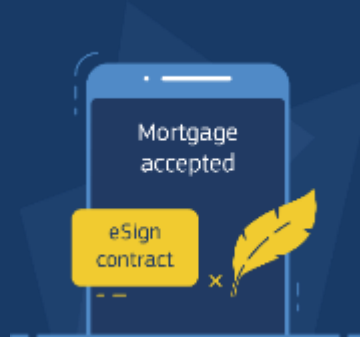




DIGITAL eSignature

***The Commission's actions on
international compatibility of
trust services***

*March 27th, 2023 - 13:30 - 17:00
(CET)*



Instructions for this event:



Please note that everyone is **muted by default**



Please note that your **camera is off by default** for privacy and bandwidth reasons



Dedicated Q&A slots are foreseen, or you can ask questions via the chat.



Agenda

Click on the topics to go straight to the section

| ID | TIME (CET) | TOPIC | SPEAKERS |
|-------|-------------|--|-------------------------|
| 1 | 13:30-13:40 | <u>Opening session</u> | Yi Qi HO |
| 2 | 13:40-13:55 | <u>Recognition of trust services under eIDAS: Policy updates</u> | Vicente ANDREU NAVARRO |
| 3 | 13:55-14:10 | <u>Actions undertaken by the Commission</u> | Apostolos Tolis APLADAS |
| 4 | 14:10-14:35 | <u>Presentation of the path to mutual recognition</u> | Olivier DELOS |
| ----- | | | |
| 5 | 14:45-15:10 | <u>Overview of the technical specifications part of the Commission pilot</u> | Olivier BARETTE |
| 6 | 15:10-15:25 | <u>Practical aspects: support of MRA element in DSS</u> | Aleksandr BELIAKOV |
| 7 | 15:25-15:55 | <u>Introduction to DSS</u> | Aleksandr BELIAKOV |
| ----- | | | |
| 8 | 16:05-16:50 | <u>Specifications of the new DSS version</u> | Aleksandr BELIAKOV |
| 9 | 16:50-17:00 | <u>Closing remarks</u> | Apostolos Tolis APLADAS |



1

Opening session

Yi Qi HO



*Back to the table
of contents*

Objectives of this information session

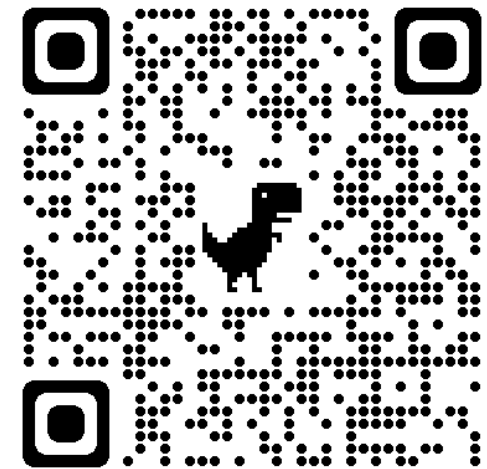
During this online information session, you can explore **how to use the new specifications** in the context of the European Commission's initiatives and discover how the implementation of the Mutual Recognition Agreement (MRA) works between the EU and a Third Country.

NEXT EVENT

EC-3rd Countries Trust Services Forum

June 12th, 2023 - 09:00 - 17:30 (CET)
Hybrid event, Brussels

Sign up by scanning the QR code below to receive updates about future milestones or upcoming events



Meet your speakers for today



Vicente ANDREU NAVARRO
Policy Officer, DG CNECT



Apostolos (Tolis) APLADAS
Project Officer, DG DIGIT



Olivier DELOS
eSignature Expert



Olivier BARETTE
eSignature Technical Lead



Aleksandr BELIAKOV
Digital Signature Service
developer




Yi Qi HO
eSignature Stakeholders Manager

2

Recognition of trust services under eIDAS: Policy updates

Vicente ANDREU NAVARRO



*Back to the table
of contents*

International cooperation and eIDAS



International cooperation and eIDAS

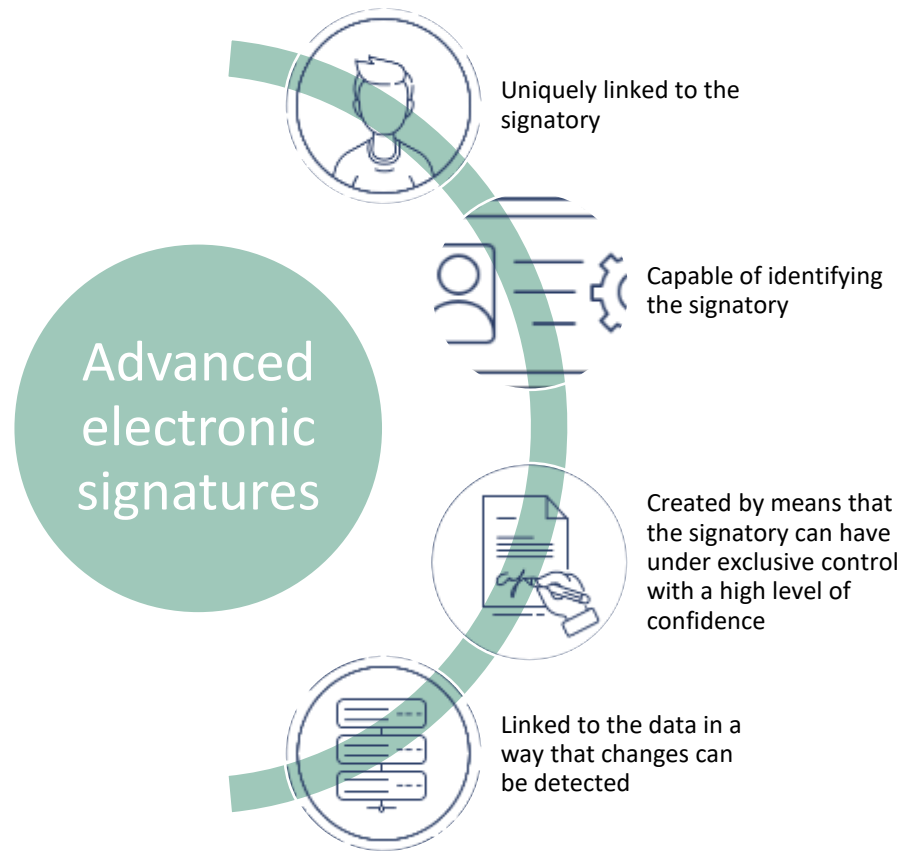
- Activities in this area are currently established at **three different levels**:
 - Association Agreements between third countries and the EU
 - Dialogues and information exchanges (formal and informal)
 - Pilots and proofs of concept
 - Participation in international initiatives:
 - UN's UNCITRAL model law
 - OECD's Guidelines for the governance of Digital Identities
- And very recently, validation tools for certain kinds of third countries' electronic signatures by EU member states (**TC AdES LOTL**).

International cooperation and eIDAS

- The possibilities for cooperation are **limited by the regulation** itself:
 - Mutual recognition of Qualified Trust Services is possible under article 14 of eIDAS
 - Mutual recognition of electronic identities is not considered in eIDAS (although, being an exclusive competence of the EU could be the object of international agreements under art. 218 TFEU)
- Mutual recognition of QTS under article 14 of eIDAS has never been implemented, the process is complex.
- The **proposal for a new eIDAS regulation modifies article 14** in order to make the process more straightforward (adding the possibility of achieving the same goal via implementing acts)

Recognition of TC electronic signatures

Only QES have the equivalent legal effect of handwritten signature in the EU, but...



...legal effects of electronic signatures cannot be denied solely on the grounds that they are in electronic form or that they are not qualified.

Recognition of TC electronic signatures

- Process triggered by the invasion of Ukraine and the need to validate Ukrainian electronic signatures in EU member states
- Set of tools that facilitate compliance with eIDAS
- Imposes no obligations to member states beyond what was already established in eIDAS (undeniability, in principle, of legal effects of electronic signatures)
- Formal checks are performed by the COM on TC's electronic signatures that offer a sufficient level of trust based on the approximation to EU regulation and standards

Inclusion in the TC AdES LOTL

- No need for an international agreement as it is not mutual recognition of qualified electronic signatures
- Formal request by the TC to DG CNECT
- Technical assessment of the legal and technical aspects of TC electronic signatures (they must be equivalent/similar to EU QES under the TC's regulations).
- Technical works addressed to include the pointers to TC LOTL in the EU TC AdES LOTL.

Effects

- Validation of TC's electronic signatures equivalent to EU QES becomes an easy task
- Although they cannot be considered as EU QES, the EU TC AdES LOTL offers the added value of the technical assessment by the COM of the electronic signatures generated in the TC
- First step towards future mutual recognition of qualified trust services

3

Actions undertaken by the Commission

Apostolos Tolis APLADAS

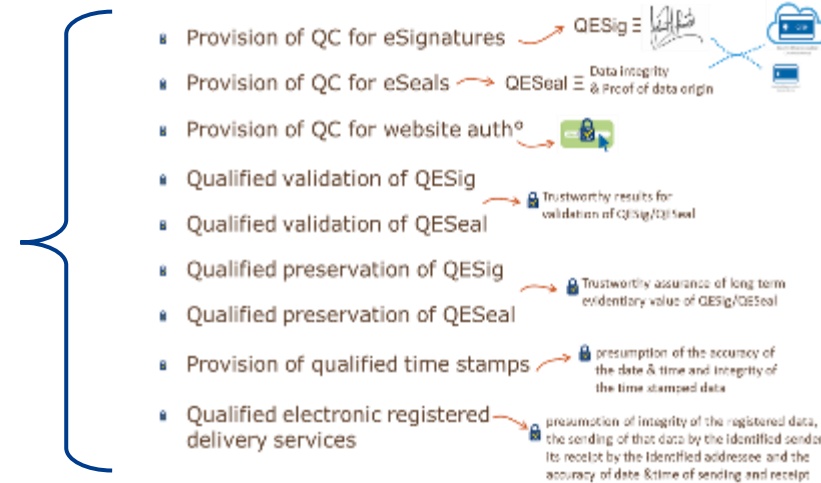


*Back to the table
of contents*

Article 14 of eIDAS Regulation

- Recognition of 3rd country TSP/TS as **legally equivalent** to EU QTSP/QTS

- **Closed** list of **9 types** of EU QTSP/QTS

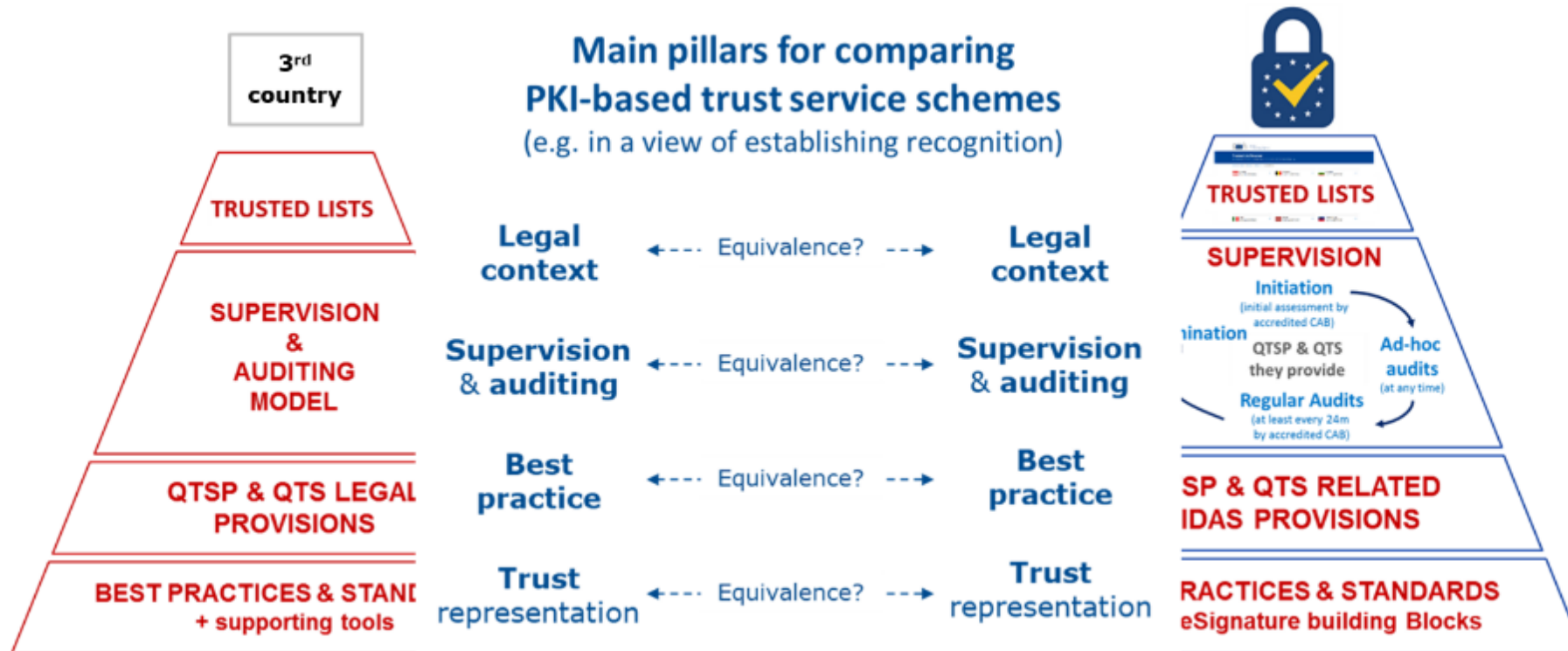


- Under an **agreement** concluded between the Union and the 3rd country or an international organisation in accordance with Article 218 TFEU
- 3rd country TSP/TS must **meet the eIDAS requirements** applicable to EU QTSP/QTS
- **Reciprocity** of the legal equivalence of EU QTSP/QTS in 3rd country or international organization

3rd country preparing to MRA under Art.14 of eIDAS

- Expected to make use of the **MRA cookbook** documentation and of the technical pilot facilities
- Allowing them to:
 - **Assess** the readiness and compliance of the 3rd country TSP/TS framework for mutual recognition with EU QTSP/QTS on four relevant pillars:
 - **Legal** framework
 - **Supervisory** framework
 - **Technical** standards and best practices
 - **Trusted list** representation of TSP/TS approval
 - Better understand the **technical implications/assumptions** for the implementation of an eIDAS MRA
 - Prepare the establishment of **suitable implementation** of 3rd country **TL** and **pointer** to EU LOTL to express a future formal MRA
 - Set-up appropriate **validation tools** for “cross-validation” of equivalent (qualified) trust services and their outputs when MRA is implemented
 - Be better prepared for **engaging a mutual recognition process** and assessment on the way towards an MRA

Main pillars for comparing PKI – based trust service schemes (e.g. in a view of establishing recognition)



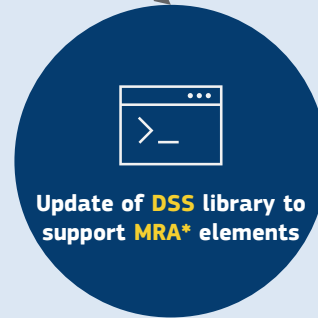
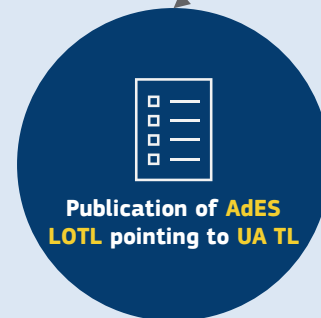
Collaboration with Ukraine (UA)

Context

Formal request
received from the
Ukrainian
Government to:



**Technical
implementation**
by EC and UA:



Supported by **eIDAS Art. 27(1)**
To be adopted by MSs on a **voluntary basis**

Recognition of UA-QES as eIDAS AdES

Legal context – eIDAS Regulation **Chapter III on Trust Services**

Article 27(1) on “Electronic signatures in public services” states:

*“If a Member State requires an **advanced electronic signature** to use an online service offered by [...] a public sector body, that Member State shall*

recognize advanced electronic signatures, [...]

*in at least **the formats or using methods defined in the implementing acts [...].**”*

X Must not be confused with Article 14(1) on “International aspects”:

*“Trust services provided by trust service providers established in a third country shall be recognised **as legally equivalent to qualified trust services** [...] where the trust services originating from the third country are recognised under **an agreement concluded** between the Union and the third country [...] in accordance **with Article 218 TFEU.**”*

Recognition of UA-QES as eIDAS AdES

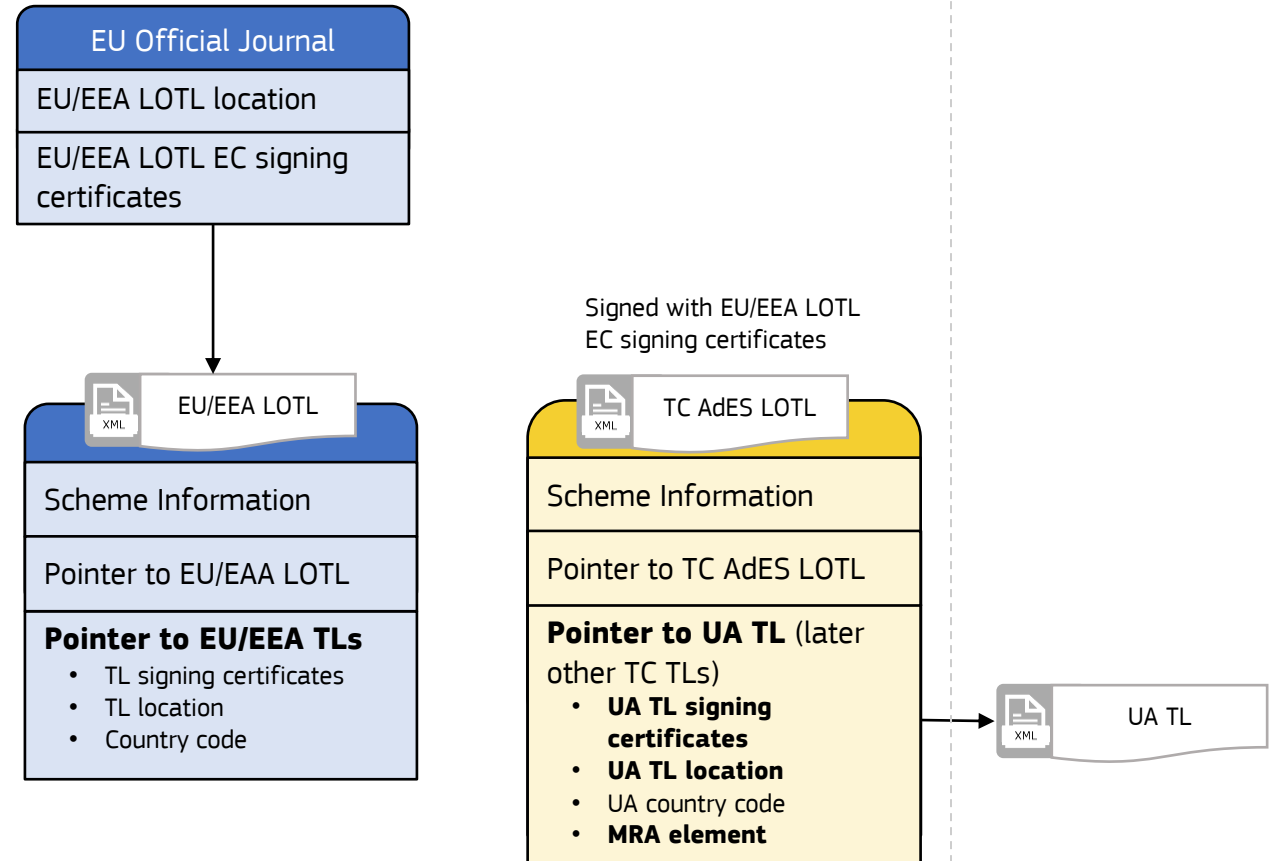
Publication of **Third Countries AdES LOTL** pointing to UA Trusted List (TL)

To allow Member States on a **voluntary** basis to:

- Download and authenticate the UA trusted list;
- Validate **UA-QES as eIDAS AdES**, using the Mutual Recognition Agreement (MRA) element.

