standard: Digital Signature Service Core Protocols, Elements, and Bindings Version 2.0 [1].

## 5.2 Response message

### 5.2.1 Component for responding to a validation request

#### 5.2.1.1 Component semantics

The validation response message resulting from one request of validation of AdES signature(s), shall contain one component for notifying the global validation result.

The validation response message may contain one or more signature results containers. Clause 5.2.3.1 specifies this component and contains references to clauses that specify requirements for its sub-components.

NOTE: The response message to a minimum signature validation request with only one signature and without any optional input is a response message with a global result component and the component identifying the response as compliant with the “validation” protocol defined in the present document.

The validation response message resulting from one request of validation of AdES signature(s), shall contain one component indicating that the response has been built using the profile specified in the present document.

#### 5.2.1.2 XML component

The element that shall be the component for responding to the validation request of AdES signature(s) shall be the root element of the message VerifyResponse as specified in the present clause.

The VerifyResponse element shall be defined as in XML Schema file "[XSDFILESIGVALPROT]", whose location is detailed in clause A.1, and is copied below for information.

<!—targetNamespace="http://uri.etsi.org/19442/v1.1.1#" 🡪

 <xs:element name="VerifyResponse" type="VerifyResponseType"/>

 <xs:complexType name="VerifyResponseType">

 <xs:extension base="dsb:ResponseBaseType">

 <xs:sequence>

 <xs:element name="OptionalOutputs" type="OptionalOutputsVerifyType" minOccurs="0" />

 </xs:sequence>

 </xs:extension>

 </xs:complexType>

NOTE: The OptionalOutputs child element is a container for one or more signature result container components.

#### 5.2.1.3 JSON component

The element that shall be the component for responding to the validation request of AdES signature(s) shall be the root element of the message VerifyResponse as specified in the present clause.

The VerifyResponse element shall be defined as in JSON Schema file "[JSONSCHEMAFILESIGVALPROT], whose location is detailed in clause A.2, and is copied below for information.

 " VerifyResponse": {

 "$xsd-type": "VerifyResponse",

 "type": "object",

 "properties": {

 "result": {

 "type": "object",

 "$ref": "<DSSXCORESCHEMAFILELOCATION>#/definitions/dss-ResultType"

 },

 "reqID": {

 "type": "string"

 },

 "profile": {

 "type": "array",

 "items": {

 "type": "string"

 }

 },

 "optOutp": {

 "type": "object",

 "$ref": "#/definitions/OptionalOutputsVerifyType"

 }

 }

 "required": ["result", "profile"]

 }

The profile array shall have one item, whose value shall be [http://uri.etsi.org/19442/v1.1.1/validationprofile#](http://uri.etsi.org/19442/v1.1.1/validationprofile), identifying the response as a validation response compliant with the validation profile specified in the present document.

NOTE: The optOutp child element implements the signature result container component.

### 5.2.2 Component for the global validation result

#### 5.2.2.1 Component semantics

This component shall contain a major result, which shall report whether the server has been able to perform its task, regardless the results obtained. This component may also contain a minor result providing additional information on the task performed by the server.

#### 5.2.2.2 Processing model

If the request did not contain the signatures-to-process-ids container and there was only one signature within the validation request message and its underlying protocol attachments, this component shall have the semantics of component Result as specified in OASIS Standard: Digital Signature Service Core Protocols, Elements, and Bindings Version 2.0 [1] and the server shall process the signature as specified in OASIS Standard: Digital Signature Service Core Protocols, Elements, and Bindings Version 2.0 [1].

Otherwise, the result major child shall have one of the values and semantics specified in [1] and if the result major child indicates that the server has successfully processed the request, then the result minor child shall have the following value:

http://uri.etsi.org/19442/v1.1.1#CheckIndividualResults

This value shall indicate that there shall be as many signature result containers as signatures validated by the server and that each signature output component shall contain a result component providing details of the result of validating that signature.

In this case, each signature shall be individually validated as specified OASIS Standard: Digital Signature Service Core Protocols, Elements, and Bindings Version 2.0 [1].

#### 5.2.2.3 XML component

The element that within the response shall notify the validation result shall be the dsb:Result element specified in clause [RESULT 3.1.12.1] of [1].

#### 5.2.2.4 JSON component

The element that within the response shall notify the validation result shall be an instance of the ResultType type specified in clause [RESULT\_JSON 3.1.12.2] of [1].

### 5.2.3 Optional components

#### 5.2.3.1 Container for optional components

##### 5.2.3.1.1 Semantics

The validation response message may also include one or more signature results containers.

Each signature result container shall include one or more optional outputs resulting from the validation of the corresponding signature.

Each signature result container may include the following components:

1. One component for referencing the signature that the rest of optional outputs within the signature results container correspond to.
2. One component for identifying the service policy under which the validation was conducted. Clause 5.2.3.3 specifies requirements for this component.
3. One component for indicating the time when the validation was conducted (validation time). Clause 5.2.3.4 specifies requirements for this component.
4. One component for indicating the signing time. Clause 5.2.3.13 specifies requirements for this component.
5. One component for indicating the identity of the signer(s). Clause 5.2.3.5 specifies requirements for this component.
6. One component for notifying the signature policy applied during the validation. Clause 5.2.3.6 specifies requirements for this component.
7. One component for notifying the set of signature policies supported by the server. Clause 5.2.3.7 specifies requirements for this component.
8. One component for returning the detailed validation report. Clause 5.2.3.8 specifies requirements for this component.
9. One component for returned the detailed validation report signed by the validation server. Clause 5.2.3.9 specifies requirements for this component.
10. One component for returning the result of transforming the input document. Clause 5.2.3.10 specifies requirements for this component.
11. One component for returning the results of validating any signed ds:Manifest present in the signature(s). Clause 5.2.3.11 specifies requirements for this component.
12. One component for supporting asynchronous processing of the response messages. Clause 5.2.3.12 specifies requirements for this component.

##### 5.2.3.1.2 XML component

The OptionalOutputs child element of VerifyResponse shall be an instance of OptionalOutputsVerifyType defined as in XML Schema file "[XSDFILESIGVALPROT]", whose location is detailed in clause A.1, and is copied below for information.

// MISSING SIGNING TIME OPTIONAL OTPUTXXXX.