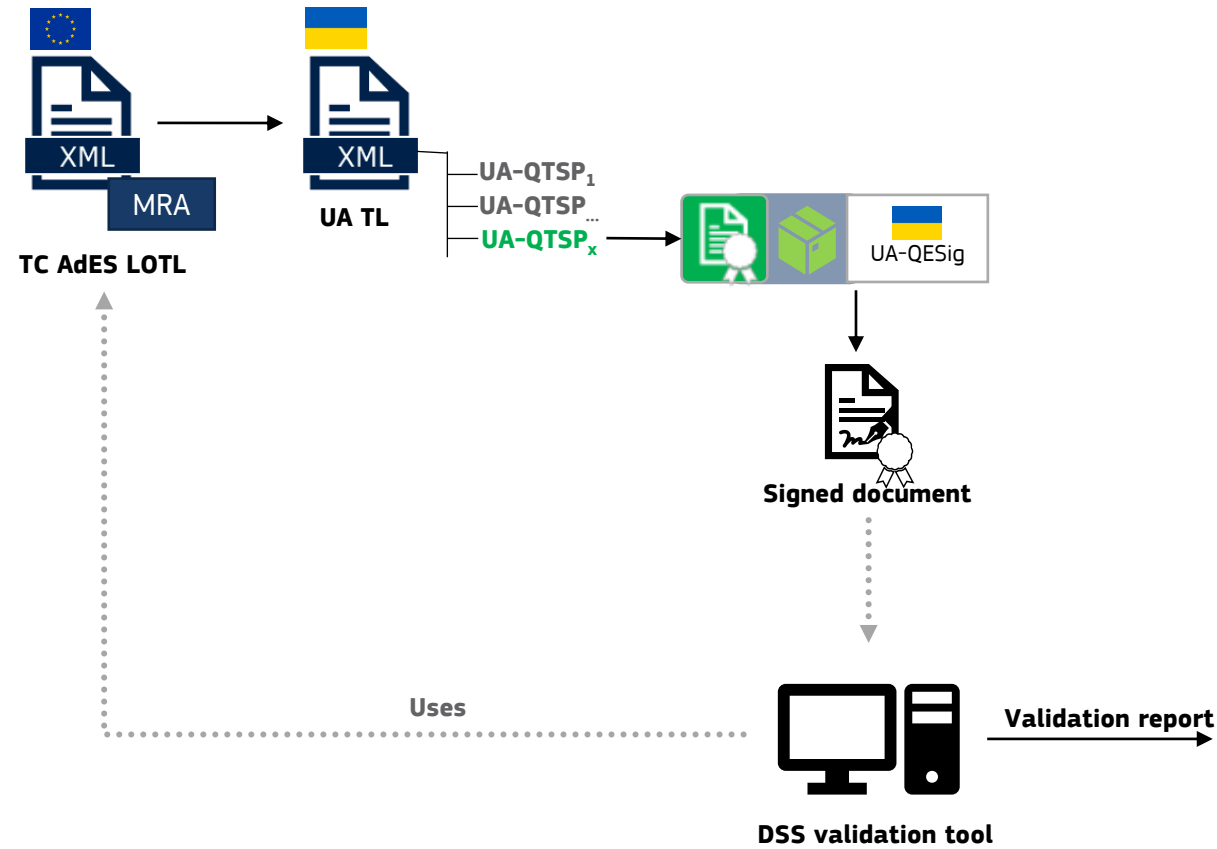
MRA element defines **equivalence** between the UA framework and the eIDAS framework.

- Equivalence statements are defined **within** the AdES LOTL
- Machine-processable



Recognition of UA-QES as eIDAS AdES

Update of **DSS library** to support the **processing of MRA elements**



Signature SIGNATURE_TEST-User-Ca_20220725-1623

Qualification: AdESig ⓘ

Qualification Details:

- The certificate is not related to a CA/QCI
- The certificate is not qualified at (best) signing time!
- The certificate is not qualified at issuance time!
- The private key does not reside in a QSCD at (best) signing time!
- The trusted list validation is supported by an enacted trust service equivalence mapping, in the context of Article 27(1) and Article 37(1) of Regulation (EU) No 910/2014.

Signature format: PKCS7-B

Indication: TOTAL_PASSED ✓

Certificate Chain:

- TEST User Ca
- Administrator ITS CCA (CA TEST)
- Central certification authority (ROOT TEST)

On claimed time: 2022-07-25 16:23:10 (UTC)

Best signature time: 2022-09-15 14:56:47 (UTC) ⓘ

Signature position: 1 out of 1

Signature scope: Full PDF (FULL)
The document ByteRange : [0, 29430, 160504, 407]

Timestamps ⓘ

Document Information

Signatures status: 1 valid signatures, out of 1

Document name: PAdES-B-B.pdf

- **Website on Pilot CEF eSig BB international compatibility**
 - <https://esignature.ec.europa.eu/intl-comp/dss-demo/>
- **Documentation**
 - MRA **CookBook**
 - eIDAS Article 14 **Assessment Check-List**
 - MRA element **specification** (and XML Schema Definition)
 - MRA element **usage**
 - **Validation policy** based on TC AdES LOTL

4

Presentation of the path to mutual recognition

Olivier DELOS



Back to the table of contents

International recognition in eIDAS

Recognition of qualified trust services

Article 14

- By means of an Art.218 TFEU (Trade) Agreement
- Recognition of **3rd Country trust service** (TC-TS) by 3rd Country trust service provider (TC-TSP) as **legally equivalent to EU qualified trust service** (EU-QTS) by EU qualified TSP (EU-QTSP).
- Provided
 - TC-TS and TC-TSP meet the eIDAS requirements for EU-QTS by EU-QTSP
 - Reciprocity

So far not implemented, i.e.

- no TC-TS/TC-TSP recognised as “qualified” in EU
- no TC-Qualified Esignature (QESig) is recognised as legally equivalent to EU-QESig
- ... but what about TC-QESig recognised as EU-AdESig ?

International recognition in eIDAS

Recognition of advanced electronic signatures

Articles 3(11), 26, 25(1), 27(1)

No territoriality requirement for advanced electronic signatures (AdESig)

- Art.3(11), 26* When TC-(Q)Esig meets the requirements of Art.26,
- Art.25(1)* they cannot be denied legal effect and admissibility as evidence in legal proceedings solely because electronic or not EU-QESig, and
- Art.27(1)* if they comply with standards listed in CID (EU) 2015/1506, Member State public sector online services requiring an AdESig shall recognise them irrespectively from where they originate







Note: applies mutatis mutandis to eSeals

Path to mutual recognition

Recognition of qualified trust services

eIDAS 1.0 (Art.14)

- Non-EU TSP/TS must meet eIDAS QTSP/QTS requirements
- Reciprocity
- Trade agreement

- Provision of QC for eSignatures → QESig 
- Provision of QC for eSeals → QESeal  Data integrity & Proof of data origin 
- Provision of QC for website auth^o 
- Qualified validation of QESig  Trustworthy results for validation of QESig/QESeal
- Qualified validation of QESeal
- Qualified preservation of QESig  Trustworthy assurance of long term evidentiary value of QESig/QESeal
- Qualified preservation of QESeal
- Provision of qualified time stamps  presumption of the accuracy of the date & time and integrity of the time stamped data
- Qualified electronic registered delivery services  presumption of integrity of the registered data, the sending of that data by the identified sender, its receipt by the identified addressee and the accuracy of date & time of sending and receipt



eIDAS 2.0 (Art.14)

- Trade agreement or **Implementing Act/Decision**
- Non-EU TSP/TS must meet eIDAS QTSP/QTS requirements
- Reciprocity

+Trusted list (MRA cookbook)

Above 9 QTSs + **potentially**

- Provision of Qualified electronic attestations of attributes
- Qualified electronic archiving
- Qualified electronic ledgers
- Qualified service for the management of remote qualified electronic signature/seal creation devices

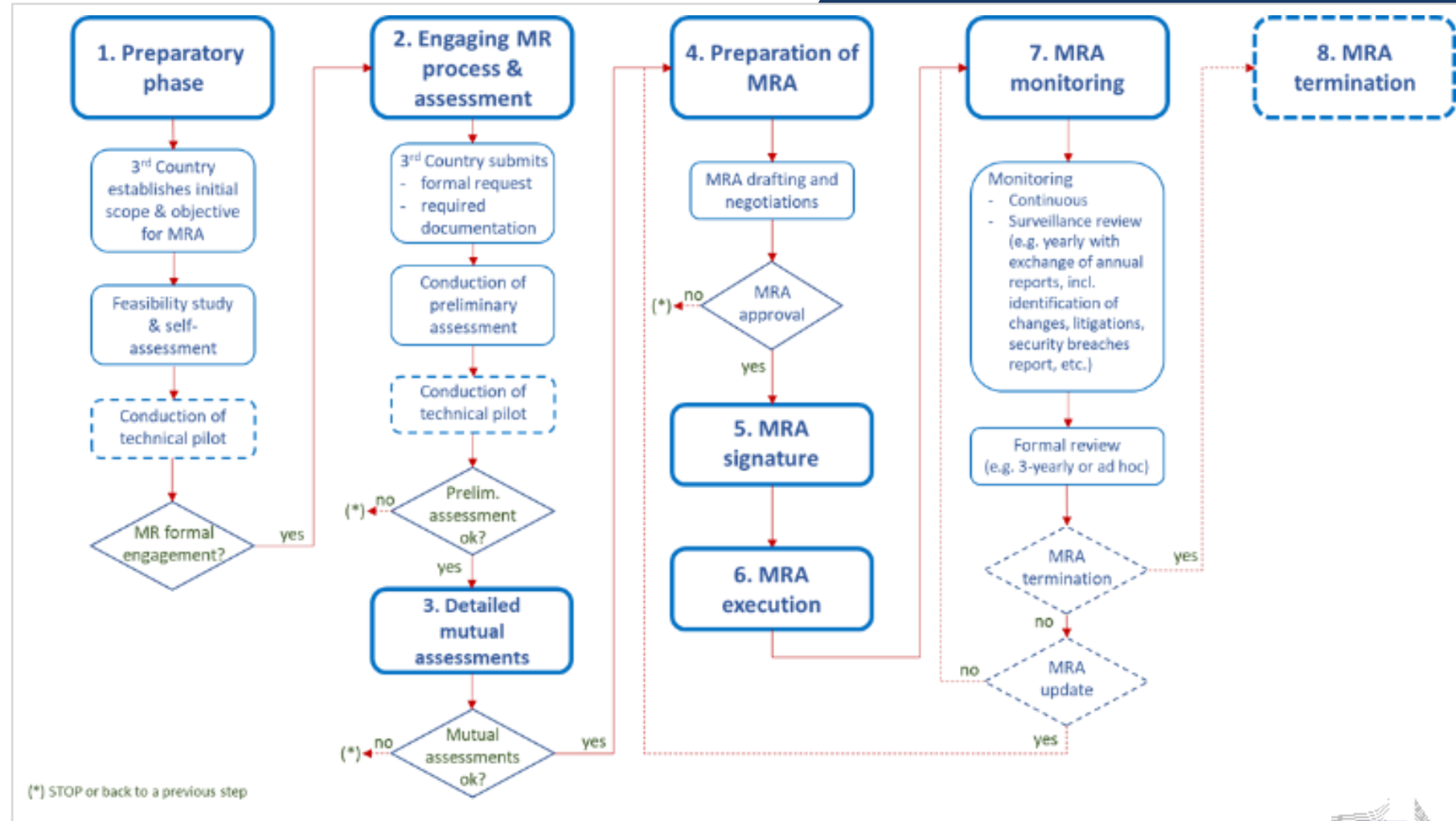
Path to mutual recognition

Recognition of qualified trust services

Art.14 Mutual recognition agreement (MRA) can be a long journey

A typical eIDAS Art.14 MRA life-cycle process flow

- **Step 1:** EC provides guidance & technical pilot tools to assist TCs.
- Note on **steps 4 & 5:** it is up to Council to decide if/when formal negotiations should be opened for conclusion of a MRA.



Path to mutual recognition

Recognition of qualified trust services

- EC provides **guidance & technical pilot tools** to assist 3rd countries
- Assessment on **four pillars** (Legal, Supervision & auditing, Technical, Trusted List)
- **Website** on Pilot CEF eSig BB international compatibility
<https://eidas.ec.europa.eu/efda/home/#/screen/international>



MRA Cookbook

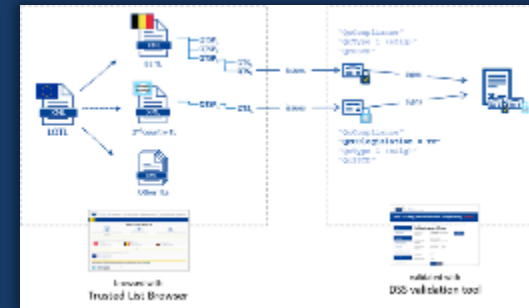


eIDAS Article 14
Assessment Check-
List
(4 pillars)



Trusted List support

- MRA element specification (and XML Schema Definition)
- MRA element usage
- In EULOTL & Foreign TL



Technical tools

- Test LOTL
- Small test PKI
- Sample signed document
- DSS library based web application to validate signatures/certificates
- Updated TL Browser

Path to mutual recognition

Recognition of qualified trust services

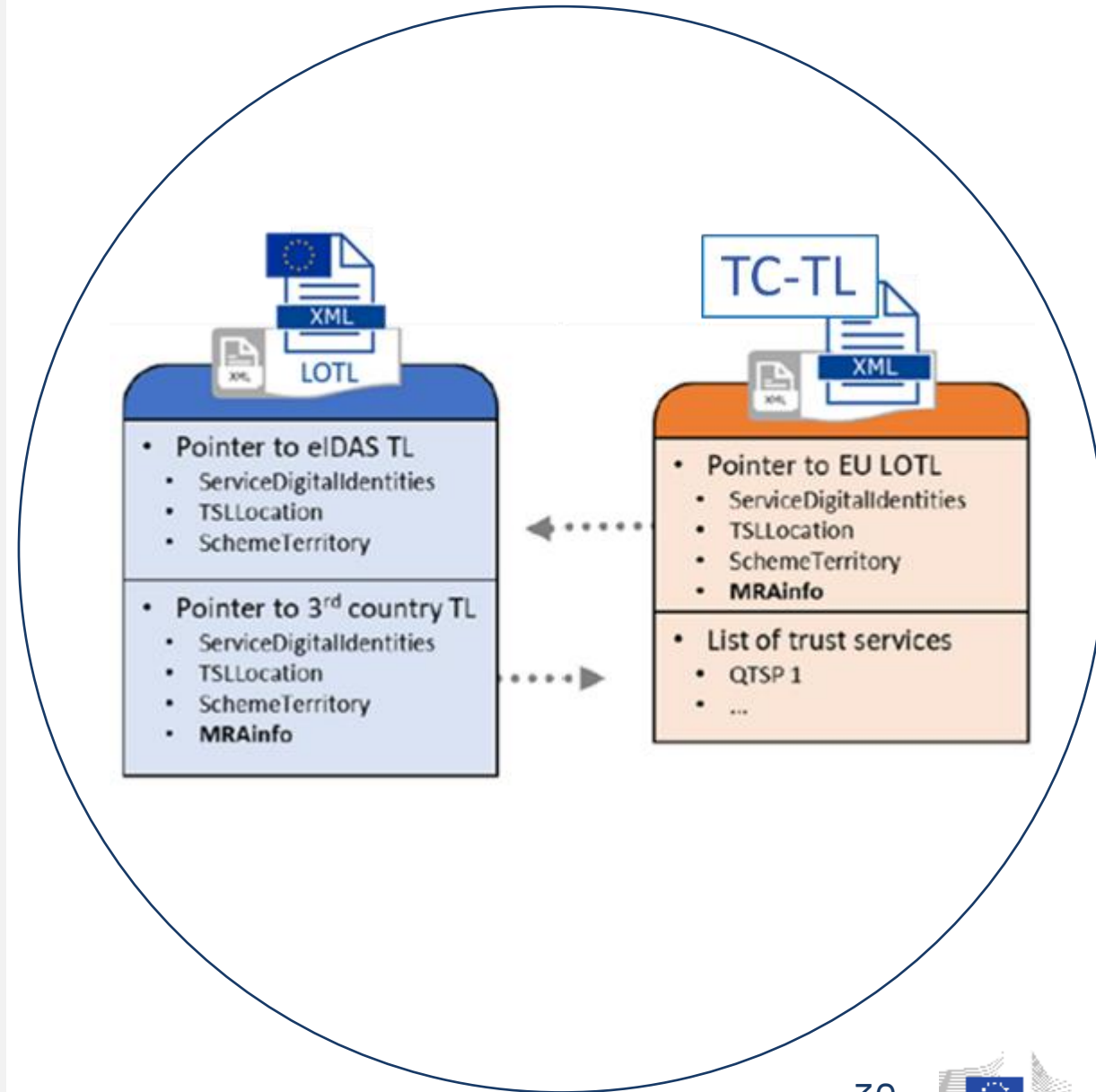
Once Art.14 MRA is *signed*

- **EU-LOTL**

- To **include a pointer** to towards the **TC-TL**
- Which pointer includes an MRA-Info element declaring, which **TC-TS type is recognised equivalent to which EU-QTS type** to facilitate validation of the TC-TS outputs as equivalent to the EU ones, with full history

- **TC-TL**

- To **include a pointer** to towards the **EU-LOTL**
- Which pointer includes an MRA-Info element declaring, which **EU-QTS type is equivalent to which TC-TS type** to facilitate validation of the EU-QTS outputs as equivalent to the TC ones, with full history



Path to mutual recognition

Recognition of advanced electronic signatures

No territoriality requirement for eSignatures to be valid EU-AdESig

COM aims to facilitate the recognition of TC electronic signatures as AdESig by Art.27 Member States public online services

TC-AdES-LOTL

- Aims to facilitate the validation and recognition of TC-QESig (*) as meeting the requirements of EU-AdESig by recognising TC-Qualified Certificates for eSignatures issued by TC-QTSP as a means to support valid EU-AdESig [under certain conditions](#)

(*) As far as they meet EU-QESig similar requirements

TC-AdES configuration mode of DSS

- Validation of TC electronic signatures as meeting the requirements of EU-AdESig based on the TC-AdES-LOTL [under certain validation assumptions](#)

Path to mutual recognition

Recognition of advanced electronic signatures

TC-AdES-LOTL – Conditions for pointing to a TC-TL

- TC to establish, maintain and publish a TL with constitutive (or assimilated) value with regards to the listing of TC-QTSP and the TC-QTS they provide.
- TC-QTSP and the TC-QTS they provide to meet similar when not equivalent requirements than those applying to EU-QTSP and EU-QTS.
 - Based on TC self-assessment using the MRA-Cookbook
- TL compliant with ETSI TS 119 612 and MRA-Cookbook requirements
 - E.g. clearly identifies the type of TC-QTS issuing TC-QC_for_eSig
- TC-QC_for_eSig meet ETSI TS 319 412 part 2 and part 5, in particular include:
 - QcCompliance, QcType and QcCClegislation statements declaring they are TC-QC_for_eSig
 - Either QcSSCD (EU QSCD) or specific CP OID (TC-TL confirmed) declaring use of TC-QSCD meeting similar requirements applicable to EU-QSCD

TC-AdES-LOTL include pointer to TC-TL with MRA-info element based on above rules

[option] TC-TL to point to the production EU-LOTL with/without MRA-info

Path to mutual recognition

Recognition of advanced electronic signatures

TC-AdES configuration mode of DSS - Validation assumptions

- Validation of TC electronic signatures based on TC-AdES-LOTL in order to be **technical able to reasonably consider them as EU-AdESig.**
- Validation based on:
 - The content of the TC signing certificate (i.e. its certificate profile);
 - Supported / confirmed by the content of the TC TL;
 - EU signatures formats (CID 2015/1506 and newer baselines) and EU-recognized signatures algorithms (from SOG-IS); and
 - Following eIDAS Art.32 requirements, as implemented by ETSI TS 119 172-4 (relying on ETSI EN 319 102-1 and ETSI TS 119 615), mutatis mutandis

...not targeting TC-QESig validation rules as applicable in TC but validating TC eSig for being “as close as possible” to EU-QESig, with the application of EU QESig validation rules, mutatis mutandis, to ensure they meet EU-AdESig requirements.



Break

10'



[Back to the table of contents](#)

5

Overview of the technical specifications part of the Commission pilot

Olivier BARETTE



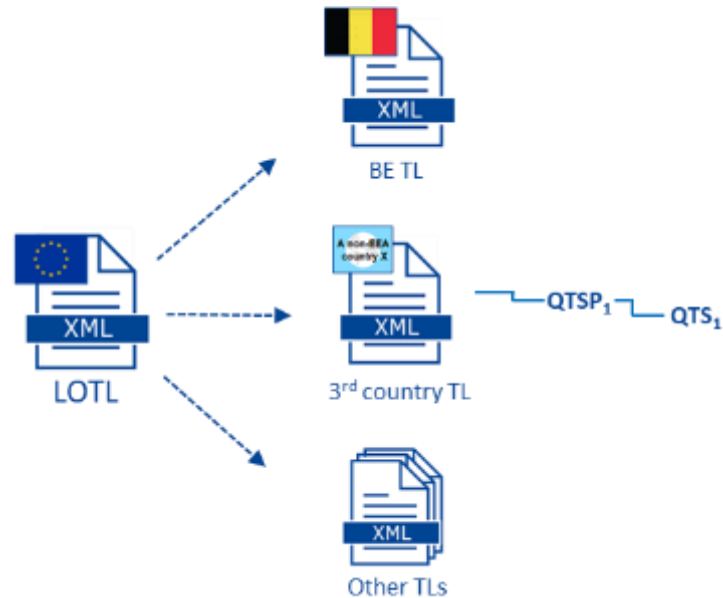
*Back to the table
of contents*

MRA-Info element

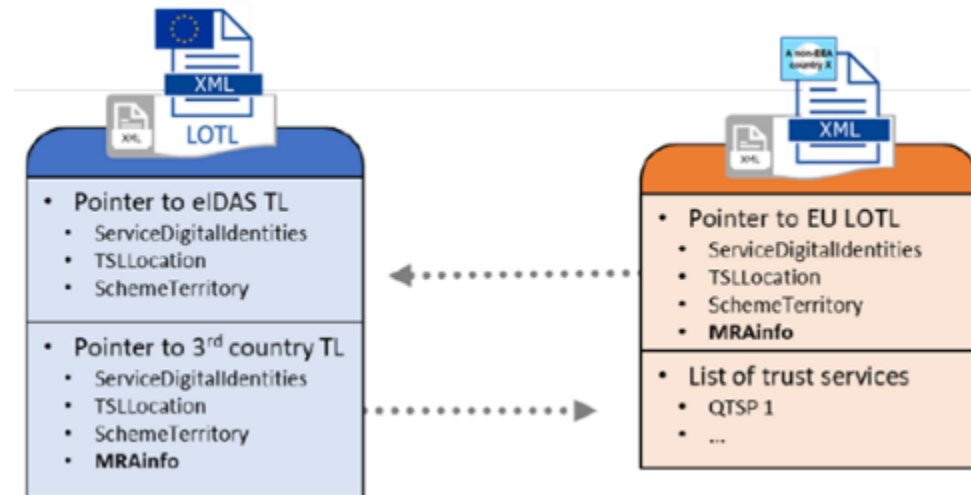
Usage n°1: Recognition of qualified trust services

Demonstrate the **technical feasibility** of mutual recognition of electronic signatures:

- ✓ 3rd Country qualified signature recognized by EU
- ✓ EU qualified signature recognized by 3rd Country



- Specification of an **MRA element** to be added in the **pointer** to the corresponding Trusted List.





TC

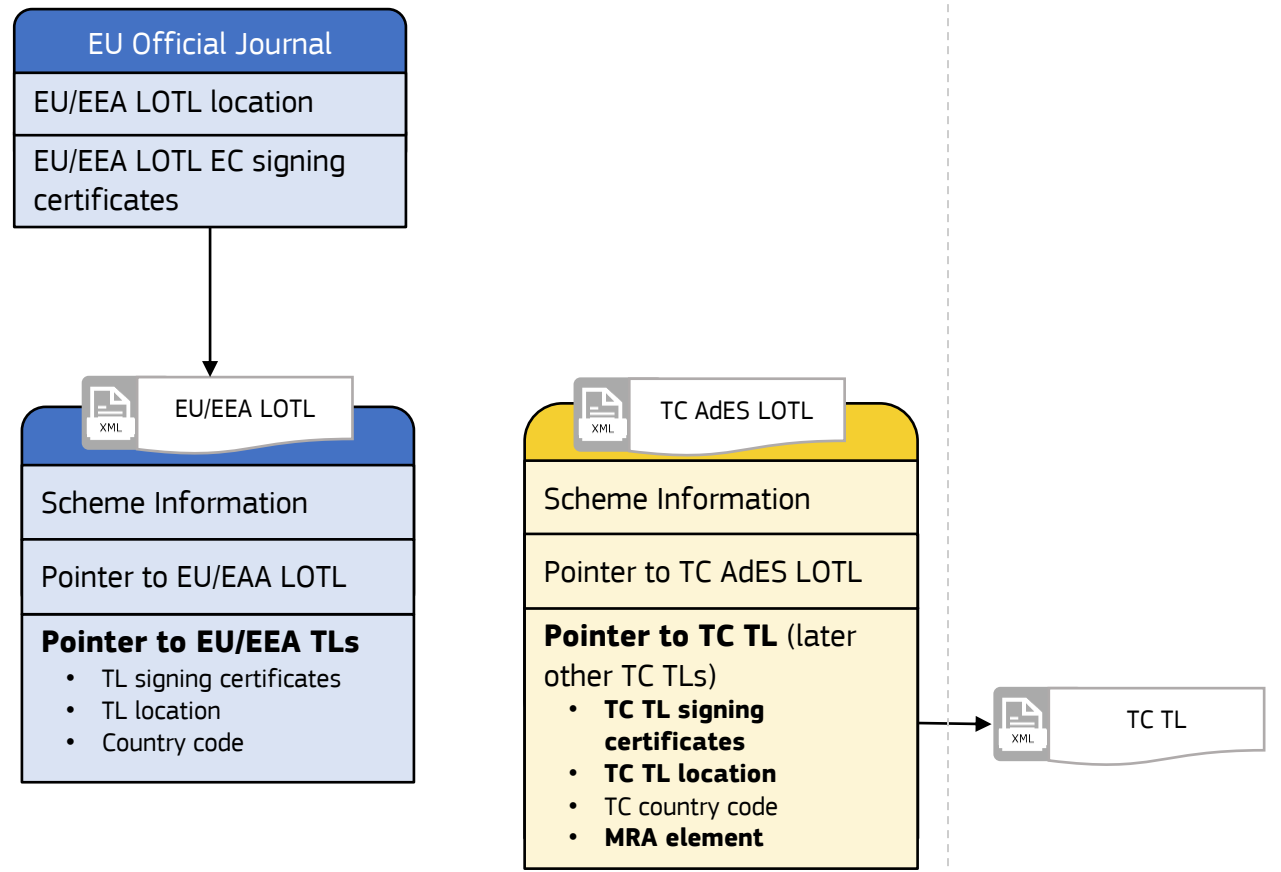
MRA-Info element

Usage n°2: Recognition of advanced electronic signatures

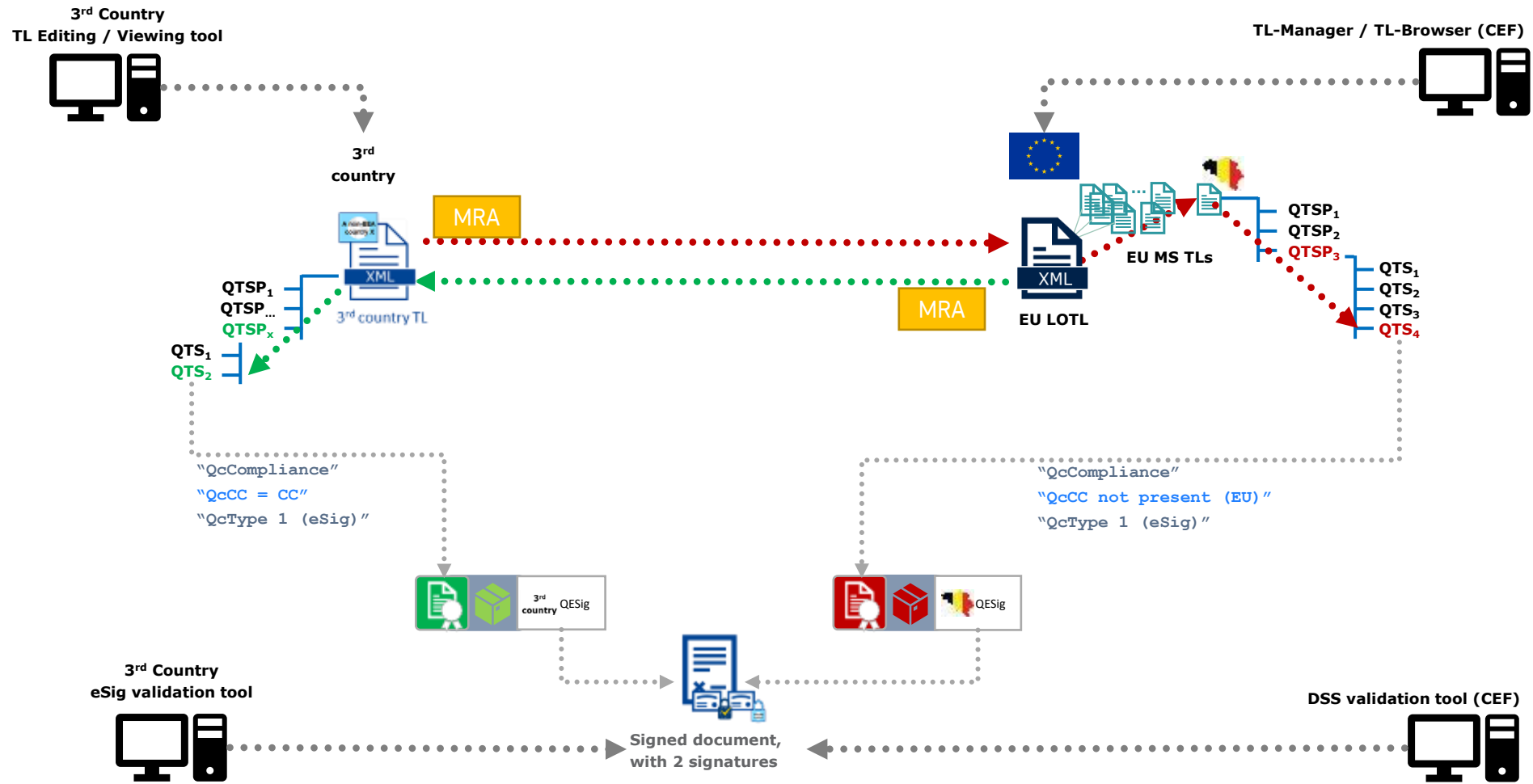
Publication of **TC AdES LOTL** pointing to TC Trusted List (TL)

To allow Member States on a **voluntary** basis to:

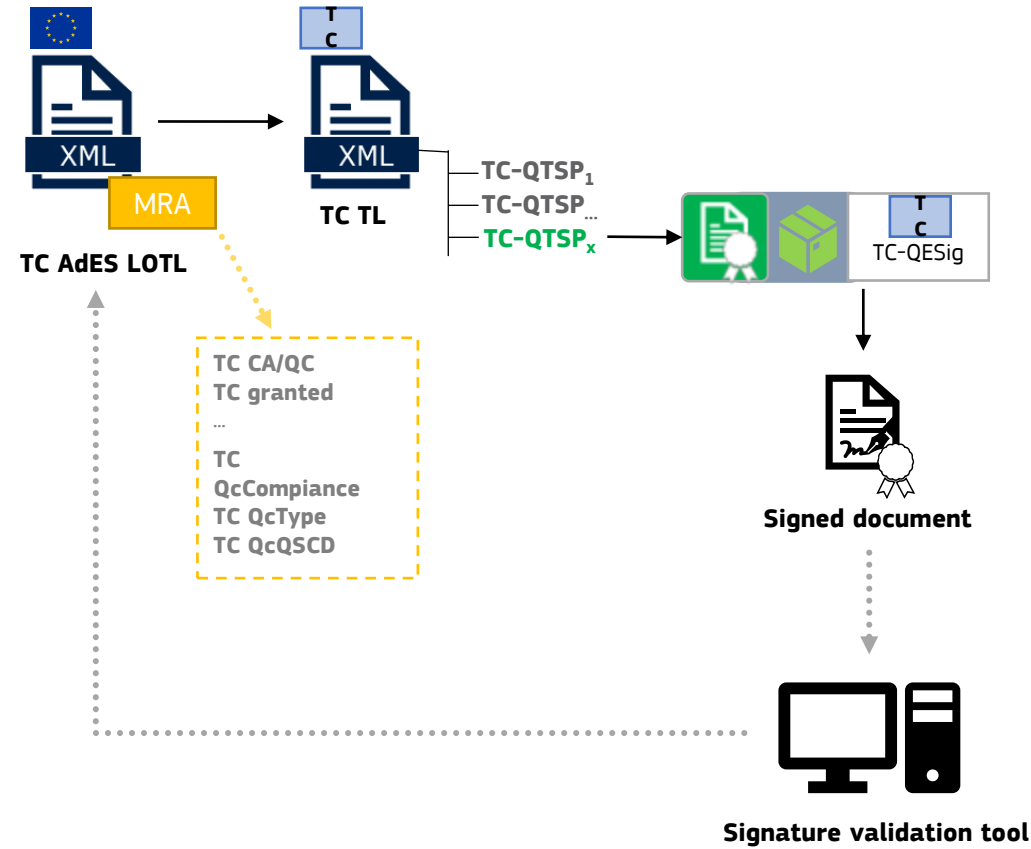
- Download and authenticate the TC trusted list;
- Validate **TC-QES as eIDAS AdES**, using the Mutual Recognition Agreement (MRA) element.



Usage n°1: Recognition of qualified trust services



Usage n°2: Recognition of advanced electronic signatures



MRA-Info element specifications

Illustration based on the TC AdES LOTL

MutualRecognitionAgreementInformation element as an additional information included to the OtherTSLPointer element of the “Pointers to other TSLs”.

This MRA Info element contains a sequence of TrustServiceEquivalenceInformation element.

```

-<OtherTSLPointer>
  +<ServiceDigitalIdentities></ServiceDigitalIdentities>
  <TSLLocation>https://czo.gov.ua/download/4/TL-UA-EC.xml</TSLLocation>
  -<AdditionalInformation>
    -<OtherInformation></OtherInformation>
    -<OtherInformation>
      <SchemeTerritory>UA</SchemeTerritory>
    </OtherInformation>
    +<OtherInformation></OtherInformation>
    -<OtherInformation></OtherInformation>
    +<OtherInformation></OtherInformation>
    -<OtherInformation></OtherInformation>
    -<mra:MutualRecognitionAgreementInformation MRADepth="1" pointedContractingPartyLegislation="https://czo.gov.ua/uaschemcinfo" pointingContractingPartyLegislation="https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=uriserv:OJ.L_.2014.257.01.0073.01.ENG" technicalType="1" version="2">
      -<mra:TrustServiceEquivalenceInformation>
        -<mra:TrustServiceLegalSchemeIdentifier></mra:TrustServiceLegalSchemeIdentifier>
        <mra:TrustServiceTSLTypeEquivalenceList>
          -<mra:TrustServiceTSLTypeListPointingParty>
            -<mra:TrustServiceTSLType>
              <ServiceTypeIdentifier>http://uri.etsi.org/TrstSvc/Svctype/CA/PKC</ServiceTypeIdentifier>
              -<AdditionalServiceInformation>
                -<URI xml:lang="en">
                  http://uri.etsi.org/TrstSvc/TrustedList/SvcInfoExt/ForeSignatures
                </URI>
              </AdditionalServiceInformation>
            </mra:TrustServiceTSLType>
          </mra:TrustServiceTSLTypeListPointingParty>
          -<mra:TrustServiceTSLTypeListPointedParty>
            -<mra:TrustServiceTSLType>
              <ServiceTypeIdentifier>http://czo.gov.ua/TrstSvc/Svctype/CA/QC</ServiceTypeIdentifier>
              -<AdditionalServiceInformation>
                -<URI xml:lang="en">
                  http://uri.etsi.org/TrstSvc/TrustedList/SvcInfoExt/ForeSignatures
                </URI>
              </AdditionalServiceInformation>
            </mra:TrustServiceTSLType>
          </mra:TrustServiceTSLTypeListPointedParty>
        </mra:TrustServiceTSLTypeEquivalenceList>
      </mra:TrustServiceEquivalenceInformation>
    </mra:MutualRecognitionAgreementInformation>
  </AdditionalInformation>
</OtherTSLPointer>

```


MRA-Info element specifications

Trust Service Equivalence Information

That contain information about the equivalence mapping :

TrustService**LegalIdentifier**

TrustService**TSLType**EquivalenceList

TrustService**EquivalenceStatus**

TrustServiceEquivalenceStatus**StartingTime**

TrustService**TSLStatus**EquivalenceList

CertificateContentReferencesEquivalenceList

TrustServiceTSL**QualificationExtension**EquivalenceList

TrustServiceEquivalence**History**

```

-<OtherTSLPointer>
+<ServiceDigitalIdentities></ServiceDigitalIdentities>
  <TSSLocation>https://czo.gov.ua/download/tl/TL-UA-EC.xml</TSSLocation>
-<AdditionalInformation>
+<OtherInformation></OtherInformation>
-<OtherInformation>
  <SchemeTerritory>UA</SchemeTerritory>
</OtherInformation>
+<OtherInformation></OtherInformation>
+<OtherInformation></OtherInformation>
+<OtherInformation></OtherInformation>
-<OtherInformation>
  -<mra:MutualRecognitionAgreementInformation MRADepth="1" pointedContractIn
lex.europa.eu/legal-content/EN/TXT/?uri=uriserv:OJ.L_.2014.257.01.0073.01.ENG" tecl
-<mra:TrustServiceEquivalenceInformation>
  <mra:TrustServiceLegalIdentifier>PKCForESig</mra:TrustServiceLegalIdenti
-<mra:TrustServiceTSLTypeEquivalenceList>
-<mra:TrustServiceTSLTypeListPointingParty>
  -<mra:TrustServiceTSLType>
    <ServiceTypeIdentifier>http://uri.etsi.org/TrstSvc/Svctype/CA/PKC</Servi
  -<AdditionalServiceInformation>
    -<URI xml:lang="en">
      http://uri.etsi.org/TrstSvc/TrustedList/SvcInfoExt/ForeSignatures
    </URI>
    </AdditionalServiceInformation>
  </mra:TrustServiceTSLType>
</mra:TrustServiceTSLTypeListPointingParty>
-<mra:TrustServiceTSLTypeListPointedParty>
  -<mra:TrustServiceTSLType>
    <ServiceTypeIdentifier>http://czo.gov.ua/TrstSvc/Svctype/CA/QC</Servi
  -<AdditionalServiceInformation>
    -<URI xml:lang="en">
      http://uri.etsi.org/TrstSvc/TrustedList/SvcInfoExt/ForeSignatures
    </URI>
    </AdditionalServiceInformation>
  </mra:TrustServiceTSLType>
</mra:TrustServiceTSLTypeListPointedParty>

```

MRA-Info element specifications

Equivalences between information in the **Pointing Party** and information in the **Pointed Party**

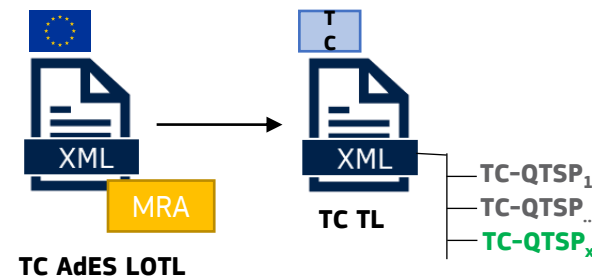
TrustService**TSLType**EquivalenceList

- Identifies the type of TC-QTS issuing TC-QC_for_eSig

CertificateContentReferencesEquivalenceList

- Identifies QcCompliance, QcType and QcCClegislation statements declaring they are TC-QC_for_eSig
- Either QcSSCD (EU QSCD) or specific CP OID (TC-TL confirmed) declaring use of TC-QSCD meeting similar requirements applicable to EU-QSCD