<!—targetNamespace="http://uri.etsi.org/19442/v1.1.1#" -->

<xs:complexType name="OptionalOutputsVerifyType">

 <xs:sequence>

 <xs:element ref="ResultsForOneSignature" maxOccurs="unbounded"/>

</xs:sequence>

</xs:complexType>

<xs:element name="ResultsForOneSignature" type="OptionalOutputsVerifyType">

<xs:complexType name="OptionalOutputsVerifyType">

 <xs:complexContent>

 <xs:extension base="dss2:OptionalOutputsBaseType">

 <xs:sequence>

 <xs:element name="Result" type="dsb:ResultType" minOccurs="0" />// note for one signature

 <xs:element ref="SignatureReference" minOccurs="0" maxOccurs="1"/>// note for the fact that there may be some identifiers in the individual report. PL and myself should coordinate how to identify the signatures and use the same mechanisms everywhere.XXXX

 <xs:element ref="dss2:VerifyManifestResults" minOccurs="0" maxOccurs="1"/>

 <xs:element ref="dss2:SigningTimeInfo" minOccurs="0" maxOccurs="1"/>

 <xs:element ref="dss2:VerificationTimeInfo" minOccurs="0" maxOccurs="1"/>

 <xs:element ref="dss2:SignerIdentity" minOccurs="0" maxOccurs="1"/>

 <xs:element ref="AppliedSignatureValidationPolicy" minOccurs="0" />

 <xs:element ref="AvailableSignatureValidationPolicies" minOccurs="0" />

 <xs:element ref="dss2:VerifyManifestResults" minOccurs="0" maxOccurs="1"/>

 <xs:element name="IndividualReport" type="etsivr:IndividualReportType" minOccurs="0"/>

 <xs:choice minOccurs="0" />

 <xs:element ref="dss2:DocumentWithSignature" />

 <xs:element name="AugmentedSignature" type="dss2:SignatureObject" />

 </xs:choice>

 </xs:sequence>

 </xs:extension>

 </xs:complexContent>

</xs:complexType>

Any children element of instances of dss:OptionalOutputsBaseType type not mentioned and profiled in the present document may be present. Their semantics and syntax shall be as specified in [1].

Neither dss2:DocumentWithSignature nor AugmentedSignature elements shall be present in a validation response message.

// validation report shall be as in TS 119 102-2XXXXX

NOTE: The optional children elements dss2:DocumentWithSignature and AugmentedSignature are containers for augmented signatures. The protocol specified in this clause is the “validation protocol”, and consequently it does not manage augmented signatures. These elements will certainly be used within the “augmentation” protocol and the “validation and augmentation” protocol. These elements are present in the definition of the OptionalInputsVerifyType type for re-using this XML schema definition in the “validation and augmentation” protocol.

##### 5.2.3.1.3 JSON component

The oplOutp child element of dss-VerifyRequest shall be an instance of OptionalOutputsVerifyType defined as in JSONSchema file "[JSONSCHEMAFILESIGVALPROT]", whose location is detailed in clause A.2, and is copied below for information.

 "OptionalOutputsVerifyType":{

 "type": "object",

 "properties": {

 "resForOneSig":{

 "type": "array",

 "items": {

 "type": "object",

 "$ref": "#/definitions/ResultsForOneSignatureType"

 }

 }

 }

 }

 "ResultsForOneSignatureType":{

 "type": "object",

 "properties": {

 "profiles": {

 "type": "array",

 "items": {

 "type": "string"

 }

 },

 "policy": {

 "type": "array",

 "items": {

 "type": "string"

 }

 },

 "verificationTimeInfo": {

 "type": "object",

 "$ref": "<DSSXCORESCHEMAFILELOCATION>#/definitions/dss-VerificationTimeInfoType"

 },

 "signerIdentity": {

 "type": "object",

 "$ref": "<DSSXCORESCHEMAFILELOCATION>#/definitions/saml2-NameIDType"

 },

 "appliedSigValPolicy": {

 "type": "string"

 },

 "availableSigValPols":{

 "type": "array",

 "items": {

 "type": "string"

 }

 },

 "detailedValReport": {

 "type": "object",

 "$ref ": <ETSIJSONVALREPORTSCHEMAFILELOCATION>#/definitions/detailedValReport

 },

 "sigDetailedValReport": {

 "type": "object",

 "$ref": <RFC7515JSONSCHEMAFILE>#/definitions/JWS

 },

 "transformed": {

 "type": "array",

 "items": {

 "type": "object",

 "$ref": "<DSSXCORESCHEMAFILELOCATION>#/definitions/dss-TransformedDocumentType"

 }

 },

 "manifestValResult": {

 "type": "object",

 "$ref": "<DSSXCORESCHEMAFILELOCATION>#/definitions/dss-VerifyManifestResultsType"

 },

 "respID" : {

 "type": "string"

 }, ,

 "augmentedSig" : {

 "type": "object",

 "properties": {

 "augSig": {

 "type": "string"

 }

 "docWithSig":{

 type: "string"

 }

 }

 },

 "sigRef" : {

 "type": "object",

 "$ref": "<DSSXCORESCHEMAFILELOCATION>#/definitions/SignatureReferenceType"

 }

 }

 }

 }

The augmentedSig property shall not be present.

#### 5.2.3.2 Identifying the profile used

##### 5.2.3.2.1 Semantics

The response shall contain one component whose value shall be an identifier notifying that the response has been built using the profile defined by the present document.

The identifier for the profile defined by the present document shall be [http://uri.etsi.org/19442/v1.1.1/validationprofile#](http://uri.etsi.org/19442/v1.1.1/validationprofile),

The response may contain additional components whose values are identifiers of other profiles that have also been used for building the response.

##### 5.2.3.2.2 XML components

The response shall have one or more dss:AppliedProfile elements. dss:AppliedProfile element is specified in in clause [DSSXOPTIONALOUTPUTBASETYPE] of [1]. The value of each dss:AppliedProfile element shall be the identifier of one profile.

The value of the first dss:AppliedProfile child element in the response shall be the one specified in clause 5.2.3.2.1 of the present document.

##### 5.2.3.2.3 JSON component

The element that shall indicate the profiles used for building the response shall be the profiles element.

The array of strings shall have at least one item. The value of the first item shall be the one specified in clause 5.2.3.2.1 of the present document.

#### 5.2.3.3 Component for indicating the service policy

#### 5.2.3.3.1 Component semantics

This component shall contain a non-ambiguous identifier of the service policy under which the client requests the server to validate the signature.

##### 5.2.3.3.2 XML component

The element that shall identify under which service policy the validation has to be conducted shall be the dss:AppliedPolicy element specified in clause [OPTIONALOUTPUTSBASETYPE\_XML ] of [1].

##### 5.2.3.3.3 JSON component

The element that shall identify under which service policy the validation has to be conducted shall be the policy element. The contents of this element shall be as specified in clause [OPTIONALOUTPUTSBASETYPE\_JSON ] of [1]

#### 5.2.3. Component for indicating validation time

##### 5.2.3.4.1 Component semantics

This component shall provide means for indicating to the client the validation time set by the server, which may be the current time or a certain time in the past.

This component shall appear only in the response to requests of validation of XAdES signatures that incorporate the component requesting the server to set the validation time at a certain time instant, specified in clause 5.1.4.6.1 of the present document.

##### 5.2.3.4.2 XML component

The element that shall report the validation time used by the server shall be the dss:VerificationTimeInfo element specified in clause [VALIDATIONTIME\_INRESP 3.1.39.1] of [1].

##### 5.2.3.4.3 JSON component

The element that shall report the validation time used by the server shall be the element verificationTimeInfo an instance of dss-VerificationTimeInfoType type specified in clause [VALIDATIONTIME\_INRESP\_JSON 3.1.39.2] of [1].

#### 5.2.3. Component for returning signer’s identity

##### 5.2.3.5.1 Component semantics

This component of a response to a validation request shall return information on the signer’s identity.

This component shall appear only in the response to requests that incorporate the component requesting the server to return these details, specified in clause 5.1.4.9.1 of the present document.