****PointingParty - **PointedParty**



How to identify the TC QTS issuing QCs in the TL: “TC_CA/QC”

```

-<OtherTSLPointer>
+<ServiceDigitalIdentities></ServiceDigitalIdentities>
  <TSLLocation>https://czo.gov.ua/download/tl/TL-UA-EC.xml</TSLLocation>
-<AdditionalInformation>
+<OtherInformation></OtherInformation>
-<OtherInformation>
  <SchemeTerritory>UA</SchemeTerritory>
</OtherInformation>
+<OtherInformation></OtherInformation>
+<OtherInformation></OtherInformation>
+<OtherInformation></OtherInformation>
-<OtherInformation>
  -<mra:MutualRecognitionAgreementInformation MRADepth="1" pointedContractingPartyLegislation="https://czo.gov.ua/uaschemeinfo" pointingContractingPartyLegislation="https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=uriserv:OJL_2014_257_01_0073_01_ENG" technicalType="1" version="2">
    -<mra:TrustServiceEquivalenceInformation>
      <mra:TrustServiceLegalIdentifier>PKCForESig</mra:TrustServiceLegalIdentifier>
      -<mra:TrustServiceTSLTypeEquivalenceList>
        -<mra:TrustServiceTSLTypeListPointingParty>
          -<mra:TrustServiceTSLType>
            <ServiceTypeIdentifier>http://uri.etsi.org/TrstSvc/Svctype/CA/PKC</ServiceTypeIdentifier>
            -<AdditionalServiceInformation>
              -<URI xml:lang="en">
                http://uri.etsi.org/TrstSvc/TrustedList/SvcInfoExt/ForeSignatures
              </URI>
            </AdditionalServiceInformation>
          </mra:TrustServiceTSLType>
        </mra:TrustServiceTSLTypeListPointingParty>
        -<mra:TrustServiceTSLTypeListPointedParty>
          -<mra:TrustServiceTSLType>
            <ServiceTypeIdentifier>http://czo.gov.ua/TrstSvc/Svctype/CA/QC</ServiceTypeIdentifier>
            -<AdditionalServiceInformation>
              -<URI xml:lang="en">
                http://uri.etsi.org/TrstSvc/TrustedList/SvcInfoExt/ForeSignatures
              </URI>
            </AdditionalServiceInformation>
          </mra:TrustServiceTSLType>
        </mra:TrustServiceTSLTypeListPointedParty>
      </mra:TrustServiceTSLTypeEquivalenceList>
    </mra:TrustServiceEquivalenceInformation>
  </mra:MutualRecognitionAgreementInformation MRADepth="1" pointedContractingPartyLegislation="https://czo.gov.ua/uaschemeinfo" pointingContractingPartyLegislation="https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=uriserv:OJL_2014_257_01_0073_01_ENG" technicalType="1" version="2">

```

How to identify the TC QC certificate profile: e.g. “TC_QcCompliance”

Validation policy (Signature Applicability Rules)

```

<mra:CertificateContentReferenceEquivalenceContext>http://ec.europa.eu/tools/lotl/mra/QcCompliance</mra:CertificateContentReferenceEquivalenceContext>
<mra:CertificateContentDeclarationPointingParty assert="none">
  <ns5:Description>
    UA qualified certificates may not be considered as equivalent to eIDAS qualified certificates. For eIDAS qualified certificates, the QcStatement "QcCompliance" (id-etsi-qcs-QcCompliance OID "0.4.0.1862.1.1") is the reference machine processable statement included in a certificate to declare (as a statement made by the issuing TSP) and to confirm (as a benchmark for establishing the content of the corresponding TL trust service entry) that it has been issued as a qualified certificate.
  </ns5:Description>
  <ns5:otherCriteriaList>
    <mra:QcStatementSet>
      <mra:QcStatement>
        <mra:QcStatementId>
          <ns4:Identifier Qualifier="OIDAsURN">urn:oid:0.4.0.1862.1.1</ns4:Identifier>
        </mra:QcStatementId>
        </mra:QcStatement>
      </mra:QcStatementSet>
    </ns5:otherCriteriaList>
  </mra:CertificateContentDeclarationPointingParty>
<mra:CertificateContentDeclarationPointedParty assert="all">
  <ns5:Description>
    For UA qualified certificates, the QcStatements "QcCompliance" (id-etsi-qcs-QcCompliance OID "0.4.0.1862.1.1") and "QcCClegislation" (id-etsi-qcs-QcCClegislation OID "0.4.0.1862.1.7") with value "UA" are, when used together, the reference machine processable statements included in a certificate to declare (as a statement made by the issuing TSP) and to confirm (as a benchmark for establishing the content of the corresponding TL trust service entry) that it has been issued as a UA qualified certificate.
  </ns5:Description>

```

```

<ns5:otherCriteriaList>
  <mra:QcStatementSet>
    <mra:QcStatement>
      <mra:QcStatementId>
        <ns4:Identifier Qualifier="OIDAsURN">urn:oid:0.4.0.1862.1.1</ns4:Identifier>
      </mra:QcStatementId>
      </mra:QcStatement>
    <mra:QcStatement>
      <mra:QcStatementId>
        <ns4:Identifier Qualifier="OIDAsURN">urn:oid:0.4.0.1862.1.7</ns4:Identifier>
      </mra:QcStatementId>
      <mra:QcStatementInfo>
        <mra:QcCClegislation>UA</mra:QcCClegislation>
      </mra:QcStatementInfo>
    </mra:QcStatementSet>
  </ns5:otherCriteriaList>

```

How to perform a TC AdES LOTL-based signature validation

Validation policy (Signature Applicability Rules)

Rules for the technical validation of digital signatures originating from 3rd countries and the determination of their **applicability to the specific context** of **Article 27** (and consequently Article 26) of the eIDAS Regulation, i.e. to determine whether they can be (technically) considered as EU advanced electronic signatures using the **TC AdES LOTL** and the corresponding TC TLs in the sense of eIDAS.

Based on **ETSI TS 119 172-4**, relying on ETSI TS 119 615 and ETSI EN 319 102, parameterized with:

- How to rely on the **TC AdES LOTL**
- How to identify the TC QTS issuing QCs (for eSig / for eSeal) in the TL: **“TC_CA/QC”**
- How to identify in the TC QC:
 - **“TC_QcCompliance”**
 - **“TC_QcType”**
 - **“TC_QcQSCD”**

The SAR document is available at: <https://eidas.ec.europa.eu/efda/tl-browser/#/screen/tc-tl> or in the MRA Bundle on the pilot page <https://eidas.ec.europa.eu/efda/intl-pilot/#/screen/home/demo>

6

Practical aspects: support of MRA element in DSS

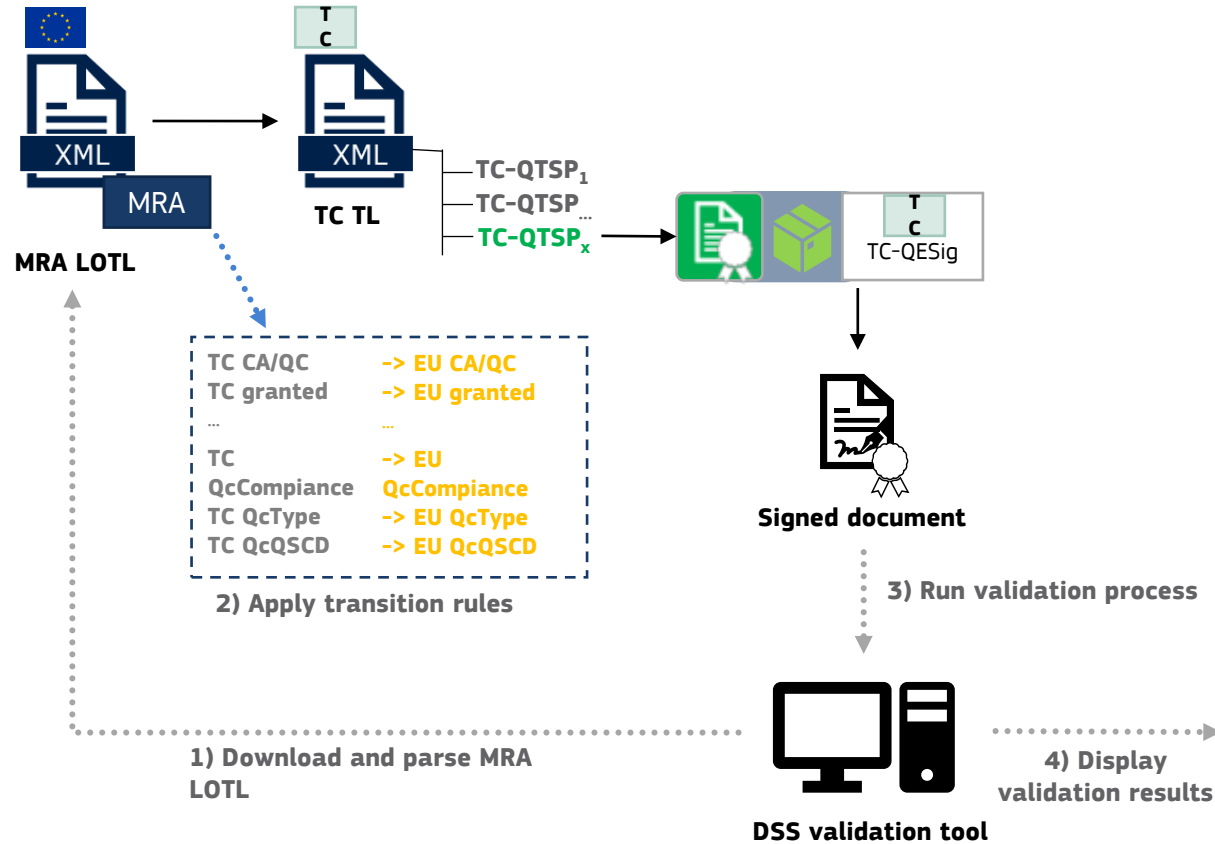
Aleksandr BELIAKOV



Back to the table of contents

1. Parsing the LOTL and MRA element;
2. Applying MRA transition rules;
3. Running signature/certificate validation;
4. Displaying the results.

Validation process using MRA element in DSS



Signature S-A0D6186DF85A5AD1B5B186144255EF7E1A795D08374F480B9C04FA6D48831A1D

Signature filename: META-INF/signatures001.xml

Qualification: AdESig

Qualification Details:

- The certificate is not related to a CA/QCI
- The certificate is not qualified at (best) signing time!
- The certificate is not qualified at issuance time!
- The private key does not reside in a QSCD at (best) signing time!

The validation is relying on the TC AdES List of the lists providing information notified by third countries to facilitate the validation of electronic signatures or seals created in third countries as meeting the requirements of (EU) advanced electronic signatures or seals in accordance with Regulation (EU) No 910/2014.

Signature format: XAdES-BASELINE-LT

Indication: TOTAL_PASSED ✓

Certificate Chain:

- Alice Doe
- TC Qualified Trust Services Provider
- Central certification authority

On claimed time: 2022-12-23 09:12:17 (UTC)

Best signature time: 2022-12-23 09:12:17 (UTC) ⓘ

Signature position: 1 out of 1

Signature scope: (FULL)
Full document

MutualRecognitionAgreementInformation element

```
▼<PointersToOtherTSL>
  ▶<OtherTSLPointer>
    ...
  </OtherTSLPointer>
  ▼<OtherTSLPointer>
    ▼<ServiceDigitalIdentities>
      ▶<ServiceDigitalIdentity>
        ...
      </ServiceDigitalIdentity>
      ▶<ServiceDigitalIdentity>
        ...
      </ServiceDigitalIdentity>
    </ServiceDigitalIdentities>
    <TSLLocation>https://gov.tc/download/tl/TL-TC-EC.xml</TSLLocation>
  ▼<AdditionalInformation>
    ...
  ▼<OtherInformation>
    ▼<mra:MutualRecognitionAgreementInformation MRADepth="1" pointedContractingPartyLegislation=
      ▶<mra:TrustServiceEquivalenceInformation>
        ...
      </mra:TrustServiceEquivalenceInformation>
      ▶<mra:TrustServiceEquivalenceInformation>
        ...
      </mra:TrustServiceEquivalenceInformation>
    </mra:MutualRecognitionAgreementInformation>
  </OtherInformation>
</AdditionalInformation>
</OtherTSLPointer>
</PointersToOtherTSL>
```

MRA Information



1

2

3

4

5

6

7

8

9

MutualRecognitionAgreementInformation element defines equivalence mapping between a pointed party's legislation (e.g. a Third Country) and the pointing party's legislation (e.g. EU).

The rules can be applied for:

- Trust Services (including TSLType, status, qualifiers, etc.);
- Certificate content (such as certificate policies, key usages, QCStatements, etc.).

[1](#)[2](#)[3](#)[4](#)[5](#)[6](#)[7](#)[8](#)[9](#)

For a Trust Service:

```
▼<mra:TrustServiceTSLStatusListPointingParty>  
  <ServiceStatus>http://uri.etsi.org/TrstSvc/TrustedList/Svcstatus/recognisedatnationallevel</ServiceStatus>  
</mra:TrustServiceTSLStatusListPointingParty>  
▼<mra:TrustServiceTSLStatusListPointedParty>  
  <ServiceStatus>http://gov.tc/TrstSvc/TrustedList/Svcstatus/granted </ServiceStatus>  
</mra:TrustServiceTSLStatusListPointedParty>
```

`http://gov.tc/TrstSvc/TrustedList/Svcstatus/granted`

defined in Third Country TL

should be understood as:

`http://uri.etsi.org/TrstSvc/TrustedList/Svcstatus/recognisedatnationallevel` in EU scope

For each matching Trust Service DSS performs transition of rules according to the mapping:

```
▼<TrustedService ServiceDigitalIdentifier="CERTIFICATE_DIIA-Qualified-Trust-Services-Provider_20221124-1149" enactedMRA="true">
  ▼<ServiceNames>
    <ServiceName lang="en">TC Qualified Trust Services Provider</ServiceName>
  </ServiceNames>
  <ServiceType>http://uri.etsi.org/TrstSvc/Svctype/CA/PKC</ServiceType>
  <Status>http://uri.etsi.org/TrstSvc/TrustedList/Svcstatus/recognisedatnationallevel</Status>
  <StartDate>2022-11-24T11:49:00Z</StartDate>
  <CapturedQualifiers/>
  ▼<AdditionalServiceInfoUris>
    <AdditionalServiceInfoUri>http://uri.etsi.org/TrstSvc/TrustedList/SvcInfoExt/ForeSignatures</AdditionalServiceInfoUri>
  </AdditionalServiceInfoUris>
  ▼<MRATrustServiceMapping>
    <TrustServiceLegalIdentifier>PKCForESig</TrustServiceLegalIdentifier>
    <EquivalenceStatusStartingTime>2022-11-24T22:00:00Z</EquivalenceStatusStartingTime>
    ▼<OriginalThirdCountryMapping>
      <ServiceType>http://gov.tc/TrstSvc/Svctype/CA/QC</ServiceType>
      <Status>http://gov.tc/TrstSvc/TrustedList/Svcstatus/granted</Status>
      <CapturedQualifiers/>
      ▼<AdditionalServiceInfoUris>
        <AdditionalServiceInfoUri>http://uri.etsi.org/TrstSvc/TrustedList/SvcInfoExt/ForeSignatures</AdditionalServiceInfoUri>
      </AdditionalServiceInfoUris>
    </OriginalThirdCountryMapping>
  </MRATrustServiceMapping>
</TrustedService>
```





For certificate content:

```
<mra:CertificateContentReferenceEquivalence>
  <mra:CertificateContentReferenceEquivalenceContext>http://ec.europa.eu/tools/lot1/mra/QcQSCD</mra:CertificateContentReferenceEquivalenceContext>
  ▼<mra:CertificateContentDeclarationPointingParty assert="all">
    ▼<ns5:otherCriteriaList>
      ▼<mra:QcStatementSet>
        ▼<mra:QcStatement>
          ▼<mra:QcStatementId>
            <ns4:Identifier Qualifier="OIDAsURN">urn:oid:0.4.0.1862.1.4</ns4:Identifier>
          </mra:QcStatementId>
          </mra:QcStatement>
        </mra:QcStatementSet>
      </ns5:otherCriteriaList>
    </mra:CertificateContentDeclarationPointingParty>
    ▼<mra:CertificateContentDeclarationPointedParty assert="all">
      ▼<ns5:PolicySet>
        ▼<ns5:PolicyIdentifier>
          <ns4:Identifier Qualifier="OIDAsURN">urn:oid:1.2.204.1.1.1.2.4.8</ns4:Identifier>
        </ns5:PolicyIdentifier>
      </ns5:PolicySet>
    </mra:CertificateContentDeclarationPointedParty>
  </mra:CertificateContentReferenceEquivalence>
```

Certificate policy 1.2.204.1.1.1.2.4.8 defined in Third Country TL

corresponds to:

0.4.0.1862.1.4 (qc-sscd) QcStatement in EU scope

The same is done for corresponding certificate's content rules:

```
▼<QcStatements enactedMRA="true">
  <QcCompliance present="false"/>
  <QcSSCD present="false"/>
  ▼<QcTypes>
    <QcType Description="qc-type-esign">0.4.0.1862.1.6.1</QcType>
  </QcTypes>
  ▼<QcCClegislation>
    <CountryName>TC</CountryName>
  </QcCClegislation>
  <SemanticsIdentifier Description="Semantics identifier for natural person">0.4.0.194121.1.1</SemanticsIdentifier>
  <OtherOIDs/>
  ▼<MRACertificateMapping>
    <EnactedTrustServiceLegalIdentifier>PKCForESig</EnactedTrustServiceLegalIdentifier>
    ▼<OriginalThirdCountryMapping>
      <QcCompliance present="true"/>
      <QcSSCD present="true"/>
      ▼<QcTypes>
        <QcType Description="qc-type-esign">0.4.0.1862.1.6.1</QcType>
      </QcTypes>
      ▼<QcCClegislation>
        <CountryName>TC</CountryName>
      </QcCClegislation>
    </OriginalThirdCountryMapping>
  </MRACertificateMapping>
</QcStatements>
```

transformed info

original content



1

2

3

4

5

6

7


8







9

Run validation process using the transformed data:

- per ETSI EN 319 102-1 (AdES validation);
- per ETSI TS 119 615 (qualification status determination).

Displaying the results

Signature S-A0D6186DF85A5AD1B5B186144255EF7E1A795D08374F480B9C04FA6D48831A1D 

| | |
|--------------------------------|---|
| Signature filename: | META-INF/signatures001.xml |
| Qualification: | AdESig  |
| Qualification Details : | <p>The certificate is not related to a CA/QC! The certificate is not qualified at (best) signing time! The certificate is not qualified at issuance time! The private key does not reside in a QSCD at (best) signing time!</p> <p>The validation is relying on the TC AdES List of the lists providing information notified by third countries to facilitate the validation of electronic signatures or seals created in third countries as meeting the requirements of (EU) advanced electronic signatures or seals in accordance with Regulation (EU) No 910/2014.</p> |
| Signature format: | XAdES-BASELINE-LT |
| Indication: | TOTAL_PASSED  |
| Certificate Chain: |  Alice Doe  TC Qualified Trust Services Provider  Central certification authority |
| On claimed time: | 2022-12-23 09:12:17 (UTC) |
| Best signature time: | 2022-12-23 09:12:17 (UTC)  |
| Signature position: | 1 out of 1 |
| Signature scope: | (FULL) Full document |

} final qualification level (in EU scope)

} qualification details (in EU scope)

} label indicating the MRA applicability

} basic signature validation information

The AdES LOTL with MRA support can be added similarly to EU LOTL:



1

2

3

4

5

6

7

8

9

The AdES LOTL with MRA support can be added similarly to EU LOTL:

1) Configure the LOTLSource:

```
LOTLSource adesLOTL = new LOTLSource();
adesLOTL.setUrl("https://ec.europa.eu/tools/lotl/mra/ades-lotl.xml"); // define the URL access point of the AdES LOTL
adesLOTL.setCertificateSource(adesTrustedKeyStore()); // provide a key source containing signing-certificate candidates of the AdES LOTL
adesLOTL.setMraSupport(true); // set the MRA support (false by default)

adesLOTL.setLotlPredicate(new XMLOtherTSLPointer().and(new TypeOtherTSLPointer(
    "http://ec.europa.eu/tools/lotl/mra/ades-lotl-tsl-type"))); // identify the TSLType pointer to be accepted for the LOTL
adesLOTL.setTIPredicate(new XMLOtherTSLPointer().and(new TypeOtherTSLPointer(
    "http://ec.europa.eu/tools/lotl/mra/ades-lotl-tsl-type")).negate()); // identify the TSLType pointer to be accepted for Tls
```



1

2

3

4

5

6

7

8

9

The AdES LOTL with MRA support can be added similarly to EU LOTL:

2) Provide the created LOTLSource to TLValidationJob:

```
TLValidationJob job = new TLValidationJob();  
job.setTrustedListCertificateSource(trustedListSource); // provide a certificate source to be populated with trust anchors  
...  
job.setListOfTrustedListSources(europeanLOTL, adesLOTL); // provide LOTLs to be loaded  
...
```

NOTE: complete configuration of TLValidationJob can be found at: [Configuration of TL validation job](#)

The AdES LOTL with MRA support can be added similarly to EU LOTL:

3) Run the TL/LOTL validation process:

```
TLValidationJob job = new TLValidationJob();
```

```
...
```

```
job.onlineRefresh(); // execute validation using online services
```

```
TLValidationJobSummary summary = job.getSummary(); // extract the downloaded information
```

4) Extract/use the validation result:

Live demo can be found at [DSS Demonstration WebApp](#)

LOTL №2 (AdES List of the Trusted Lists) <https://ec.europa.eu/tools/lotl/mra/ades-lotl.xml>

| | |
|-------------------------------|---|
| Last download attempt | 02-Mar-2023 13:19:51 |
| Last success download | 02-Mar-2023 13:19:51 |
| Download status | SYNCHRONIZED ✔ 18-Feb-2023 10:35:17 |
| Parsing status | SYNCHRONIZED ✔ 18-Feb-2023 10:35:17 |
| Validation status | SYNCHRONIZED ✔ 18-Feb-2023 10:35:17 |
| Sequence number | 2 |
| LOTL Issue Date | 23-Jan-2023 09:00:00 |
| LOTL Next Update | 22-Jul-2023 09:00:00 |
| TL Distribution Points | https://ec.europa.eu/tools/lotl/mra/ades-lotl.xml |
| Indication | TOTAL_PASSED |
| Signing Time | 23-Jan-2023 13:05:18 |
| Signing Certificate | <p>Subject Name commonName=European Commission,organizationName=COMMISSION DE L'UNION EUROPEENNE - COMMISSIE VAN DE EUROPESE UNIE,organizationIdentifier=NTRBE-0949383342,organizationalUnitName=Directorate-General for Informatics,countryName=BE</p> <p>Issuer Name commonName=QuoVadis Belgium Issuing CA G2,organizationName=QuoVadis Trustlink BVBA,organizationIdentifier=NTRBE-0537698318,countryName=BE</p> <p>Serial Number 362671125803220245397713982254227056881701192273</p> <p>Start 21-Jan-2021 09:18:38</p> <p>End 21-Jan-2024 09:58:00</p> |

Trusted Lists loaded

| Country | Seq num | Last success download | Download result | Parsing result | Validation result | TL Next Update | N° TSP | N° TS | N° Certs |
|---------|---------|-----------------------|--------------------------------------|--------------------------------------|--------------------------------------|----------------------|--------|-------|----------|
| Ukraine | 3 | 02-Mar-2023 13:19:52 | ✔ | ✔ | ✔ | 17-May-2023 17:44:00 | 2 | 6 | 6 |

7

Introduction to DSS

Aleksandr BELIAKOV



*Back to the table
of contents*



1

2

3

4

5

6

7

8

9

- 1) Introduction to DSS and integration:
 - About DSS;
 - Integration of DSS to a Maven project.
- 2) Signature creation with DSS:
 - 3 atomic methods;
 - parameters configuration.
- 3) Signature augmentation:
 - TSPSource configuration;
 - Revocation data access.
- 4) Signature validation:
 - CertificateVerifier configuration;
 - XML Validation policy.