##### 5.2.3.5.2 XML component

The element that shall contain the identity of the signer shall be the dss:SignerIdentit element, specified in clause [DhSSXIGNERIDENITY\_INRESP 5.5.7.1] of [1].

##### 5.2.3.5.3 JSON component

The element that shall contain the identity of the signer shall be the signerIdentity element, whose contents shall be as specified in clause [DSSXSIGNERIDENITY\_INRESP 5.5.7.2] of [1].

#### 5.2.3. Component for notifying the signature policy applied during the validation

##### 5.2.3.6.1 Component semantics

This component shall provide means for returning to the client the identifier of the signature validation policy applied by the server for validating the signature(s).

This component shall appear in the response to requests of validation of digital signatures that incorporate the component requesting the server to validate the signature(s) using a certain signature validation policy, specified in clause 5.1.4.10.1 of the present document, if the server has been able to check all the constraints defined in the signature validation policy identified in the present component, on all the validated signatures. It may also appear in responses to requests that do not incorporate the aforementioned component.

##### 5.2.3.6.2 XML component

The element that shall notify to the client the signature validation policy against which the signature has been validated shall be the AppliedSignatureValidationPolicy element.

The AppliedSignatureValidationPolicy element shall be defined as in XML Schema file "[XSDFILESIGVALPROT]", whose location is detailed in clause A.1, and is copied below for information.

<!—targetNamespace="http://uri.etsi.org/19442/v1.1.1#" 🡪

 <xs:element name="AppliedSignatureValidationPolicy" type="xs:anyURI"/>

The AppliedSignatureValidationPolicy element shall have as value the unique identifier of the signature validation policy used by the server for validating the signature as an URI. If the identifier of the signature validation policy is an OID, then the value of this element shall be an URN indicating the value of the aforementioned OID as specified in RFC 3061 [18].

##### 5.2.3.6.3 JSON component

The element that shall notify to the client the signature validation policy against which the signature has been validated shall be the appliedSigValPol.

The value of this element shall be the unique identifier of the signature validation policy used by the server for validating the signature as an URI. If the identifier of the signature validation policy is an OID, then the value of this element shall be an URN indicating the value of the aforementioned OID as specified in RFC 3061 [18]

#### 5.2.3. Component for notifying the signature policies under which the server can conduct validation

##### 5.2.3.7.1 Component semantics

This component shall provide means for returning to the client the identifiers of the signature validation policies under which the server can validate signatures.

This component shall appear only in the response to requests of validation of digital signatures that incorporate the component requesting the server to validate the signature(s) using a certain signature validation policy, specified in clause 5.1.4.10.1 of the present document.

This component shall only appear if the server is not able to validate the signature under the signature validation policy requested by the client in the aforementioned component of the validation request.

##### 5.2.3.7.2 XML component

The element that shall notify to the client the signature validation policies under which the server can conduct validation of digital signatures shall be the AvailableSignatureValidationPolicies element.

The AvailableSignatureValidationPolicies element shall be defined as in XML Schema file "[XSDFILESIGVALPROT]", whose location is detailed in clause A.1, and is copied below for information.

<!—targetNamespace="http://uri.etsi.org/19442/v1.1.1#" 🡪

 <xs:element name="AvailableSignatureValidationPolicies" type="AvailableSignatureValidationPoliciesType"/>

 <xs:complexType name="AvailableSignatureValidationPoliciesType">

 <xs:sequence>

 <xs:element name="AvailableSignatureValidationPolicyID" type="xs:anyURI" minOccurs="0" maxOccurs="unbounded"/>

 </xs:sequence>

 </xs:complexType>

Each AvailableSignatureValidationPolicyID child element shall have as value the unique identifier of one signature validation policy against which the server is able to validate digital signatures, as an URI. If the identifier of the signature validation policy is an OID, then the value of this element shall be an URN indicating the value of the aforementioned OID as specified in RFC 3061 [18].

##### 5.2.3.7.3 JSON component

In the JSON profile derived from OASIS specifications the element that shall notify to the client the signature validation policies under which the server can conduct validation of digital signatures shall be the availableSigValPols element.

Each item of the array shall have as value the unique identifier of one signature validation policy against which the server is able to validate digital signatures, as an URI. If the identifier of the signature validation policy is an OID, then the value of this element shall be an URN indicating the value of the aforementioned OID as specified in RFC 3061 [18].

#### 5.2.3. Component for returning the detailed validation report

##### 5.2.3.8.1 Component semantics

This component shall contain the detailed validation report for each digital signature that the server has validated in response to one request.

###### 5.2.3.8.1.1 ETSI XML protocol

The element that shall contain the detailed validation report for one digital signature shall be the IndividualReport, which is an instance of etsivr:IndividualReportType, as specified in [11].

###### 5.2.3.8.1.2 ETSI JSON protocol

The element that shall contain the detailed validation report for all the digital signatures shall be the detailedValReports. This element is an array of items. Each item shall contain an instance of <ETSITA110102-2>, as specified in [11].

// CHECK: NO JSON SYNTAX FOR VALIDATION REPORT. CHECK COHERENCE OF TYPE

#### 5.2.3. Component for returned the detailed validation report signed

##### 5.2.3.9.1 Component semantics

This component shall provide means for returning a signature within the validation report.

This component shall appear in the response to requests of validation of AdES signatures that incorporate the component requesting the server to sign the detailed validation report, specified in clause 5.1.4.12.1 of the present document.

This component may also appear in the response to request of validation of AdES signatures that does not incorporate the component requesting the server to sign the detailed validation report, specified in clause 5.1.4.12.1 of the present document.

##### 5.2.3.9.2 XML component

The element that shall return the signature over the validation report shall be a ds:Signature element generated by the server.

This ds:Signature element shall be enveloped within the IndividualReport element, which shall be an instance of etsivr:IndividualReportType, as specified in [11]

##### 5.2.3.9.3 JSON component

The element that shall return the signature over the validation report shall be sigDetailedValReport. This element shall contain an instance of a JSON Web Signature (JWS signature hereinafter) as specified in [13].

Each JWS shall envelop one individual validation report, generated by the server, which shall be an instance of <ETSITA110102-2\_JSON>, as specified in [11].

#### 5.2.3. Component for returning the result of transforming the input document

##### 5.2.3.10.1 Component semantics

This component shall provide means for returning to the client the result obtained by the server after applying a sequence of transformations to one input document.

This component shall appear only in the response to requests of validation of XAdES signatures that incorporate the component requesting the server to return one or more transformed input documents, specified in clause 5.1.4.13 of the present document.

##### 5.2.3.10.2 XML component

In the JSON profile derived from OASIS specifications the element transformed element shall be an array. Each item of the array shall contain the results obtained by the server after applying a sequence of transformations to one input document. Each item of the array shall be an instance of the the dss-TransformedDocumentType element specified in clause [DSSXTRANSFORMEDDOCUMENT\_INRESP 3.1.46.1] of [1].

##### 5.2.3.10.3 JSON component

The element that shall return the result obtained by the server after applying a sequence of transformations to one input document shall be an instance of dss-TransformedDocumentType type specified in clause [DSSTRANSFORMEDDOCUMENT\_INRESP\_JSON 3.1.46.2] of [1].