Part 1 : Introduction to DSS and integration

- Introduction;
- DSS structure;
- Requirements;
- Integration of DSS to a Maven project.



1

2

3

4

5

6

7

8

9

The DSS (Digital Signature Service) project is an open-source software library, aimed at providing implementation of the standards for Advanced Electronic Signature creation, augmentation and validation in line with European legislation and the eIDAS Regulation in particular.

This project is available in **Java** language.



1

2

3

4

5

6

7

8

9

DSS consists of:

- **DSS core:** <https://github.com/esig/dss> - the main repository containing code of the framework (*distributed under **LGPL-2.1 license***);
- **DSS demonstrations:** <https://github.com/esig/dss-demonstrations> - repository containing integration examples of the framework. It includes Spring Web Application, JavaFX standalone application, etc. (*distributed under **LGPL-2.1 license** and **EUPL-1.1** for MOCCA module*).

[1](#)[2](#)[3](#)[4](#)[5](#)[6](#)[7](#)[8](#)[9](#)

Requirements for DSS Core:

- **Java 8 and higher** (tested up to Java 19) for the usage is required. For build Java 11 is a minimum requirement;
- **Maven 3.6 and higher**;
- **Memory and Disk:** see minimal requirements for the used JVM. We recommend at least 2 GB memory;
- **Operating system:** no specific requirements (tested on Windows, Linux, MacOS).



1

2

3

4

5

6

7

8

9

Access:

Starting from version 5.11.1 the artifacts of DSS are available at **Maven Central**.

See the [official Maven repository](#).



1

2

3

4

5

6

7

8

9

Integration of DSS to a Maven project.

Include DSS modules to the project:

1) include dss-bom to simplify version management:

```
<dependencyManagement>
  <dependencies>
    <dependency>
      <groupId>eu.europa.ec.joinup.sd-dss</groupId>
      <artifactId>dss-bom</artifactId>
      <version>5.11.1</version>
      <type>pom</type>
      <scope>import</scope>
    </dependency>
  </dependencies>
</dependencyManagement>
```





1

2

3

4

5

6

7

8

9

Integration of DSS to a Maven project.

Include DSS modules to the project:

- 1) include dss-bom to simplify version management;
- 2) include required modules*:

```
<dependencies>
  <dependency>
    <groupId>eu.europa.ec.joinup.sd-dss</groupId>
    <artifactId>dss-utils-apache-commons</artifactId>
  </dependency>
  <dependency>
    <groupId>eu.europa.ec.joinup.sd-dss</groupId>
    <artifactId>dss-pades-pdfbox</artifactId>
  </dependency>
  ...
</dependencies>
```

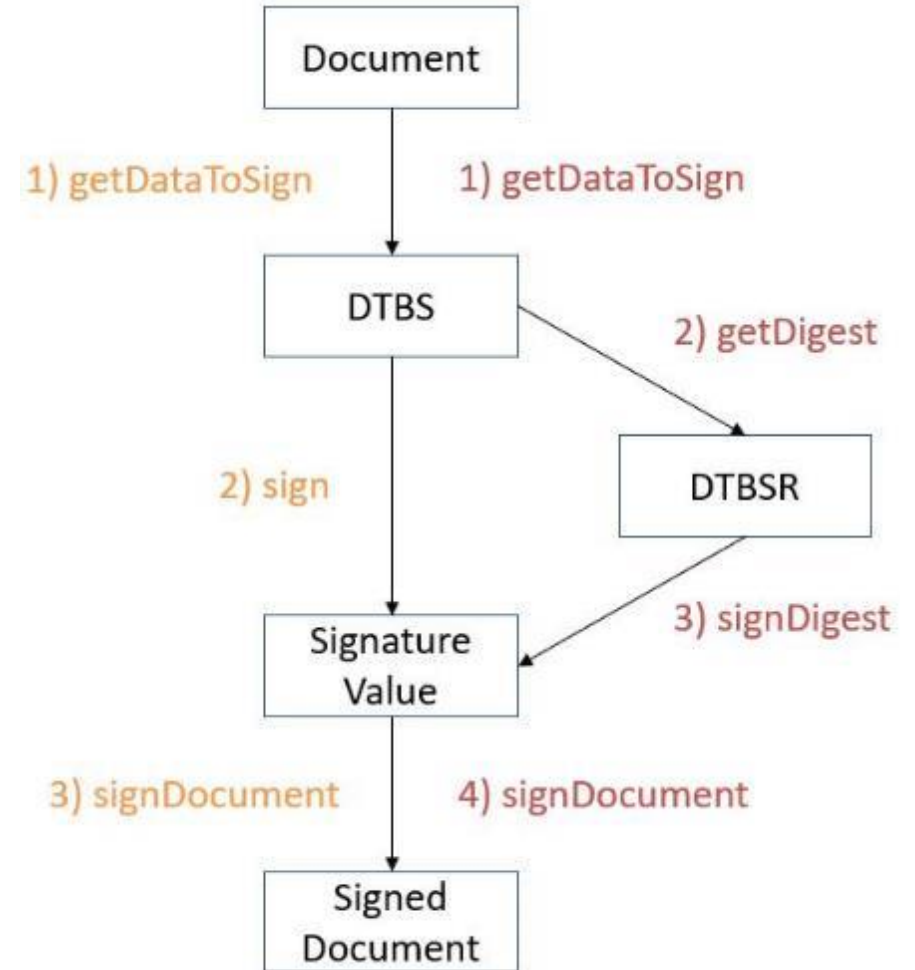
* See list of the available modules by the [link](#)

Part 2 : Signature creation with DSS

- signature creation in 3 stateless methods;
- signature service;
- signature creation parameters.

Signature creation in DSS is done in **3 (or 4) atomic steps**.

This provides **implementation-independent** signature creation with support of different signing architectures (client-sign, server-sign, etc.).





This example demonstrates the use of the **PAdESService** for a PAdES signature creation:

```
// CertificateVerifier provides a configuration of trust anchors, revocation sources, AIA, etc.  
CommonCertificateVerifier commonCertificateVerifier = new CommonCertificateVerifier();  
// configure CertificateVerifier
```

```
// Create PAdESService for signature  
PAdESService service = new PAdESService(commonCertificateVerifier);
```

```
// Get the DataToBeSigned  
ToBeSigned dataToSign = service.getDataToSign(toSignDocument, parameters);
```

← 1) DTBS

```
// This function obtains the signature value for signed information using the  
// private key and specified algorithm  
DigestAlgorithm digestAlgorithm = parameters.getDigestAlgorithm();  
SignatureValue signatureValue = signingToken.sign(dataToSign, digestAlgorithm, privateKey);
```

← 2) Signature Value

```
// Create the signed document  
DSSDocument signedDocument = service.signDocument(toSignDocument, parameters, signatureValue);
```

← 3) Signed Document

Both methods have a set of common parameters:

```
ToBeSigned dataToSign = service.getDataToSign(toSignDocument, parameters);
```



```
DSSDocument signedDocument = service.signDocument(toSignDocument, parameters, signatureValue);
```

- DSSDocument
- SerializableSignatureParameters (PAdESSignatureParameters for PAdES).



1

2

3

4

5

6

7

8

9

DSSDocument is a representation of a document in DSS.

The following implementations are provided:

- **InMemoryDocument** : fully loads the document in memory. This type of DSSDocument can be instantiated with an array of bytes or an InputStream.
- **FileDocument** : created from an existing local file.
- **DigestDocument** : only contains pre-computed digest values for a given document. That allows a user to avoid sending the full document (detached signatures). *Not possible with PAdES!*
- *and some other (for specific formats) ...*



1

2

3

4

5

6

7

8

9

PAdESSignatureParameters is a configuration allowing customization of the creating signature.

It has the following required parameters:

- **setSigningCertificate** – sets the signing-certificate used to create the signature;
- **setCertificateChain** – sets the certificate chain of the signing-certificate;
- **setSignatureLevel** – sets the target signature level (e.g. PAdES-BASELINE-B);
- **bLevel** – contains a set of common signed parameters across various signature formats.

Note: implementations for other formats work similarly.



1

2

3

4

5

6

7

8

9

Example of `DSSDocument` and `PAdESSignatureParameters` configuration:

```
DSSDocument toSignDocument = new FileDocument("path_to_document/sample.pdf");
```

```
PAdESSignatureParameters parameters = new PAdESSignatureParameters();  
parameters.setSigningCertificate(*signingCertificate*);  
parameters.setCertificateChain(*certificateChain*);  
parameters.setSignatureLevel(SignatureLevel.PAdES_BASELINE_B);  
parameters.bLevel().setSigningDate(new Date());
```

```
ToBeSigned dataToSign = service.getDataToSign(toSignDocument, parameters);
```

...

```
DSSDocument signedDocument = service.signDocument(toSignDocument, parameters, signatureValue);
```

Note: ***the same** document and parameters shall be provided **to both methods**.*



1

2

3

4

5

6

7

8

9

Complete signature creation sample using a **SignatureTokenConnection** (PKCS#12 key store):

```
try (Pkcs12SignatureToken token = new Pkcs12SignatureToken("src/main/resources/user_a_rsa.p12",
    new PasswordProtection("password".toCharArray())) {
    // the document to be signed
    DSSDocument toSignDocument = new FileDocument("path_to_document/sample.pdf");

    // Extract the key for signature creation
    DSSPrivateKeyEntry privateKey = signingToken.getKeys().get(0);

    // Preparing parameters for the PAdES signature
    PAdESSignatureParameters parameters = new PAdESSignatureParameters();
    parameters.setSignatureLevel(SignatureLevel.PAdES_BASELINE_B);
    parameters.setDigestAlgorithm(DigestAlgorithm.SHA256);

    // Extract the signing-certificate
    parameters.setSigningCertificate(privateKey.getCertificate());
    // Extract the certificate chain
    parameters.setCertificateChain(privateKey.getCertificateChain());

    // Create common certificate verifier
    CommonCertificateVerifier commonCertificateVerifier = new CommonCertificateVerifier();
    // Create PAdESService for signature
    PAdESService service = new PAdESService(commonCertificateVerifier);

    // Get the SignedInfo segment that need to be signed.
    ToBeSigned dataToSign = service.getDataToSign(toSignDocument, parameters);

    // This function obtains the signature value for signed information using the
    // private key and specified algorithm
    DigestAlgorithm digestAlgorithm = parameters.getDigestAlgorithm();
    SignatureValue signatureValue = signingToken.sign(dataToSign, digestAlgorithm, privateKey);

    // We invoke the padesService to sign the document with the signature value obtained in
    // the previous step.
    DSSDocument signedDocument = service.signDocument(toSignDocument, parameters, signatureValue);
}
```



1

2

3

4

5

6

7

8

9

Part 3 : Signature extension

- signature extension method;
- configuration of a timestamp service;
- configuration of CRL/OCSP sources;
- configuration of trusted certificate source.

[1](#)[2](#)[3](#)[4](#)[5](#)[6](#)[7](#)[8](#)[9](#)

Augmentation of a signature to a higher level is done with `DocumentSignatureService.extendDocument(...)` method.

```
// Init service for signature augmentation
PAdESService padesService = new PAdESService(certificateVerifier);
...

// Init parameters and identify the target signature level
PAdESSignatureParameters parameters = new PAdESSignatureParameters();
parameters.setSignatureLevel(SignatureLevel.PAdES_BASELINE_T);

// extend the created earlier signature
DSSDocument tLevelDocument = padesService.extendDocument(signedDocument, parameters);
```



1

2

3

4

5

6

7

8

9

To extend a signature to **PAdES-BASELINE-T** level, a **TSPSource** should be provided.

1) Create a **TSPSource** with a link to the TSP server.

```
OnlineTSPSource tspSource = new OnlineTSPSource("http://dss.nowina.lu/pki-factory/tsa/good-tsa");
```

2) Provide authentication credentials:

```
TimestampDataLoader dataLoader = new TimestampDataLoader();  
dataLoader.addAuthentication("nowina.lu", 80, "https", "login", "password");  
tspSource.setDataLoader(dataLoader);
```

3) Set the **TSPSource** to the used signature creation service (i.e. to **PAdESService**):

```
PAdESService service = new PAdESService(commonCertificateVerifier);  
service.setTspSource(tspSource);
```



1

2

3

4

5

6

7

8

9

To extend a signature to **PADES-BASELINE-LT** level, the following should be configured:

- **CRL and/or OCSP sources** – to provide the means to access revocation data;
- **Trusted certificate source** – to ensure the revocation request is performed only to trusted sources.



1

2

3

4

5

6

7

8

9

The way revocation data is retrieved should be defined with an instance of **CRLSource** and/or **OCSPSource**.

The following implementations are provided:

OnlineCRLSource/OnlineOCSPSource – retrieves CRL/OCSP from the remote source using the given URL address;

JdbcCacheCRLSource/JdbcCacheOCSPSource – retrieves CRL/OCSP from the JDBC database (and stores retrieved values in the database).



1

2

3

4

5

6

7

8

9

To configure the trusted sources, an instance of **CommonTrustedCertificateSource** shall be provided to the **CertificateVerifier**.

To configure trusted certificate Source, the certificates can be added manually:

```
CertificateSource trustedCertSource = new CommonTrustedCertificateSource();  
trustedCertSource.addCertificate(rootCertificateOne);  
trustedCertSource.addCertificate(rootCertificateTwo);  
...
```

From other certificate source, e.g. using a keystore:

```
KeyStoreCertificateSource keystore = new KeyStoreCertificateSource(new File("src/main/resources/keystore.p12"),  
    "PKCS12", getPassword());  
trustedCertSource.importAsTrusted(keystore);
```

Or automatically from the EU LOTL using **TLValidationJob** (see [11.1. Configuration of TL validation job](#)).



1

2

3

4

5

6

7

8

9

PAdES-BASELINE-LT signature augmentation example:

```
PAdESSignatureParameters parameters = new PAdESSignatureParameters();  
parameters.setSignatureLevel(SignatureLevel.PAdES_BASELINE_LT);  
  
CommonCertificateVerifier certificateVerifier = new CommonCertificateVerifier();  
// init revocation sources for CRL/OCSP requesting  
certificateVerifier.setCrlSource(new OnlineCRLSource());  
certificateVerifier.setOcspSource(new OnlineOCSPSource());  
  
// Trust anchors should be defined for revocation data requesting  
certificateVerifier.setTrustedCertSources(getTrustedCertificateSource());  
  
// Init service for signature augmentation  
PAdESService padesService = new PAdESService(certificateVerifier);  
  
// extend the created earlier signature  
DSSDocument ltLevelDocument = padesService.extendDocument(tLevelDocument, parameters);
```

Extension to **PADES-BASELINE-LTA** level requires:

- TSPSource;
- CRL and/or OCSP sources;
- Trusted certificates source.



1

2

3

4

5

6

7

8

9

PAdES-BASELINE-LTA signature augmentation example:

```
PAdESSignatureParameters parameters = new PAdESSignatureParameters();
parameters.setSignatureLevel(SignatureLevel.PAdES_BASELINE_LTA);

CommonCertificateVerifier certificateVerifier = new CommonCertificateVerifier();
// init revocation sources for CRL/OCSP requesting
certificateVerifier.setCrlSource(new OnlineCRLSource());
certificateVerifier.setOcspSource(new OnlineOCSPSource());

// Trust anchors should be defined for revocation data requesting
certificateVerifier.setTrustedCertSources(getTrustedCertificateSource());

// Init service for signature augmentation
PAdESService padesService = new PAdESService(certificateVerifier);

// init TSP source for timestamp requesting
padesService.setTspSource(getOnlineTSPSource());

// extend the created earlier signature
DSSDocument ltaLevelDocument = padesService.extendDocument(signedDocument, parameters);
```




1

2

3

4

5

6

7

8

9

Part 4 : Signature validation

- document validator;
- certificate verifier configuration;
- validation policy.



1

2

3

4

5

6

7

8

9

Signature validation can be performed using **SignedDocumentValidator** independently of the document format:

```
// SignedDocumentValidator will automatically select the available validator able to process the document format
```

```
DocumentValidator documentValidator = SignedDocumentValidator.fromDocument(document);
```

```
// Certificate Verifier provides a configuration to perform the validation process
```

```
documentValidator.setCertificateVerifier(cv);
```

```
// Execute the validation
```

```
Reports reports = documentValidator.validateDocument();
```

```
// We have 4 reports
```

```
// The diagnostic data which contains all used and static data
```

```
DiagnosticData diagnosticData = reports.getDiagnosticData();
```

```
// The detailed report which is the result of the process of the diagnostic data and the validation policy
```

```
DetailedReport detailedReport = reports.getDetailedReport();
```

```
// The simple report is a summary of the detailed report (more user-friendly)
```

```
SimpleReport simpleReport = reports.getSimpleReport();
```

```
// The JAXB representation of the ETSI Validation report (ETSI TS 119 102-2)
```

```
ValidationReportType etsiValidationReport = reports.getEtsiValidationReportJaxb();
```

CertificateVerifier provides configuration for:

- AIA certificates source;
- CRL and/or OCSP sources;
- Trusted certificates source;
- Adjunct certificate sources;
- Behavior configuration through alerts (e.g. exception, log, etc.);
- other settings.