#### 5.2.3. Component for returning the result of validating ds:Manifest elemens in XAdES signatures

##### 5.2.3.11.1 Component semantics

This component shall not be present within a response to a request of validation of CAdES or PAdES signatures.

This component shall contain the result of the validation(s) performed by the server on signed the ds:Manifest elements present within a XAdES signature.

##### 5.2.3.11.2 XML component

The element that shall contain the result of the validation(s) performed by the server on signed the ds:Manifest elements present within the XAdES signature(s) shall be the dss:VerifyManifestResults element, specified in clause [VERIFYMANIFESTSRESULT\_INRESP 5.5.1.1] of [1].

##### 5.2.3.11.3 JSON component

The element that shall contain the i result of the validation(s) performed by the server on signed the ds:Manifest elements present within the XAdES signature(s) shall be the manifestValResult element. This element shall be an instance of dss-VerifyManifestResultsType type specified in clause [VERIFYMANIFESTSRESULT\_INRESP 5.5.1.2] of [1].

#### 5.2.3. Components for asynchronous processing

##### 5.2.3.12.1 Introduction

For managing asynchronous processing protocol detailed in clause 5.1.4.16.1, the components specified in clauses 5.2.3.12.2 and 5.2.3.12.3 are required:

##### 5.2.3.12.2 Component for indicating not completion of signature(s) validation

###### 5.2.3.12.2.1 Semantics

This component of a response to a validation request shall indicate that the server has not completed the validation of the signature(s) and that consequently the completion of the validation is pending.

###### 5.2.3.12.2.2 XML component

The element that shall indicate that the completion of the validation of the signature(s) is pending shall be shall be the dss:ResultMajor child element of VerifyResponse set to the value "urn:oasis:names:tc:dss:1.0:profiles:asynchronousprocessing:resultmajor:Pending", as specified in clause [VALIDATIONPENDING\_INRESP 5.2.1] of OASIS Standard: Digital Signature Service Core Protocols, Elements, and Bindings Version 2.0[1].

###### 5.2.3.12.2.3 JSON component

The element that shall indicate that the completion of the validation of the signature(s) is pending shall be shall be the maj child element of result child of dss-VerifyResponse set to the value "urn:oasis:names:tc:dss:1.0:profiles:asynchronousprocessing:resultmajor:Pending", as specified in clause [VALIDATIONPENDING\_INRESP 5.2.1] of OASIS Standard: Digital Signature Service Core Protocols, Elements, and Bindings Version 2.0 [1].

##### 5.2.3.12.3 Component for correlating subsequent requests to the initial response

###### 5.2.3.12.3.1 Semantics

This component of a response to a validation request shall contain an identifier for correlating subsequent pending-requests associated to a certain initial request.

###### 5.2.3.12.3.2 XML component

The component that shall contain an identifier for correlating subsequent pending-requests associated to a certain initial request shall be the dss2:ResponseID attribute as specified in clause [ASYNCORRELATINGID\_INRESP 5.2.2] of OASIS Standard: Digital Signature Service Core Protocols, Elements, and Bindings Version 2.0 [1].

###### 5.2.3.12.3.3 JSON component

The element that shall contain an identifier for correlating subsequent pending-requests associated to a certain initial request shall be respID whose value shall be as specified in clause [ASYNCORRELATINGID\_INRESP 5.2.2] of OASIS Standard: Digital Signature Service Core Protocols, Elements, and Bindings Version 2.0 [1].

#### 5.2.3.13 Component for returning signing time information

###### 5.2.3.13.1 Semantics

This component shall provide means for indicating to the client information on the signing time.

This component shall appear only in the response to requests of validation of XAdES signatures that incorporate the component requesting the server to return information of signing time, specified in clause 5.1.4.16.1 of the present document.

###### 5.2.3.13.2 XML component

The element that shall convey information on signing time shall be the dss:SigningTimeInfo element specified in clause [SIGNINGTIME\_INRESP 3.1.39.1] of [1].

###### 5.2.3.13.3 JSON component

The element that shall convey information on signing time shall be the element signingTimeInfo an instance of dss2-SigningTimeInfoType:SigningTimeBoundaries type specified in clause [SIGNINGTIME\_INRESP\_JSON 3.1.39.2] of [1].

# 6 Protocol for augmentation of AdES signatures.

## 6.1 Request message

### 6.1.1 Component for requesting augmentation of signatures

// add optional component for requesting a certain level of quality for time-stamp tokens in the augmentation. XXX

NO COMPONENT FOR AUGMENTATION POLICY IDENTIFIER

#### 6.1.1.1 Semantics

The signature augmentation request message shall allow requesting to the server the augmentation of one or more AdES signatures.

The request message shall be able to submit the signatures to be augmented, following the same principles as in the validation request message, either:

1. within the signature object container (if the signature is a non-embedded signature) or
2. within the input documents container (if the signature is an embedded signature) or
3. within an underlying protocol attachment (if the signature is an embedded signature), in which case this attachment shall be referenced within the input documents container.

The request message shall not include neither transformed documents neither digests of signed documents.

NOTE: In essence, an augmentation request message needs to submit only the AdES signatures to be augmented. However if these are embedded signatures, the message may submit the signed document(s) with the embedded signatures. Obviously, the server does not need the embedding documents during the augmentation process.

The request message shall allow to identify the signatures that the server is requested to augment.

The request message shall not be able to request augmenting different signatures to different levels: the server shall, consequently, try to augment all the identified signatures to the same level.

The request message shall allow the client to request the server to augment the signatures to a certain specific level.

#### 6.1.1.2 XML component

The element that shall be the component for requesting the augmentation of AdES signature(s) shall be the root element of the message AugmentRequest as specified in the present clause.

The AugmentRequest element shall be defined as in XML Schema file "[XSDFILESIGVALPROT]", whose location is detailed in clause A.1, and is copied below for information.

<!—targetNamespace="http://uri.etsi.org/19442/v1.1.1#" 🡪

 <xs:element name="AugmentRequest" type="AugmentRequestType"/>

 <xs:complexType name=="AugmentRequestType">

 <xs:extension base="dss2:RequestBaseType">

 <xs:sequence>

 <xs:element ref="dss2:InputDocuments" minOccurs="0"/>

 <xs:element ref="returnAugmentedSignature"/>// IS this needed? Not implicit in the AugmentRequest?

 <xs:element ref="ProcessSignatures" minOccurs="0"/>// Check name in the validation protocol

 <xs:element ref="dss2:SignatureObject" minOccurs="0"/>

 </xs:sequence>

 </xs:extension>

 </xs:complexType>

dss2:InputDocuments element shall only contain one or more dss2:Document children elements. These children elements shall contain one or more embedded AdES signatures or references to underlying protocol attachments where the documents with embedded AdES signatures.

The AugmentRequest element shall have one dss2:Profile child element, and it shall have the value [http://uri.etsi.org/19442/v1.1.1/augmentationprofile#](http://uri.etsi.org/19442/v1.1.1/validationprofile), identifying the request as an augmentation request compliant with the augmentation protocol specified in the present document.