1

2

3

4

5

6

7

8

9

CertificateVerifier configuration:

```
// Instantiate the Certificate Verifier
CertificateVerifier cv = new CommonCertificateVerifier();

// We can inject several sources. eg: OCSP, CRL, AIA, trusted lists

// Capability to download resources from AIA
cv.setAIASource(new DefaultAIASource());

// Capability to request OCSP Responders
cv.setOcspSource(new OnlineOCSPSource());

// Capability to download CRL
cv.setCrlSource(new OnlineCRLSource());

// Create an instance of a trusted certificate source
CommonTrustedCertificateSource trustedCertSource = new CommonTrustedCertificateSource();
// import the keystore as trusted
trustedCertSource.importAsTrusted(keystoreCertSource);

// Add trust anchors (trusted list, keystore,...) to a list of trusted certificate sources
cv.addTrustedCertSources(trustedCertSource);

// Additionally add missing certificates to a list of adjunct certificate sources (not trusted certificates)
cv.addAdjunctCertSources(adjunctCertSource);
```

For more information about
CertificateVerifier please see
[7.1.4. CertificateVerifier configuration](#)



1

2

3

4

5

6

7

8

9

It is possible to customize validation process using an XML Validation Policy:

Every constraint defines a behavior in case failure:

- **FAIL** – the check will interrupt validation;
- **WARN** – a warning will be returned;
- **INFO** – information messages to be returned;
- **IGNORE** – skip check.

```
...
<SigningCertificate>
  <Recognition Level="FAIL" />
  <Signature Level="FAIL" />
  <NotExpired Level="FAIL" />
  <AuthorityInfoAccessPresent Level="WARN" />
  <RevocationInfoAccessPresent Level="WARN" />
  <RevocationDataAvailable Level="FAIL" />
  <AcceptableRevocationDataFound Level="FAIL" />
  <CRLNextUpdatePresent Level="WARN" />
  <RevocationFreshness Level="IGNORE" Unit="DAYS" Value="0" />
  <KeyUsage Level="WARN">
    <Id>nonRepudiation</Id>
  </KeyUsage>
  <PolicyTree Level="WARN" />
  <NameConstraints Level="WARN" />
...
</SigningCertificate>
...
```

[1](#)[2](#)[3](#)[4](#)[5](#)[6](#)[7](#)[8](#)[9](#)

Customized XML Validation Policy must be provided to the **DocumentValidator**:

```
// SignedDocumentValidator will automatically select the available validator able to process the document format  
SignedDocumentValidator documentValidator = SignedDocumentValidator.fromDocument(document);
```

...

```
// Load custom XML Validation Policy file  
File customPolicy = new File("/path/to/validation/policy.xml");  
Reports reports = validator.validateDocument(customPolicy);
```

Information about validation policy constraints can be found at [7.2. AdES validation constraints/policy](#)



1

2

3

4

5

6

7

8

9

- 1) Introduction to DSS and integration:
 - About DSS;
 - Integration of DSS to a Maven project.
- 2) Signature creation with DSS:
 - 3 atomic methods;
 - parameters configuration.
- 3) Signature augmentation:
 - TSPSource configuration;
 - Revocation data access.
- 4) Signature validation:
 - CertificateVerifier configuration;
 - XML Validation policy.



Break

10'



*Back to the table
of contents*

8

Specifications of the new DSS version

Aleksandr BELIAKOV



| M | T | W | T | F | S | S |
|----|----|----|----|----|----|----|
| | | | | | | 1 |
| 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| 16 | 17 | 18 | 19 | 20 | 21 | 22 |
| 23 | 24 | 25 | 26 | 27 | 28 | 29 |
| 30 | 31 | | | | | |

[Back to the table of contents](#)



1

2

3

4

5

6

7

8

9





- 1) DSS 5.12 version:
 - Available on Maven Central;
 - new features.
- 2) Migration to the new version:
 - common difficulties;
 - code changes.
- 3) Future versions of DSS:
 - migration to Jakarta.

DSS 5.12 version is available
at Maven Central:

[Home](#) » [eu.europa.ec.joinup](#) » [sd-dss](#)

Group: Europa EC Joinup SD DSS

Sort: [popular](#) | [newest](#)

- | | | |
|---|--|-------------------|
|  | 1. DSS Test eu.europa.ec.joinup.sd-dss » dss-test DSS Test contains useful classes to do test. Last Release on Feb 28, 2023 | 25 usages LGPL |
|  | 2. DSS Document eu.europa.ec.joinup.sd-dss » dss-document DSS Document contains the code for the creation and validation of XAdES, CADES, PADES and ASiC signatures. Last Release on Feb 28, 2023 | 14 usages LGPL |
|  | 3. DSS Service Provider Interface eu.europa.ec.joinup.sd-dss » dss-spi DSS Service Provider Interface contains the contract interface shared between the applet and the server-side of DSS. Last Release on Feb 28, 2023 | 13 usages LGPL |
|  | 4. DSS JAXB Parsers eu.europa.ec.joinup.sd-dss » dss-jaxb-parsers DSS JAXB Parsers contains parsers to parse/print enumerations from/to XML Last Release on Feb 28, 2023 | 12 usages LGPL |



1

2

3

4

5

6

7

8

9

New features available at DSS 5.12:

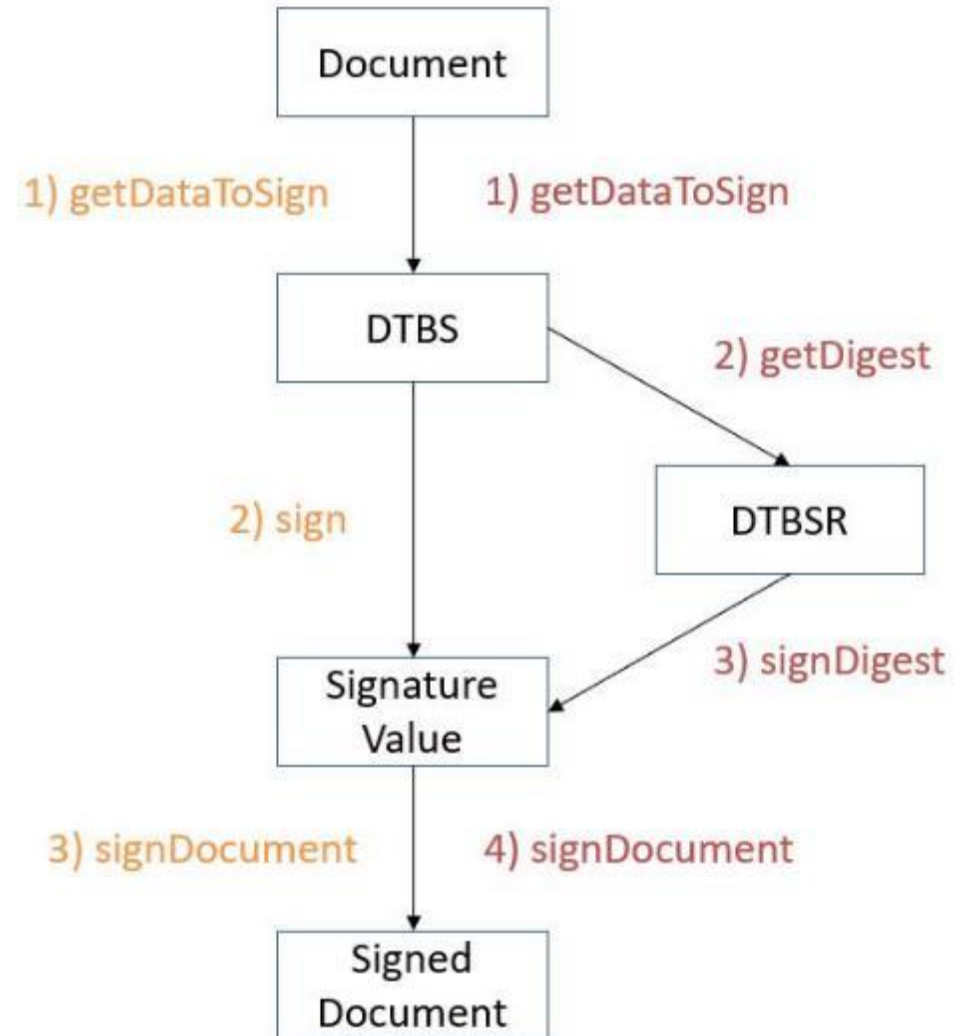
- PAdES:
 - signature creation with external CMS provider;
 - PDF/A validation support;
 - optional VRI dictionary creation;
 - improved security and performance.
- Validation process:
 - improved conformance to RFC 5280, ETSI EN 319 102-1;
 - improved handling of expired cryptographic algorithms.
- REST/SOAP webservices:
 - added a setter of default validation policy;
 - added a signing method with provided Signature Algorithm.
- DSS standalone:
 - signing of multiple documents, extension of documents, validation of documents.
- Dependencies update + bug fixes;
- Java 19 support;
- and others ...

Common DSS signing workflow:

In PAdES methods

1) getDataToSign and **3) signDocument**
generate:

- PDF document revision;
- CMS signed data.





1

2

3

4

5

6

7

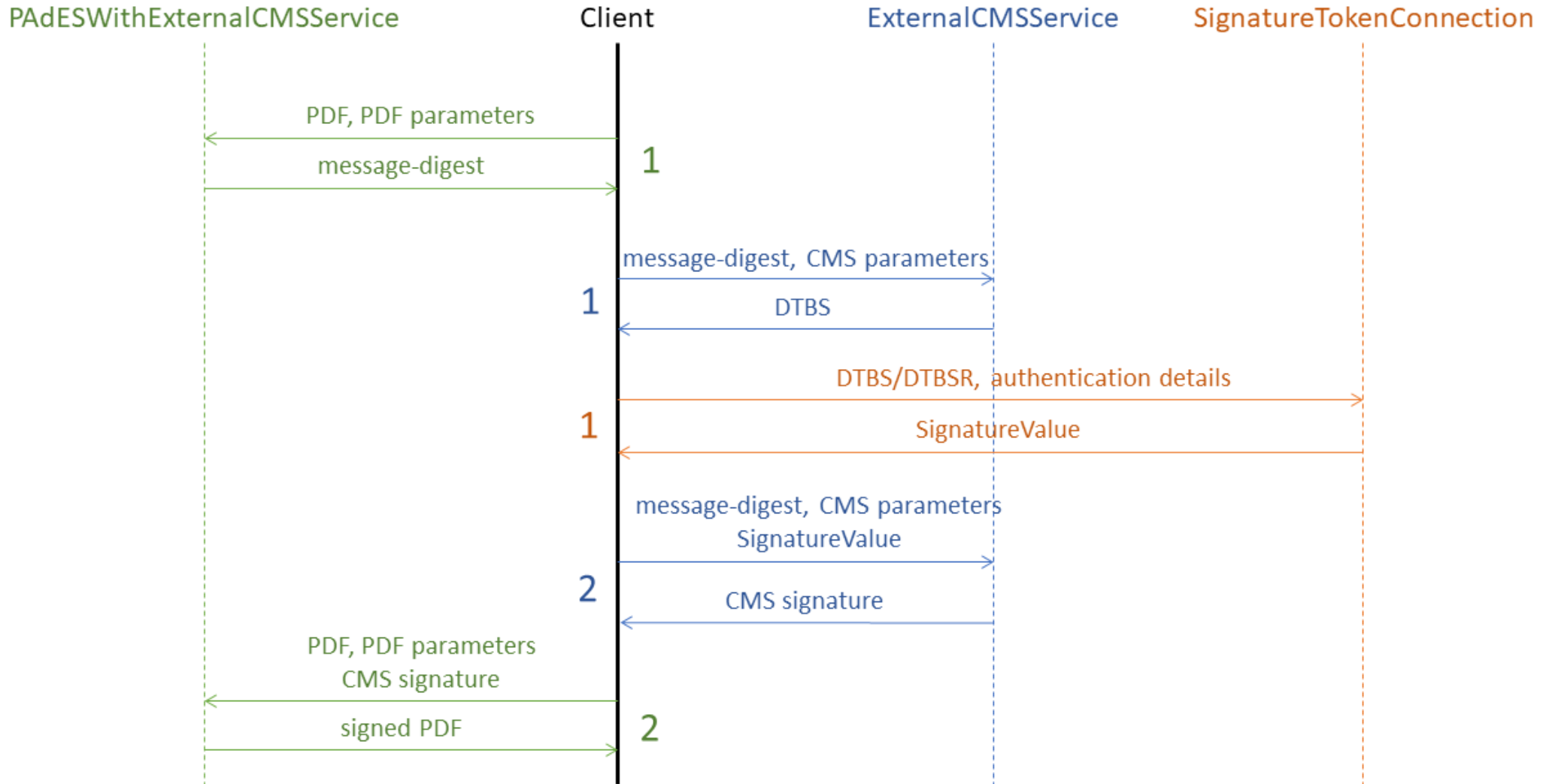
8

9

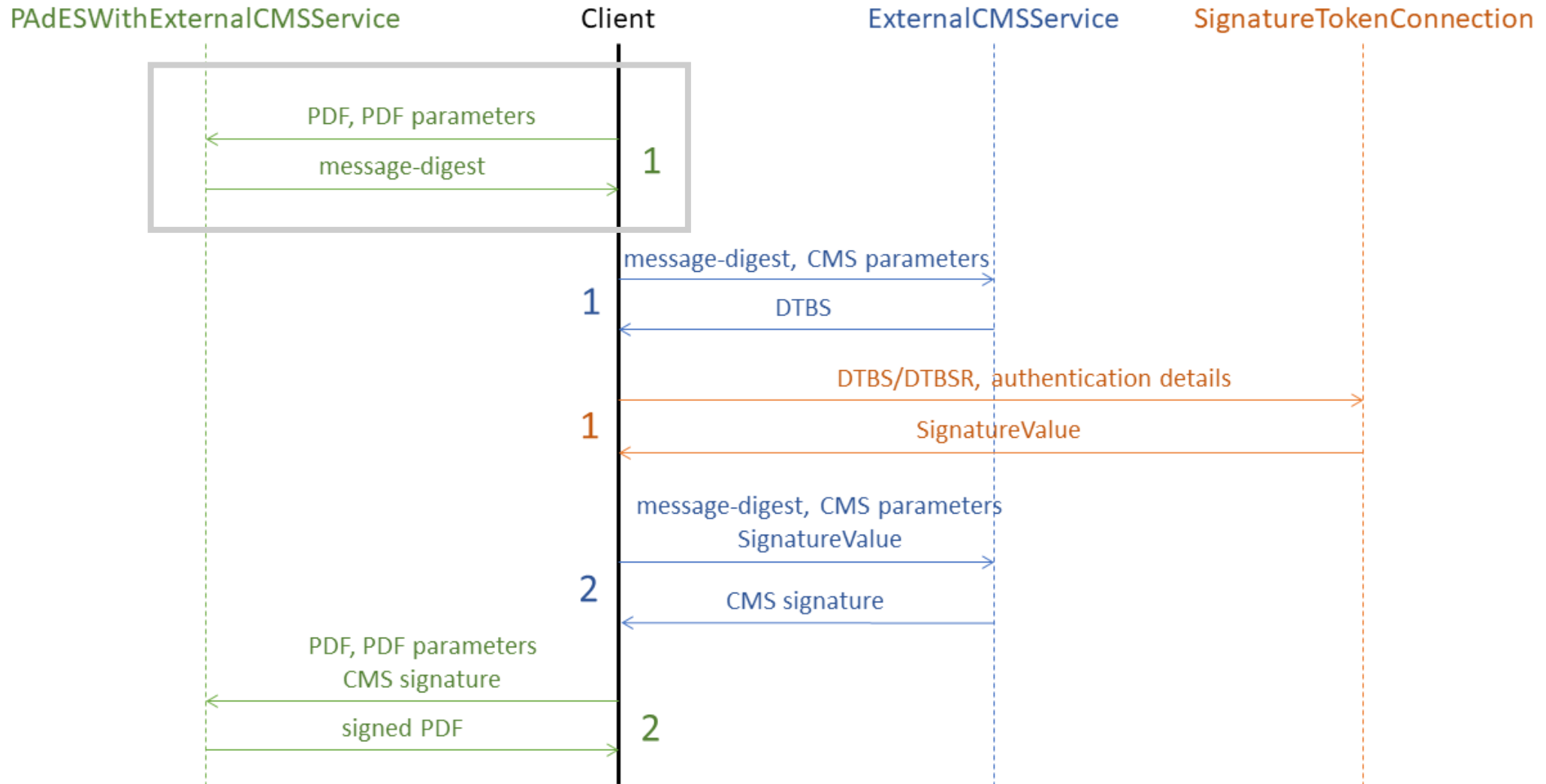
DSS 5.12 provides a possibility to delegate one or another part of PAdES signature creation to external service:

- **PAdESWithExternalCMSProvider** – prepares a PDF document for signing and builds a message-digest of the signed Byte Range;
- **ExternalCMSService** – builds a CMS to be incorporated within a PDF document.

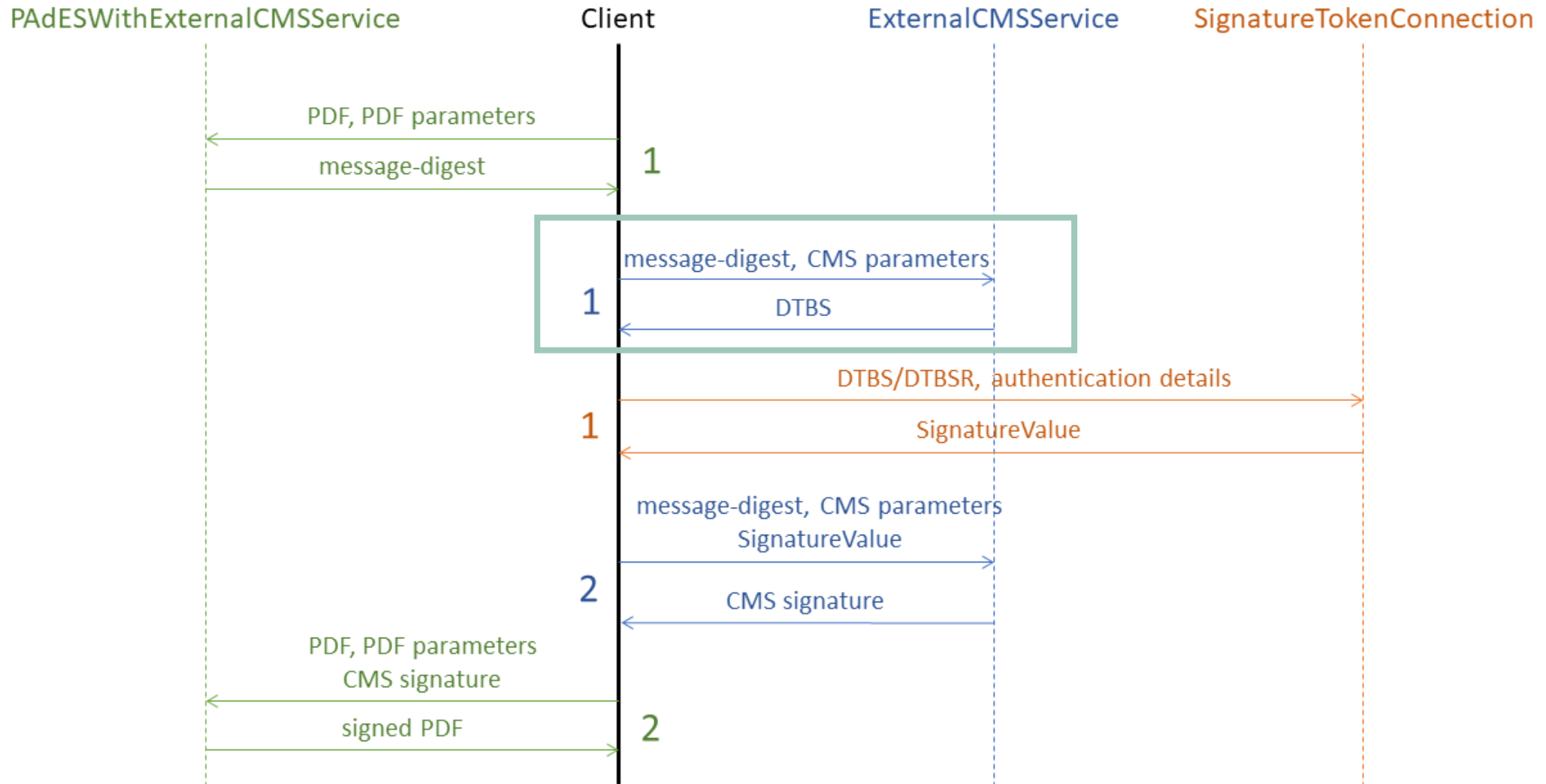
PAdES creation with external CMS provider