The returnAugmentedSignature element shall indicate the level to which the client requests to augment the signatures.

The dss2:SignatureObject child element shall not contain any time-stamp token.

The ProcessSignatures element shall identify the signatures to be augmented. The requirements specified in clause 5.1.4.2.2.2 shall apply to this element.

#### 6.1.1.3 JSON component

The element that shall be the component for requesting the validation of AdES signature(s) shall be the root element of the message dss-AugmentRequest as specified in the present clause.

The dss-AugmentRequest element shall be defined as in JSON Schema file "[JSONSCHEMAFILESIGVALPROT], whose location is detailed in clause A.2, and is copied below for information.

"AugmentRequest": {

 "$xsd-type": "AugmentRequest",

 "type": "object",

 "properties": {

 "inDocs": {

 "type": "object",

 "$ref": "<DSSXCORESCHEMAFILELOCATION>#/definitions/dss2-InputDocumentsType"

 },

 "reqID": {

 "type": "string"

 },

 "profile": {

 "type": "array",

 "items": {

 "type": "string"

 }

 },

 "returnAugmentedSig": {

 "type": "string"

 },

 "processSigs": {

 "type": "object",

 "$ref": "#/definitions/SigsRefsType"

 },

 "sigObj": {

 "type": "object",

 "$ref": "<DSSXCORESCHEMAFILELOCATION>#/definitions/dss-SignatureObjectType"

 }

 }

 "required": ["returnAugmentedSig"]

 }

The profile array shall have one item, and it shall have the value [http://uri.etsi.org/19442/v1.1.1/augmentationprofile#](http://uri.etsi.org/19442/v1.1.1/validationprofile), identifying the request as an augmentation request compliant with this augmentation protocol.

#### 6.1.1.43 Processing model

If the request does not have the signatures-to-process-ids container then the server shall try to augment all the signatures present within the request message.

If the request does have the signatures-to-process-ids container then the server shall try to augment all the signatures referenced within the signatures-to-process-ids container.

### 6.1.2 Component for identifying the level the signatures are requested to be augmented to

#### 6.1.2.1 Component semantics

//URIS for CAdES with Evidence records XXX

This component shall have as value a URI reference identifying the pre-defined level to which the server is requested to augment the signature after its validation.

The value of this component shall take one of the values in Table 5 and Table 6 except the URIs identifying levels AdES-B-B, AdES-E-BES, AdES-E-EPES, AdES-B, AdES-BES, and AdES-EPES.

NOTE 1: The levels mentioned in the former paragraph are not the result of an augmentation operation, but the result of the actual generation of the AdES signature.

Table 5 lists the URIs for the levels specified for AdES signatures in ETSI EN 319 122, ETSI EN 319 132, and ETSI EN 319 142. Numbers in column Notes correspond to the numbers of the notes that follow the table.

Table : AdES signature levels in ETSI EN 319 1X2 and URIs

|  |  |  |
| --- | --- | --- |
| Signature level | URI | Notes |
| AdES-B-B | [http://www.etsi.org/ades/191x2/level/baseline/B-B#](http://www.etsi.org/ades/191x2/level/baseline/B-B) | 2 |
| AdES-B-T | [http://www.etsi.org/ades/191x2/level/baseline/B-T#](http://www.etsi.org/ades/191x2/level/baseline/B-T) | 2 |
| AdES-B-LT | [http://www.etsi.org/ades/191x2/level/baseline/B-LT#](http://www.etsi.org/ades/191x2/level/baseline/B-LT) | 2 |
| AdES-B-LTA | [http://www.etsi.org/ades/191x2/level/baseline/B-LTA#](http://www.etsi.org/ades/191x2/level/baseline/B-LTA) | 2 |
| AdES-E-BES | [http://www.etsi.org/ades/191x2/level/extended/E-BES#](http://www.etsi.org/ades/191x2/level/extended/E-BES) | 2 |
| AdES-E-EPES | [http://www.etsi.org/ades/191x2/level/extended/E-EPES#](http://www.etsi.org/ades/191x2/level/extended/E-EPES) | 2 |
| AdES-E-T | [http://www.etsi.org/ades/191x2/level/extended/E-T#](http://www.etsi.org/ades/191x2/level/extended/E-T) | 2 |
| AdES-E-C | [http://www.etsi.org/ades/191x2/level/extended/E-C#](http://www.etsi.org/ades/191x2/level/extended/E-C) | 3 |
| AdES-E-X | [http://www.etsi.org/ades/191x2/level/extended/E-X#](http://www.etsi.org/ades/191x2/level/extended/E-X) | 3 |
| AdES-E-X-Long | [http://www.etsi.org/ades/191x2/level/extended/E-X-Long#](http://www.etsi.org/ades/191x2/level/extended/E-X-Long) | 3 |
| AdES-E-X-L | [http://www.etsi.org/ades/191x2/level/extended/E-X-L#](http://www.etsi.org/ades/191x2/level/extended/E-X-L) | 3 |
| AdES-E-A | [http://www.etsi.org/ades/191x2/level/extended/E-A#](http://www.etsi.org/ades/191x2/level/extended/E-A) | 3 |
| AdES-E-LTV | [http://www.etsi.org/ades/191x2/level/extended/E-LTV#](http://www.etsi.org/ades/191x2/level/extended/E-LTV) | 4 |

NOTE 2: The levels identified in these rows are levels that CAdES, PAdES, and XAdES signatures can reach.

NOTE 3: The levels identified in these rows are levels that CAdES and XAdES signatures can reach, but not PAdES signatures.

NOTE 4: The levels identified in these rows are levels that only PAdES signatures can reach.

Table 6 lists the URIs for the levels specified for AdES signatures in the different ETSI TSs, namely ETSI TS 101 733, ETSI TS 102 778, ETSI TS 101 903, ETSI TS 103 171, ETSI TS 103 172, and ETSI TS 103 173. Numbers in column Notes correspond to the numbers of the notes that follow the table.

Table : AdES signature levels in ETSI EN 319 1X2 and URIs

|  |  |  |
| --- | --- | --- |
| Signature level | URI | Notes |
| AdES-B | [http://www.etsi.org/ades/etsits/level/baseline/B-B#](http://www.etsi.org/ades/etsits/level/baseline/B-B) | 5 |
| AdES-T | [http://www.etsi.org/ades/etsits/level/baseline/B-T#](http://www.etsi.org/ades/etsits/level/baseline/B-T) | 5 |
| AdES-LT | [http://www.etsi.org/ades/etsits/level/baseline/B-LT#](http://www.etsi.org/ades/etsits/level/baseline/B-LT) | 5 |
| AdES-LTA | [http://www.etsi.org/ades/etsits/level/baseline/B-LTA#](http://www.etsi.org/ades/etsits/level/baseline/B-LTA) | 5 |
| AdES-BES | [http://www.etsi.org/ades/etsits/level/BES#](http://www.etsi.org/ades/etsits/level/BES) | 5 |
| AdES-EPES | [http://www.etsi.org/ades/etsits/level/EPES#](http://www.etsi.org/ades/etsits/level/EPES) | 5 |
| AdES-T | [http://www.etsi.org/ades/etsits/level/T#](http://www.etsi.org/ades/etsits/level/T) | 5 |
| AdES-C | [http://www.etsi.org/ades/etsits/level/C#](http://www.etsi.org/ades/etsits/level/C) | 6 |
| AdES-X | [http://www.etsi.org/ades/etsits/level/X#](http://www.etsi.org/ades/etsits/level/X) | 6 |
| AdES-X-Long | [http://www.etsi.org/ades/etsits/level/X-Long#](http://www.etsi.org/ades/etsits/level/X-Long) | 6 |
| AdES-X-L | [http://www.etsi.org/ades/etsits/level/X-L#](http://www.etsi.org/ades/etsits/level/X-L) | 6 |
| AdES-A | [http://www.etsi.org/ades/etsits/level/A#](http://www.etsi.org/ades/etsits/level/A) | 6 |
| AdES-LTV | [http://www.etsi.org/ades/etsits/level/LTV#](http://www.etsi.org/ades/etsits/level/LTV) | 7 |

NOTE 5: The levels identified in these rows are levels that CAdES, PAdES, and XAdES signatures can reach.

NOTE 6: The levels identified in these rows are levels that CAdES and XAdES signatures can reach, but not PAdES signatures.

NOTE 7: The levels identified in these rows are levels that only PAdES signatures can reach.

#### 6.1.2.2 XML component

The element that shall request the augmenting of the signature after its validation shall be the ReturnAugmentedSignature element.

The ReturnAugmentedSignature element shall be defined as in XML Schema file "[XSDFILESIGVALPROT]", whose location is detailed in clause A.1, and is copied below for information.

<!—targetNamespace="http://uri.etsi.org/19442/v1.1.1#" 🡪

 <xs:element name="ReturnAugmentedSignature" type="ReturnAugmentedSignatureType"/>

 <xs:complexType name="ReturnAugmentedSignatureType">

 <xs:attribute name="Level" type="xs:anyURI" use="required" />

 </xs:complexType>

 The Level attribute of the aforementioned element shall have the value indicating the level to which the server is requested to augment the signatures.

**Processing model.**

If the server manages to augment the signatures, it shall return each augmented signature within an AugmentedSignature element or within the dss:DocumentWithSignature within the response message as specified in clause 6.2.1.2.

#### 6.1.2.3 JSON component

The element that shall request the augmenting of the signature after its validation shall be the returnAugmentedSig component.

The aforementioned component shall take one of the values in Table 5 and Table 6 except the URIs identifying levels AdES-B-B, AdES-E-BES, AdES-E-EPES, AdES-B, AdES-BES, and AdES-EPES.

The aforementioned component shall have the value indicating the level to which the server is requested to augment the signatures.

**Processing model.**

If the server manages to augment the signatures, it shall return each augmented signature within the augmentedSig element or within the docWithSignature as specified in clause 6.2.1.2.

## 6.2 Response message

### 6.2.1 Component for responding to augmentation request

#### 6.2.1.1 Component semantics

The signature augmentation response message shall allow to the server return to the client one or more AdES, which shall be the result of augmenting the AdES signatures submitted by the client to the requested level.

The signature augmentation response message shall have one or more signature result containers for returning either:

1. the augmented signature or
2. an indication of error while trying to augment the signature and a reference to the non augmented signature.

The response message shall be able to return the augmented signatures either:

1. within the signature object container (if the signature is a non-embedded signature) or
2. within the document with signature container (if the signature is an embedded signature) or
3. within an underlying protocol attachment (if the signature is an embedded signature), in which case this attachment shall be referenced within the document with signature container.

NOTE: In essence, an augmentation response message needs to submit only the augmented AdES signatures. However if these are embedded signatures, the message may submit the signed document(s) with the embedded signatures.

#### 6.2.1.2 XML component

The element that shall be the component for responding to a request of augmentation of AdES signature(s) shall be the root element of the message AugmentResponse as specified in the present clause.

The AugmentResponse element shall be defined as in XML Schema file "[XSDFILESIGVALPROT]", whose location is detailed in clause A.1, and is copied below for information.

<!—targetNamespace="http://uri.etsi.org/19442/v1.1.1#" 🡪

 <xs:element name="AugmentResponse" type="AugmentResponseType"/>

 <xs:complexType name="AugmentResponseType">

 <xs:extension base="dss2:ResponseBaseType">

 <xs:sequence>

 <xs:element ref="AugmentSignatureResult" maxOccurs="unbounded"/>

 </xs:sequence>

 </xs:extension>

 </xs:complexType>

The AugmentResponse element shall have one dss2:AppliedProfile child element, and it shall have the value [http://uri.etsi.org/19442/v1.1.1/augmentationprofile#](http://uri.etsi.org/19442/v1.1.1/validationprofile), identifying the request as an augmentation request compliant with the augmentation protocol specified in the present document.

The AugmentSignatureResult element shall contain the details of the process performed by the server for augmenting one AdES signature present in the request.

#### 6.2.1.3 JSON component

The element that shall be the component for responding to the validation request of AdES signature(s) shall be the root element of the message AugmentResponse as specified in the present clause.

The AugmentResponse element shall be defined as in JSON Schema file "[JSONSCHEMAFILESIGVALPROT], whose location is detailed in clause A.2, and is copied below for information.

 "AugmentResponse": {

 "$xsd-type": "AugmentResponse",

 "type": "object",

 "properties": {

 "result": {

 "type": "object",

 "$ref": "<DSSXCORESCHEMAFILELOCATION>#/definitions/dss-ResultType"

 },

 "reqID": {

 "type": "string"

 },

 "profile": {

 "type": "array",

 "items": {

 "type": "string"

 }

 },

 "augmentSigsResults": {

 "type": "array",

 "items": {

 "type": "object",

 "$ref": "#/definitions/AugmentSigResultType"

 }

 }

 }

 "required": ["result"]

 }

The profile array shall have one item, whose value shall be [http://uri.etsi.org/19442/v1.1.1/validationprofile#](http://uri.etsi.org/19442/v1.1.1/validationprofile), identifying the response as a validation response compliant with the validation profile specified in the present document.

### 6.2.2 Component for the global augmentation result

#### 6.2.2.1 Component semantics

This component shall contain a major result, which shall report whether the protocol has been successfully executed, regardless all, or only some, or none of the signatures whose augmentation was requested, have actually been augmented or not.

NOTE: "The protocol has been successfully executed" means that the request was fully correct, that the server correctly understood it, and that the server tried to augment all the signatures identified in the request. But it does not mean that the server has been able to successfully augment all of them. This will be indicated by the individual reports present within each AugmentSignatureResult component.

This component may also contain a minor result providing additional information on the task performed by the server.

#### 6.2.2.2 Processing model

The result major child of this component shall have the semantics of component Result as specified in OASIS Standard: Digital Signature Service Core Protocols, Elements, and Bindings Version 2.0 [1].