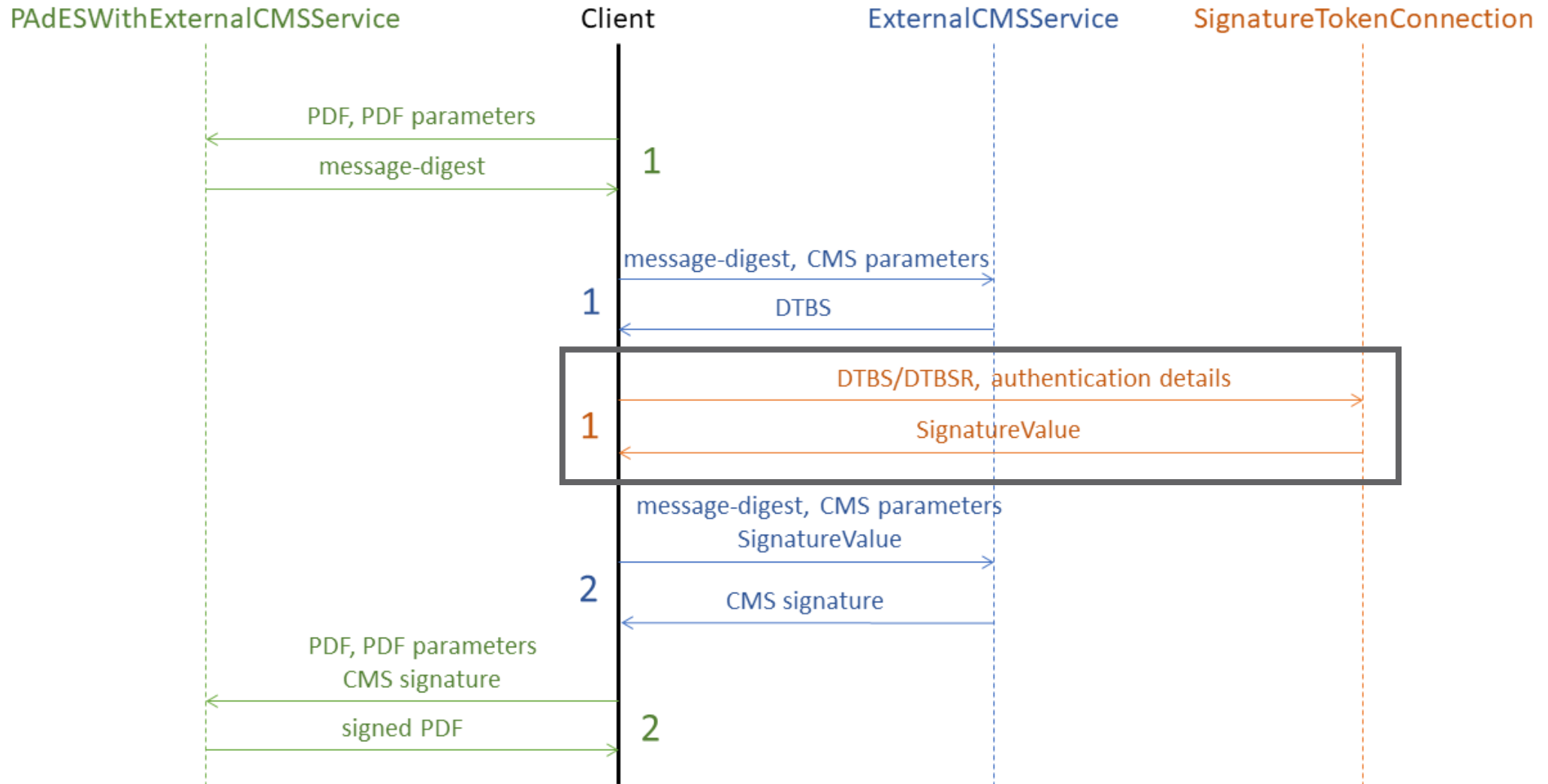
PAdES creation with external CMS provider



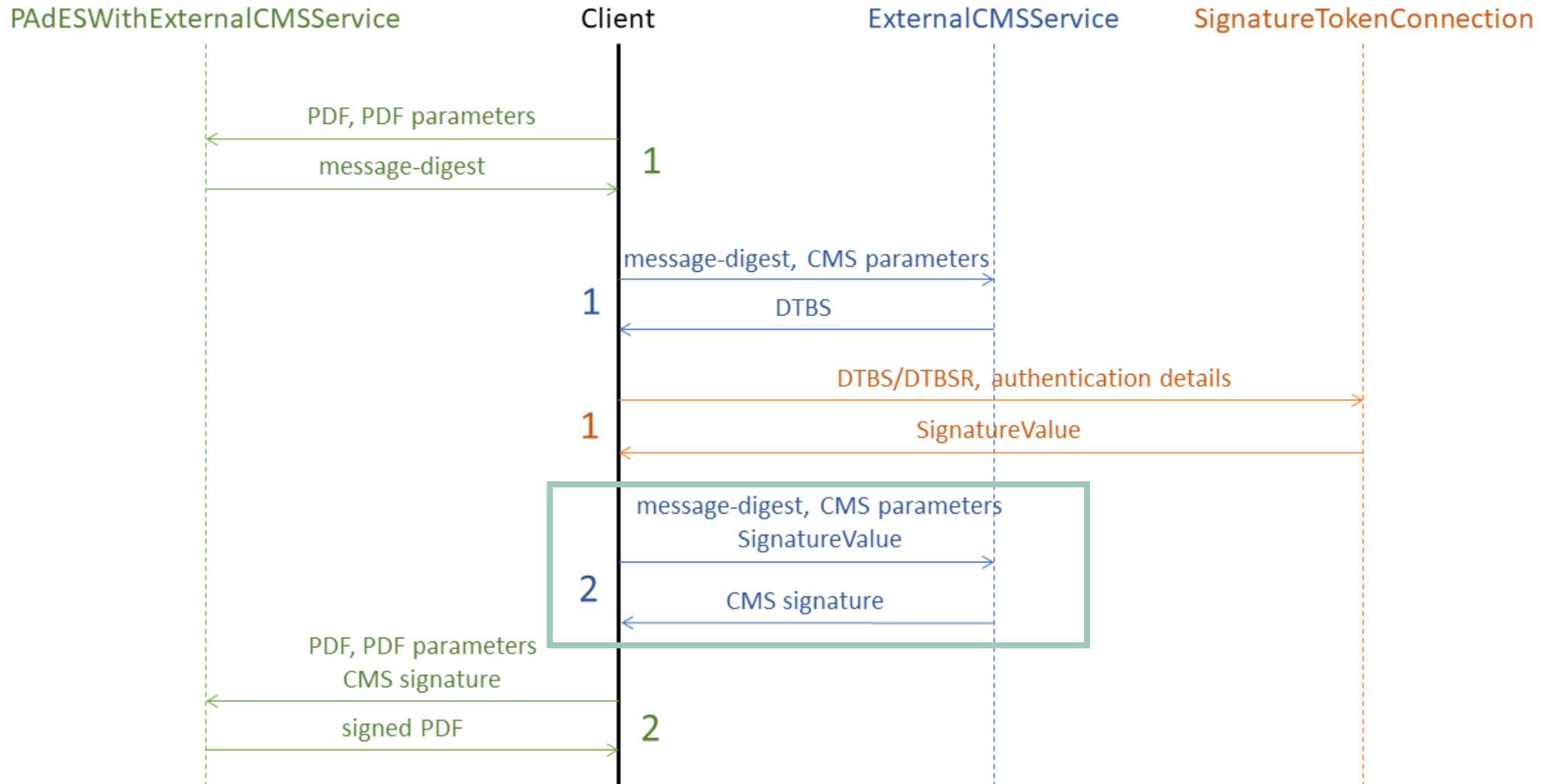
PAdES creation with external CMS provider



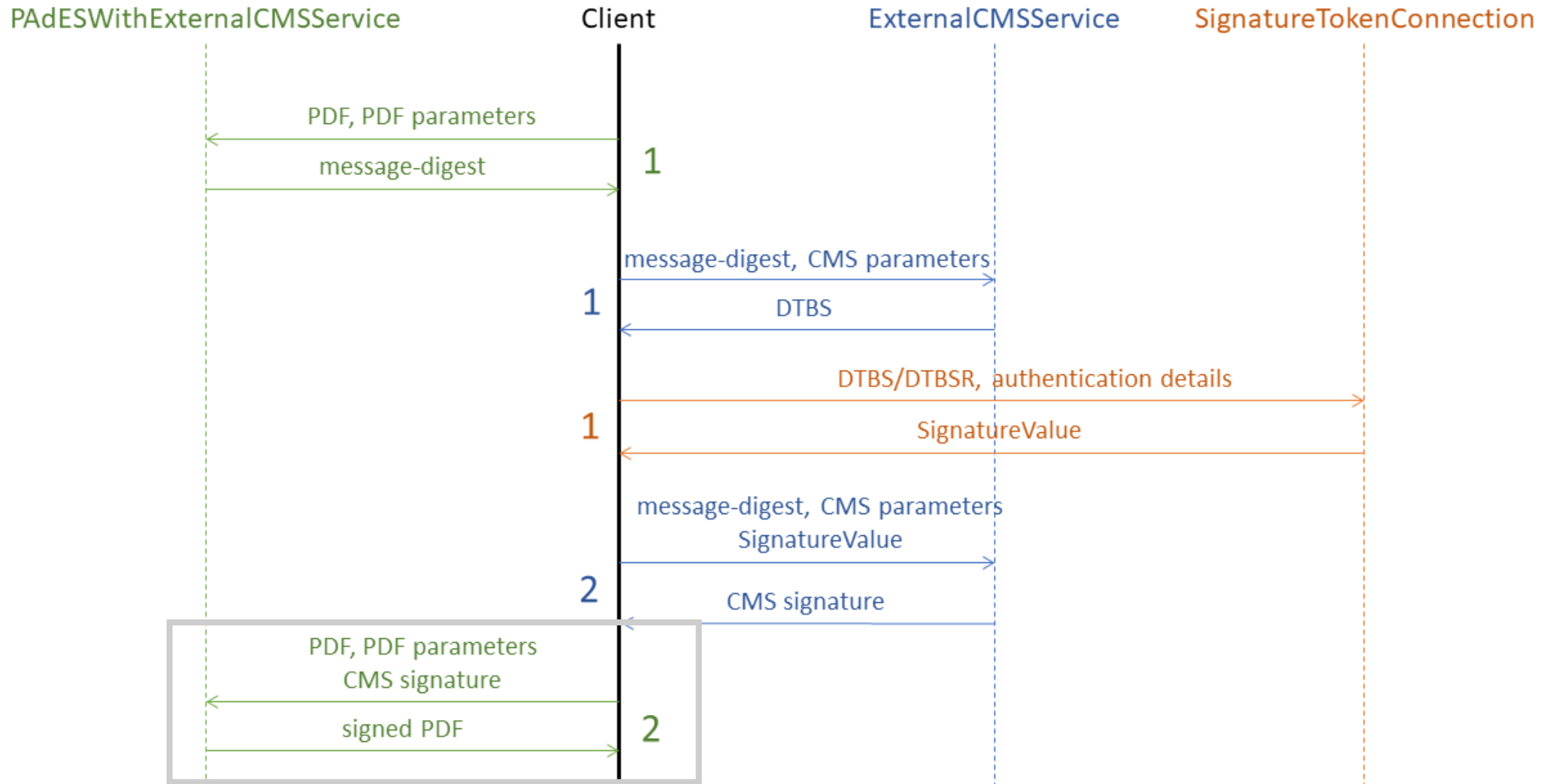
PAdES creation with external CMS provider



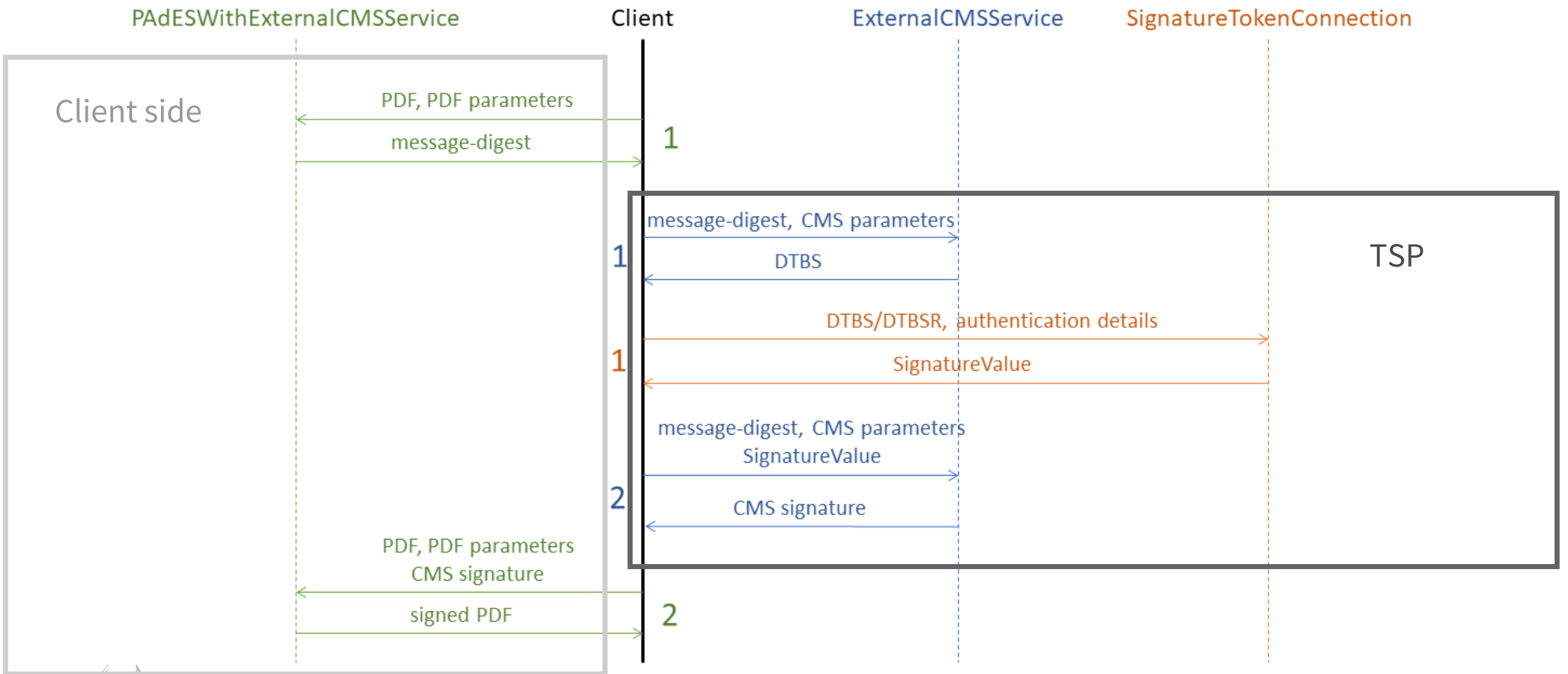
PAdES creation with external CMS provider



PAdES creation with external CMS provider



PAdES creation with external CMS provider





1

2

3

4

5

6

7

8

9

Example of PAdES creation using external CMS provider:

```
// Instantiate PDF signature service with external CMS
PAdESWithExternalCMSService service = new PAdESWithExternalCMSService();

// Configure signature parameters
PAdESSignatureParameters signatureParameters = new PAdESSignatureParameters();
signatureParameters.setSignatureLevel(SignatureLevel.PAdES_BASELINE_B);
...

// Prepare the PDF signature revision and compute message-digest of the byte range content
DSSMessageDigest messageDigest = service.getMessageDigest(toSignDocument, signatureParameters);

// Obtain CMS signature from external CMS signature provider
DSSDocument cmsSignature = getExternalCMSSignature(messageDigest);

// Embed the obtained CMS signature to a PDF document with prepared signature revision
DSSDocument signedDocument = service.signDocument(toSignDocument, signatureParameters, cmsSignature);
```

[1](#)[2](#)[3](#)[4](#)[5](#)[6](#)[7](#)[8](#)[9](#)

DSS 5.12 provides a possibility to verify PDF document on conformance to PDF/A specification.

The verification process is conducted by a [VeraPDF](#) library.

To include PDF/A validation the following steps are required:

1) Add dss-pdf/a module to the pom.xml file of your Maven project:

```
<dependency>  
  <groupId>eu.europa.ec.joinup.sd-dss</groupId>  
  <artifactId>dss-pdf/a</artifactId>  
  <version>${dss.version}</version>  
</dependency>
```

DSS 5.12 provides a possibility to verify PDF document on conformance to **PDF/A** specification.

The verification process is conducted by a [VeraPDF](#) library.

To include PDF/A validation the following steps are required:

2) Create `/META-INF/services/eu.europa.esig.dss.validation.DocumentValidatorFactory` file within resources folder of your project with the following content:

```
eu.europa.esig.dss.pdfa.validation.PDFADocumentValidatorFactory
```


[1](#)[2](#)[3](#)[4](#)[5](#)[6](#)[7](#)[8](#)[9](#)

Information about PDF/A validation can be extracted from validation report:

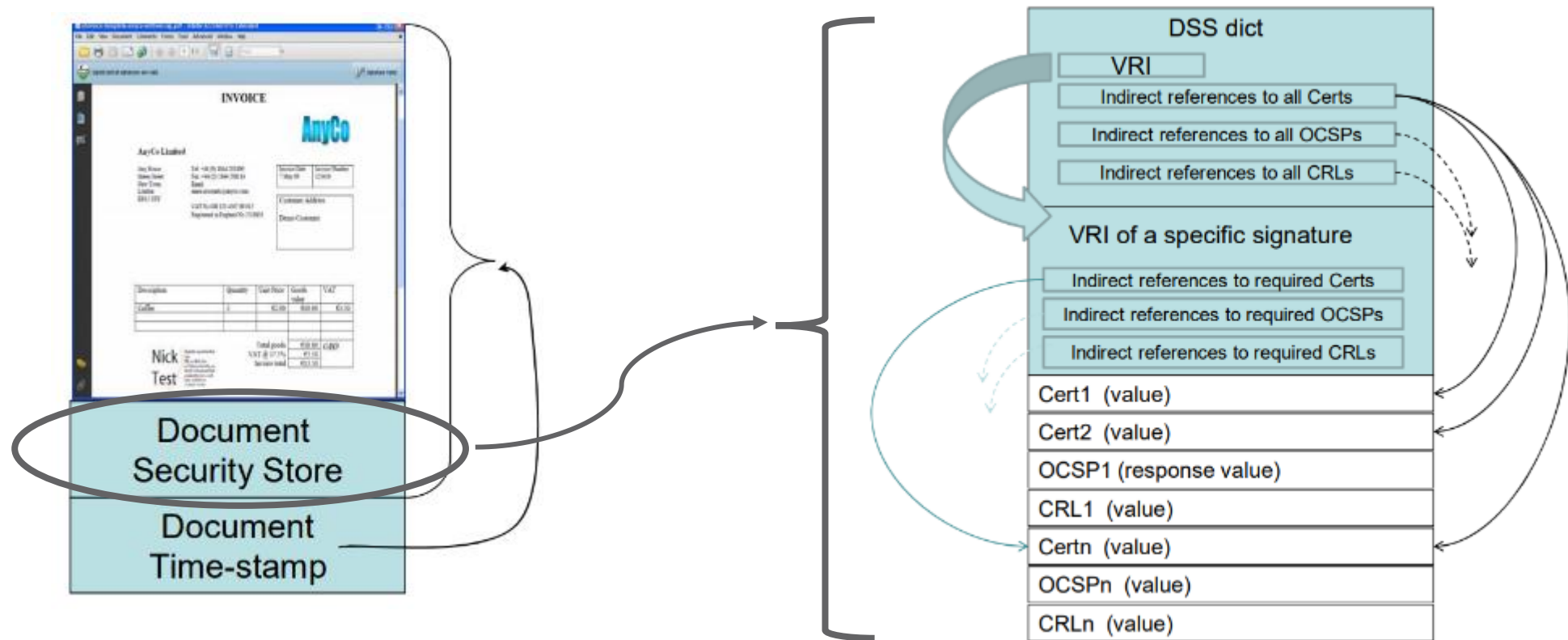
```
// Create a DocumentValidator to perform validation against PDF/A specification
DocumentValidator documentValidator = SignedDocumentValidator.fromDocument(pdfDocument);

// Execute validation
Reports reports = documentValidator.validateDocument();

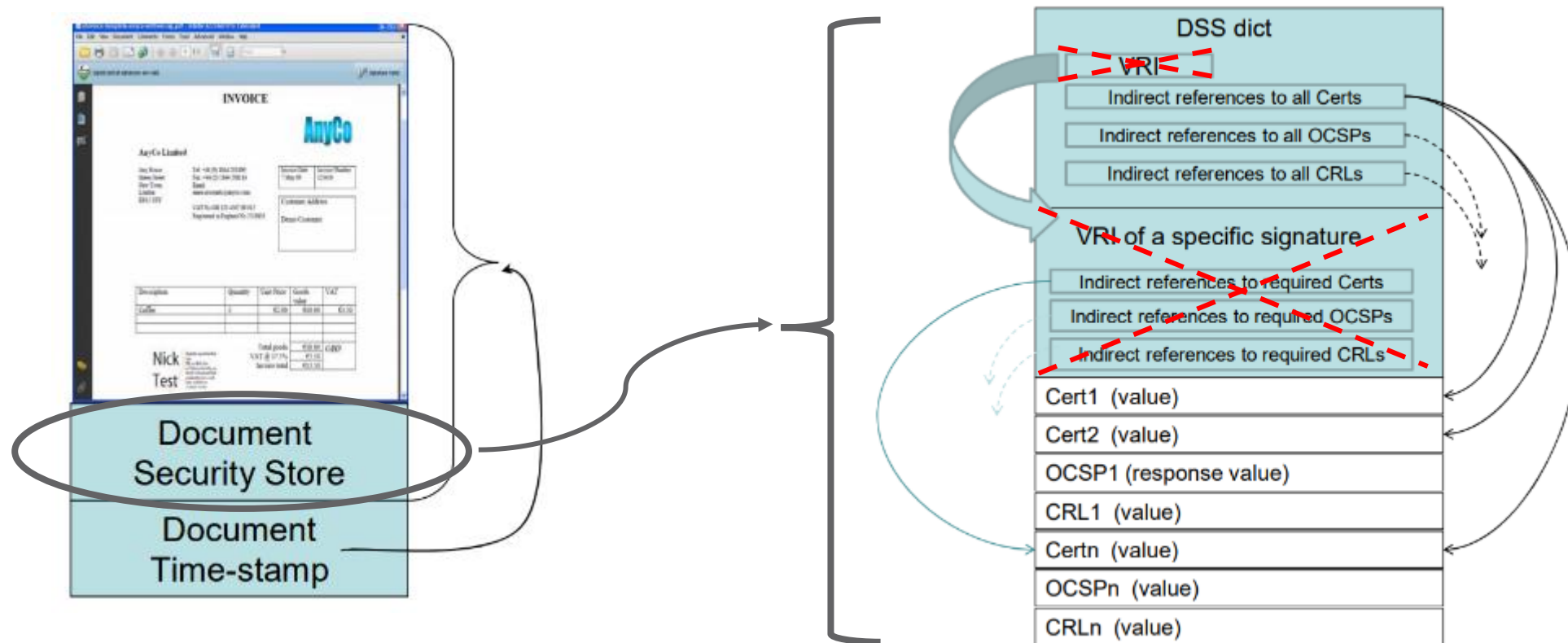
// Extract the interested information from DiagnosticData
DiagnosticData diagnosticData = reports.getDiagnosticData();

// This variable contains the name of the identified PDF/A profile (or closest if validation failed)
String profileId = diagnosticData.getPDFAProfileId();
// Checks whether the PDF document is compliant to the identified PDF profile
boolean compliant = diagnosticData.isPDFACompliant();
// Returns the error messages occurred during the PDF/A verification
Collection<String> errorMessages = diagnosticData.getPDFAVValidationErrors();
```