In addition to that, if the result major child indicates that the server has successfully processed the request, then the result minor child shall have the following value:

http://uri.etsi.org/19442/v1.1.1#CheckIndividualResults

This value shall indicate that there shall be as many signature result containers as signatures the client requested to augment. It shall also indicate that each signature result container shall contain a result component providing details of the result of trying to augment that signature, and that if the augmentation succeeded, the augmented signature shall be present.

#### 6.2.2.2 XML component

The element that within the response shall notify the validation result shall be the dsb:Result element specified in clause [RESULT 3.1.12.1] of [1].

#### 6.2.2.3 JSON component

The element that within the response shall notify the validation result shall be an instance of the ResultType type specified in clause [RESULT\_JSON 3.1.12.2] of [1].

### 6.2.3 Signature result container

#### 6.2.2.1 Semantics

The requirements of clause shall apply for this component.

In addition, if the server has succeeded in the validation and augmentation of the signature corresponding to this component, this component:

1. Shall have either:
* one document with signature container if the augmented signature is an embedded signature or
* one augmented signature container if the augmented signature is non-embedded.
1. Shall have a result component with its major result child set to a value indicating that the process has succeeded and with its minor result child set to [http://uri.etsi.org/19442/v1.1.1 - validationAndAugmentationSuccess](http://uri.etsi.org/19442/v1.1.1#validationAndAugmentationSuccess)

If the server has not succeeded in the augmentation of the signature this component:

1. Shall have a reference to the signature that the client request to validate and augment in the augmentation request.
2. Shall not contain neither the document with signature container or the augmented signature container.
3. Shall have a result component with its major result child set to <http://uri.etsi.org/19442/v1.1.1#augmentationFailure>. The result component shall not have a minor result child.

 In addition to that, this component shall include a result component whose contents shall be as specified in table below.

##### 6.2.2.1.2 XML component

The component for reporting on the augmentation of one AdES signature shall be an item of an insthe element AugmentSignatureResult as specified in the present clause.

The AugmentSignatureResult element shall be defined as in XML Schema file "[XSDFILESIGVALPROT]", whose location is detailed in clause A.1, and is copied below for information.

<!—targetNamespace="http://uri.etsi.org/19442/v1.1.1#" 🡪

 <xs:element name="AugmentSignatureResult" type="AugmentSignatureResultType"/>

 <xs:complexType name="AugmentSignatureResultType">

 <xs:sequence>

 <xs:element ref="dss2:Result" />

 <xs:element ref="SignatureReference" minOccurs="0" maxOccurs="1"/>

 <xs:choice minOccurs="0" />

 <xs:element ref="dss2:DocumentWithSignature" maxOccurs="unbounded"/>

 <xs:element name="AugmentedSignature" type="dss2:SignatureObject" />

 </xs:choice>

 </xs:sequence>

 </xs:complexType>

##### 6.2.2.1.3 JSON component

The component for reporting on the augmentation of one AdES signature shall be each of the items of augmentSigsResults array. Each item shall be an instance of AugmentSigResultType type.

The AugmentSigResultType typeshall be defined as in JSON Schema file "[JSONSCHEMAFILESIGVALPROT]", whose location is detailed in clause A.1, and is copied below for information.

 "AugmentSigResultType": {

 "$xsd-type": "AugmentSignatureResultType",

 "type": "object",

 "properties": {

 "result": {

 "type": "object",

 "$ref": "<DSSXCORESCHEMAFILELOCATION>#/definitions/dss-ResultType"

 },

 "docWithSignature": {

 "type": "object",

 "$ref": "<DSSXCORESCHEMAFILELOCATION>#/definitions/dss2-DocWithSignatureType"

 },

 "augmentedSig": {

 "type": "object",

 "$ref": "<DSSXCORESCHEMAFILELOCATION>#/definitions/dss2-SignatureObjectType"

 },

 "sigRefs": {

 "type": "array",

 "items": {

 "type": "object",

 "$ref": "#/definitions/SigsRefsType"

 }

 }

 }

 "required": ["result"]

 }

# 7 Protocol for validation and augmentation of AdES signatures.

## 7.1 Request message

### 7.1.1 Component for requesting validation and augmentation

#### 7.1.1.1 Component semantics

The requirements for this component shall be the requirements specified in clauses 5.1.1.1 except for the following difference:

1. This component shall have a profile identifier that shall identify the message as compliant with the “validation and augmentation” protocol specified in the present document.

#### 7.1.1.2 XML component

The element that shall be the component for requesting the validation and augmentation of AdES signature(s) shall be the root element of the message VerifyRequest as specified in 5.1.1.2 with the following difference:

1. the VerifyRequest element shall have one dss2:Profile child element, and it shall have the value [http://uri.etsi.org/19442/v1.1.1/validationAndAugmentationprofile#](http://uri.etsi.org/19442/v1.1.1/validationAndAugmentationprofile), identifying the request as a validation and augmentation request compliant with the validation and augmentation profile specified in the present document.

#### 7.1.1.3 JSON component

The element that shall be the component for requesting the validation and augmentation of AdES signature(s) shall be the root element of the message dss-VerifyRequest as specified in 5.1.1.3 with the following difference:

1. the profile array shall have one item, and it shall have the value <http://uri.etsi.org/19442/v1.1.1/validationAndAugmentationprofile>, identifying the request as a validation and augmentation request compliant with the validation and augmentation profile specified in the present document.

### 7.1.2 Components for submitting signatures and signed documents

The components for submitting signatures and signed documents shall be the same as the components used for submitting signatures and signed documents in the request messages of the validation protocol.

Requirements in clause 5.1.2.1 apply for submitting the signatures to be validated and augmented. The XML elements used for submitting the signatures shall be the ones specified in clause 5.1.2.2. The JSON elements for submitting the signatures shall be the ones specified in clause 5.1.2.3.

Requirements in clause 5.1.3.1 apply for submitting the documents signed by the signatures. The XML elements used for submitting the documents shall be the ones specified in clause 5.1.3.2. The JSON elements used for submitting the documents shall be the ones specified in clause 5.1.3.3.

### 7.1.3 Optional components

#### 7.1.3.1 Container for optional components

##### 7.1.3.1.1 Semantics

The requirements in clause 5.1.4.1.1 apply to this component, except for the following difference:

1. This component shall include a component for requesting the augmentation of the signatures to a certain level.

NOTE: Despite the fact that this component is mandatory for the validation and augmentation protocol, it appears within the group of optional inputs, which are not optional any more, for keeping as much alignment with DSS-X OASIS protocol, which does not make the separation between “validation” protocol and “validation and augmentation” protocol.

##### 7.1.3.1.2 XML component

The container for the optional components shall be an instance of the OptionalInputsVerifyType type specified in clause 5.1.4.1.2.

All the requirements of clause 5.1.4.1.2 shall apply in the validation and augmentation request message with the following difference:

1. the ReturnAugmentedSignature element shall be present.

##### 7.1.3.1.2 JSON component

The container for the optional components shall be an instance of the OptionalInputsVerifyType type specified in clause 5.1.4.1.3.

All the requirements of clause 5.1.4.1.3 shall apply in the validation and augmentation request message with the following difference:

1. the returnAugmentedSig element shall be present.

### 7.1.4 Processing model

When the server receives an “validation and augmentation” request message, it:

1. shall proceed to validate the signatures that the client requests to be validated, as a server serving a “validation” request message would do,
2. shall proceed to augment the signatures that have been validated to the level indicated in the ReturnAugmentedSignature element.
3. shall generate one signature result container for each signature that the client requested to validate and augment. A signature result container shall include the components directly related to the signature validation process as specified in clause 5.2. In addition to that, if the server has managed to successfully augment the signature, the signature result container shall include a signature object container with the augmented signature; otherwise, it shall notify an error in the augmenting process.

## 7.2 Response message

### 7.2.1 Component for responding to validation and augmentation request

#### 7.2.1.1 Component semantics

The requirements for this component shall be the requirements specified in clause 5.2.1.1, except by the following differences:

1. This component shall have a profile identifier that shall identify the message as compliant with the “validation and augmentation” protocol specified in the present document.

#### 7.2.1.2 XML component

The element that shall be the component for responding to a request of validation and augmentation of AdES signature(s) shall be the root element of the message VerifyResponse as specified in clause 5.2.1.2 with the following difference:

1. the VerifyRequest element shall have one dss2:Profile child element, and it shall have the value [http://uri.etsi.org/19442/v1.1.1/validationAndAugmentationprofile#](http://uri.etsi.org/19442/v1.1.1/validationAndAugmentationprofile), identifying the response as a validation and augmentation response compliant with the augmentation and validation profile specified in the present document.

#### 7.2.1.3 JSON component

The element that shall be the component for responding to a request of validation and augmentation of AdES signature(s) shall be an instance of VerifyResponse as specified in 5.2.1.3 with the following difference:

1. the profile array shall have one item, and it shall have the value <http://uri.etsi.org/19442/v1.1.1/validationAndAugmentationprofile>, identifying the response as a validation and augmentation response compliant with the validation and augmentation profile specified in the present document.

### 7.2.2 Component for the global validation and augmentation result

#### 7.2.2.1 Semantics

This component shall contain a major result, which shall report whether the server has been able to perform its task, regardless the results obtained. This component may also contain a minor result providing additional information on the task performed by the server.

#### 7.2.2.2 Processing model

The result major child of this component shall have the semantics of component Result as specified in OASIS Standard: Digital Signature Service Core Protocols, Elements, and Bindings Version 2.0 [1].

If the result major child indicates that the server has successfully processed the request, then the result minor child shall have the following value:

http://uri.etsi.org/19442/v1.1.1#CheckIndividualResults

This value shall indicate that there shall be as many signature result containers as signatures the client requested to validate and augment. It shall also indicate that each signature result container shall contain a result component providing details of the result of trying to validate and augment that signature, and that if the augmentation succeeded, the augmented signature shall be present.

#### 7.2.2.3 XML component

The element that shall notify the global validation and augmentation result shall be the dsb:Result element specified in clause [RESULT 3.1.12.1] of [1].

#### 7.2.2.4 JSON component

The element that shall notify the global validation and augmentation result shall be an instance of the ResultType type specified in clause [RESULT\_JSON 3.1.12.2] of [1].

### 7.2.3 Optional components

#### 7.2.3.1 Container for optional components

##### 7.2.3.1.1 Semantics

The requirements in clause 5.2.3.1.1 apply to the component consisting in a sequence of signature results container..

##### 7.2.3.1.2 XML component

The element for incorporating the sequence of signature result containers shall be an instance of OptionalOutputsVerifyType type specified in clause 5.2.3.1.2. The requirements specified in clause 5.2.3.1.2 shall apply for this element.

1. If the augmentation of a signature, which is reported within a certain ResultsForOneSignature element, has succeeded, this ResultsForOneSignature element shall include the augmented signature. If the signature is embedded then the signature and its embedding document shall be present within the dss2:DocumentWithSignature child element. Otherwise the augmented signature shall be present within the AugmentedSignature child element.

##### 7.2.3.1.3 JSON component

The element for incorporating the sequence of signature result containers shall be the oplOutp property, instance of OptionalOutputsVerifyType type as specified in clause 5.2.3.1.3. The requirements specified in clause 5.2.3.1.3 shall apply for this element:

1. If the augmentation of a signature, which is reported within a certain item of resForOneSig array, has succeeded, this item of the array shall include the augmentedSig property for incorporating the augmented signature. If the signature is embedded then the signature and its embedding document shall be incorporated within the docWithSig property of augmentedSig. Otherwise the augmented signature shall be incorporated within augSig property of augmentedSig.

#### 7.2.3.2 Signature result container

##### 7.2.3.2.1 Semantics

The requirements in clause 5.2.3.1.1 shall apply to this component with the following difference.

1. If the augmentation of the signature reported within the signature result container has succeeded, the signature result container shall include a component for returning the augmented signature to the client.

##### 7.2.3.2.2 XML component

The element for implementing the signature results container shall be the ResultsForOneSignature element as specified in clause 5.2.3.1.2. The requirements specified in clause 5.2.3.1.2 shall apply for this element with the following difference:

1. If the augmentation of a signature, which is reported within a certain ResultsForOneSignature element, has succeeded, this ResultsForOneSignature element shall include the augmented signature. If the signature is embedded then the signature and its embedding document shall be present within the dss2:DocumentWithSignature child element. Otherwise the augmented signature shall be present within the AugmentedSignature child element.

##### 7.2.3.2.3 JSON component

The element for implementing the signature results container shall be the resForOneSig property as specified in clause 5.2.3.1.3. The requirements specified in clause 5.2.3.1.3 shall apply for this element with the following difference:

1. If the augmentation of a signature, which is reported within a certain item of resForOneSig array, has succeeded, this item of the array shall include the augmentedSig property for incorporating the augmented signature. If the signature is embedded then the signature and its embedding document shall be incorporated within the docWithSig property of augmentedSig. Otherwise the augmented signature shall be incorporated within augSig property of augmentedSig.

Annex A (normative):
XML and JSON Schema files

# A.1 XML Schema file location for namespace [http://uri.etsi.org/19442/v1.1.1#](http://uri.etsi.org/19442/v1.1.1)

The file at [XSDFILESIGVALPROT\_URL] ([XSDFILESIGVALPROT]) contains the definitions of elements and types defined within the namespace whose URI value is [http://uri.etsi.org/19442/v1.1.1#](http://uri.etsi.org/19442/v1.1.1).

# A.2 JSON Schema file location for “$schema” "http://etsi.org/119442/v1.1.1/json#"

The file at [JSONFILESIGVALPROT\_URL] ([JSONSCHEMAFILESIGVALPROT]) contains the definitions of elements and types defined within the JSON schema whose "$schema" value is "http://etsi.org/119442/v1.1.1/json#"

Annex (informative):
Change History

| Date | Version | Information about changes |
| --- | --- | --- |
| <Month year> | <#> | <Changes made are listed in this cell> |

# History

|  |
| --- |
| **Document history** |
| V0.0.3 | September 2017 | For ESI#60 |
| V0.0.4 | November 2017 | Further updates for ESI review before public availability (same content as v0.0.3f) |
| V0.0.5 | November 2017 | Stable draft for public review |

*Latest changes made on 2016-05-20*