In DSS 5.11 and before a VRI (Validation Related Information) dictionary has been created together with DSS (Document Security Store):



In DSS 5.12 the VRI dictionary is optional and not included by default:



[1](#)[2](#)[3](#)[4](#)[5](#)[6](#)[7](#)[8](#)[9](#)

In **DSS 5.12** the VRI dictionary can be included through **PAdESSignatureParameters**:

```
PAdESSignatureParameters signatureParameters = new PAdESSignatureParameters();  
signatureParameters.setSignatureLevel(SignatureLevel.PAdES_BASELINE_LT);  
...  
signatureParameters.setIncludeVRIDictionary(true);
```

[1](#)[2](#)[3](#)[4](#)[5](#)[6](#)[7](#)[8](#)[9](#)

New validation constraints allowing to perform validation of certain certificate extensions:

- **<CA>** - checks if the CA certificate has `BasicConstraints.CA=true` certificate extension;
- **<MaxPathLength>** - checks the validity of `BasicConstraints.pathLenConstraint` certificate extension constraint when present;
- **<PolicyTree>** - verifies the available `CertificatePolicies` extension value according to enforced `PolicyConstraints` in the certificate path (note: `PolicyMappings` extension is not yet supported);
- **<NameConstraints>** - verifies the certificate's subject names according to enforced `NameConstraints` in the certificate path (note: checks on `directoryName` are only supported);
- **<SupportedCriticalExtensions>** - allows definition of an allowed list of critical certificate extensions;
- **<ForbiddenExtensions>** - allows definition of forbidden certificate extensions.

List of modifications within default validation policy is available by [link](#).

[1](#)[2](#)[3](#)[4](#)[5](#)[6](#)[7](#)[8](#)[9](#)

Behavior change in case of missed revocation data:

DSS 5.11 in before:

- **INDETERMINATE / TRY_LATER** – if no applicable revocation data found.

DSS 5.12:

- **INDETERMINATE / CERTIFICATE_CHAIN_GENERAL_FAILURE**
 - absence of revocation data is threatened as RFC 5280 validation failure.
(see “5.2.6 X.509 certificate validation” of ETSI EN 319 102-1 for more information).

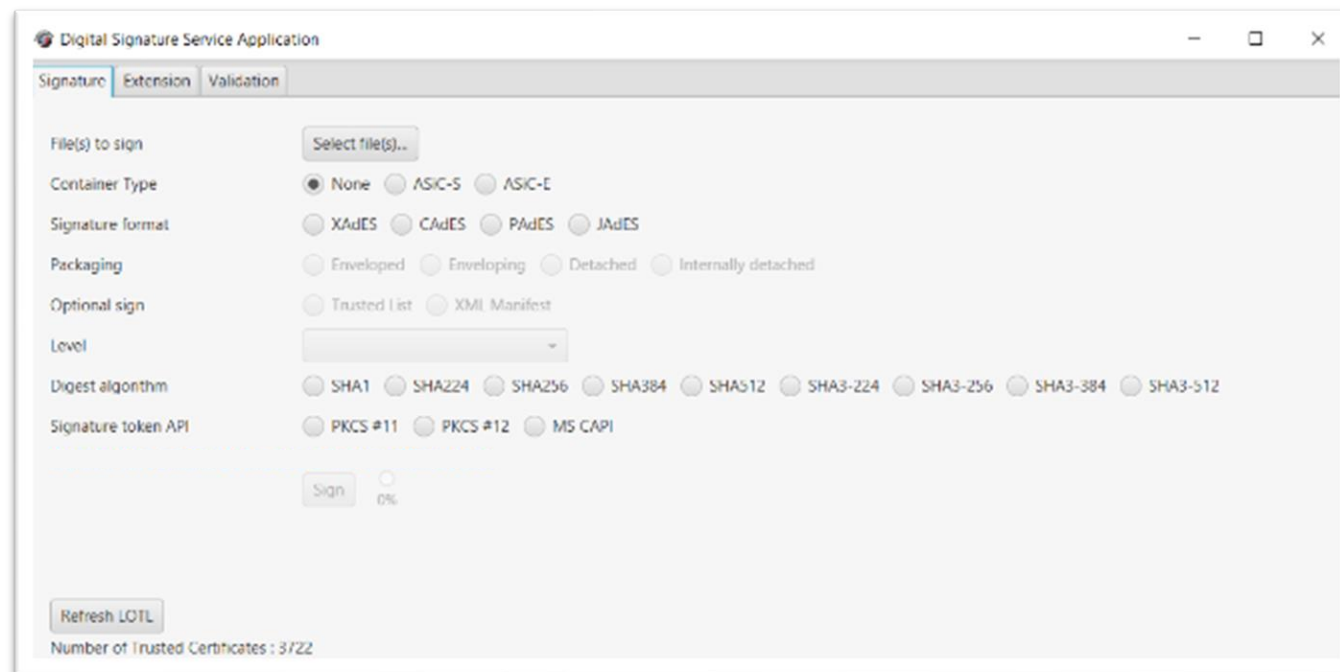
DSS Standalone is a JavaFX application allowing to use certain DSS functions on a local computer.

Can be downloaded from [DSS Demonstrations WebApp](#).

The screenshot shows the 'DSS Demonstration WebApp' interface. The top navigation bar includes the European Commission logo and a breadcrumb trail: 'European Commission > DIGITAL > eSignature > Digital Signature Services > Standalone application'. The main content area is titled 'Standalone application' and contains two sections: 'e-Signature' and 'Server side'. The 'e-Signature' section lists various functions, with 'Standalone application' highlighted in a green box. The 'Server side' section lists other functions like 'Fetch a signature' and 'Validate a signature'. Below these sections is the 'EU LOTL' section. To the right of the navigation menu, there is a 'Standalone application' section with a heading 'Download the standalone application (Windows, x64)'. Below this heading, there are two bullet points: 'Minimal ZIP (application + bat file [uses system properties])' and 'Complete ZIP (application + bat file + OpenJDK + JavaFX SDK)'. Below the bullet points is a 'More info...' section. The 'More info...' section contains text: 'This demo is a standalone application which uses JavaFX. The application connects directly to the CA's infrastructure to retrieve information such as CRL, OCSP, certificates from AIA... All DSS business logic is embedded inside this application (CAdES, PAdES, XAdES, ASiC). This application doesn't require a DSS server.' Below the text is a diagram showing the interaction between a user and the 'Standalone application + SSCD'. The diagram consists of a vertical box on the left labeled 'Standalone application + SSCD' and a person icon on the right. Eight numbered steps are shown as arrows between the user and the application: 1. Fill form and select file to sign (from user to app), 2. Load certificates (from app to user), 3. Select certificate (from user to app), 4. Compute digest (from app to user), 5. Enter pin code (from user to app), 6. Sign digest (from app to user), 7. Sign document (from app to user), and 8. Download signed document (from app to user).

DSS 5.12 provides the following new features in Standalone application:

- signing of multiple documents (e.g. with XAdES, JAdES , ASiC);
- extension of signatures (-T, -LT, -LTA levels);
- validation of signed documents.



Signature extension tab:

Digital Signature Service Application

Signature Extension Validation

Signed file C:\Users\AleksandrBeliakov\Downloads\sample_document.pdf

Original file(s) 0 files

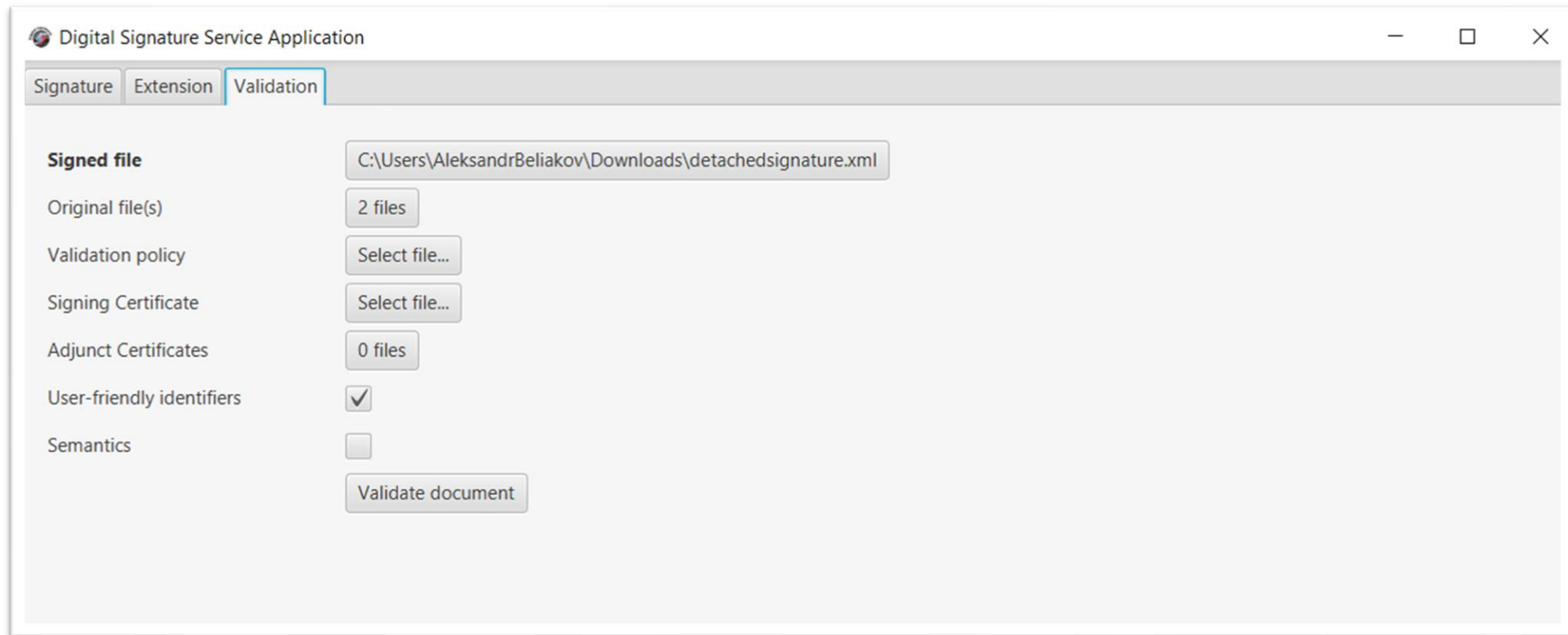
Container Type None ASiC-S ASiC-E

Signature format XAdES CAdES PAdES JAdES

Level PAdES-BASELINE-T

Extend signature(s)

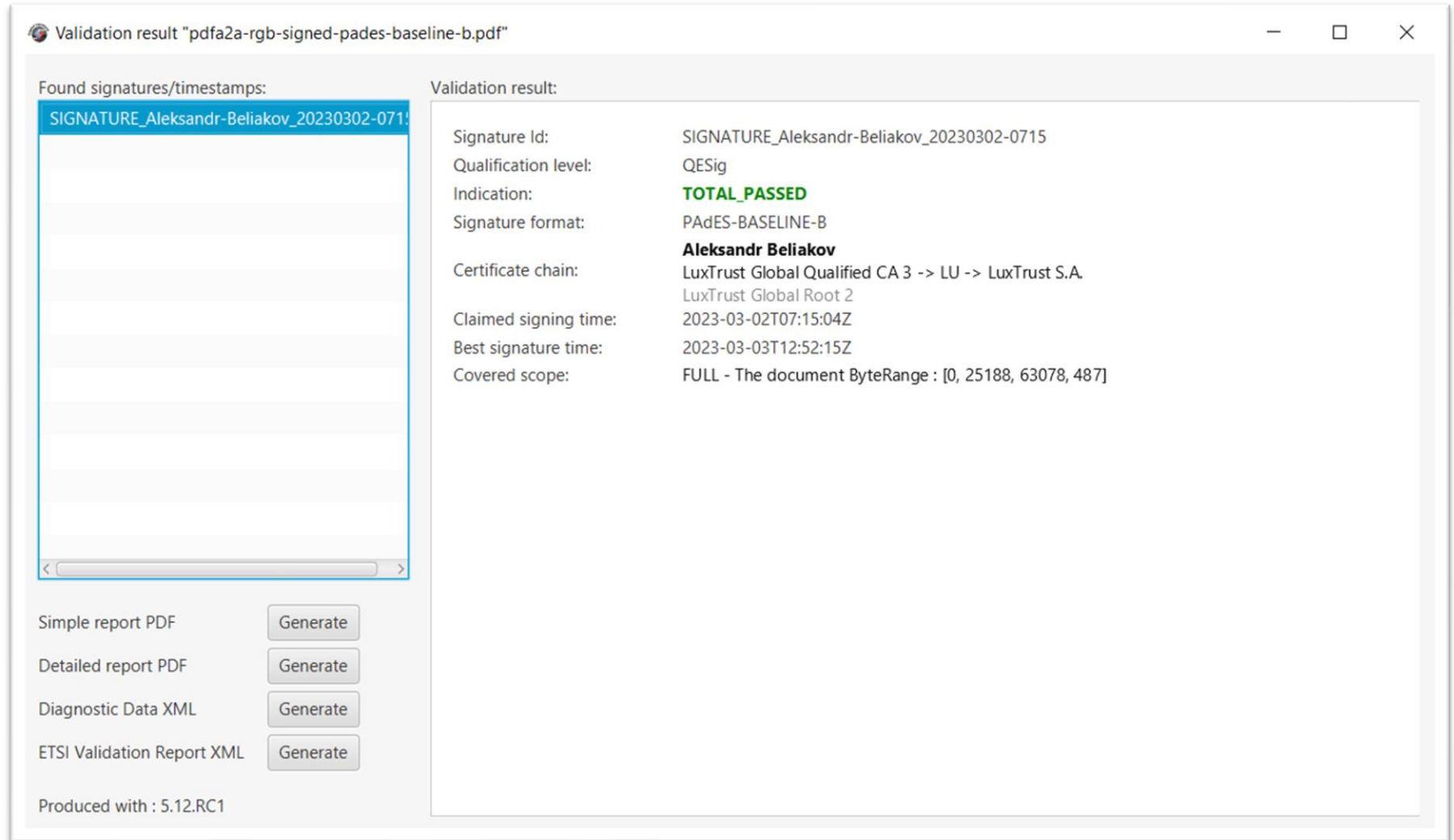
Signature validation tab:



The screenshot shows the 'Digital Signature Service Application' window with the 'Validation' tab selected. The interface includes the following elements:

- Signature** | **Extension** | **Validation**
- Signed file**: C:\Users\AleksandrBeliakov\Downloads\detachedsignature.xml
- Original file(s)**: 2 files
- Validation policy**: Select file...
- Signing Certificate**: Select file...
- Adjunct Certificates**: 0 files
- User-friendly identifiers**:
- Semantics**:
- Validate document** button

Validation result:



Validation result "pdfa2a-rgb-signed-pades-baseline-b.pdf"

Found signatures/timestamps:

| |
|--|
| SIGNATURE_Aleksandr-Beliakov_20230302-0715 |
| |
| |
| |
| |
| |
| |
| |
| |
| |
| |

Validation result:

| | |
|-----------------------|--|
| Signature Id: | SIGNATURE_Aleksandr-Beliakov_20230302-0715 |
| Qualification level: | QESig |
| Indication: | TOTAL_PASSED |
| Signature format: | PAdES-BASELINE-B |
| Certificate chain: | Aleksandr Beliakov LuxTrust Global Qualified CA 3 -> LU -> LuxTrust S.A. LuxTrust Global Root 2 |
| Claimed signing time: | 2023-03-02T07:15:04Z |
| Best signature time: | 2023-03-03T12:52:15Z |
| Covered scope: | FULL - The document ByteRange : [0, 25188, 63078, 487] |

Simple report PDF

Detailed report PDF

Diagnostic Data XML

ETSI Validation Report XML

Produced with : 5.12.RC1



1

2

3

4

5

6

7

8
















9

Common difficulties in recent versions:

- protocol version error in a TL Validation Job;
- SunCertPathBuilderException: unable to find valid certification path to requested target;
- cryptographic constraints failure starting from 01st of January;
- PAdES validation takes more time;
- performance downgrade in TL Validation Job loading (starting from DSS 5.12).

Protocol version error in a TL Validation Job (5.11.1 and before):

The error occurs due to enforcement of a higher version of SSL protocol by remote server.

| | | | | | | | | | | | |
|---|---------|---|----------------------|---|---|---|----------------------|---|----|----|---|
|  | Finland | 43 | 03-Mar-2023 14:19:55 |  |  |  | 16-May-2023 09:18:21 | 1 | 19 | 19 | ▼ |
|  | Unknown | - | - |  |  |  | - | 0 | 0 | 0 | ▲ |
| URL | | https://www.ssi.gouv.fr/uploads/tl-fr.xml | | | | | | | | | |
| Last download attempt | | 03-Mar-2023 14:19:48 | | | | | | | | | |
| Download status | | ERROR  (03-Mar-2023 14:19:48) | | | | | | | | | |
| Download error message | | Unable to process GET call for url [https://www.ssi.gouv.fr/uploads/tl-fr.xml]. Reason : [Received fatal alert: protocol_version] | | | | | | | | | |
| Parsing status | | REFRESH_NEEDED  03-Mar-2023 14:19:55 | | | | | | | | | |
| Validation status | | REFRESH_NEEDED  03-Mar-2023 14:19:55 | | | | | | | | | |
|  | Croatia | 46 | 03-Mar-2023 14:19:55 |  |  |  | 02-Sep-2023 09:48:46 | 4 | 29 | 29 | ▼ |

[1](#)[2](#)[3](#)[4](#)[5](#)[6](#)[7](#)[8](#)[9](#)

Protocol version error in a TL Validation Job (5.11.1 and before):

Can be fixed either by enforcement of TLSv1.3 version of SSL-protocol (shall be supported by JVM):

```
dataLoader.setSslProtocol("TLSv1.3");
```

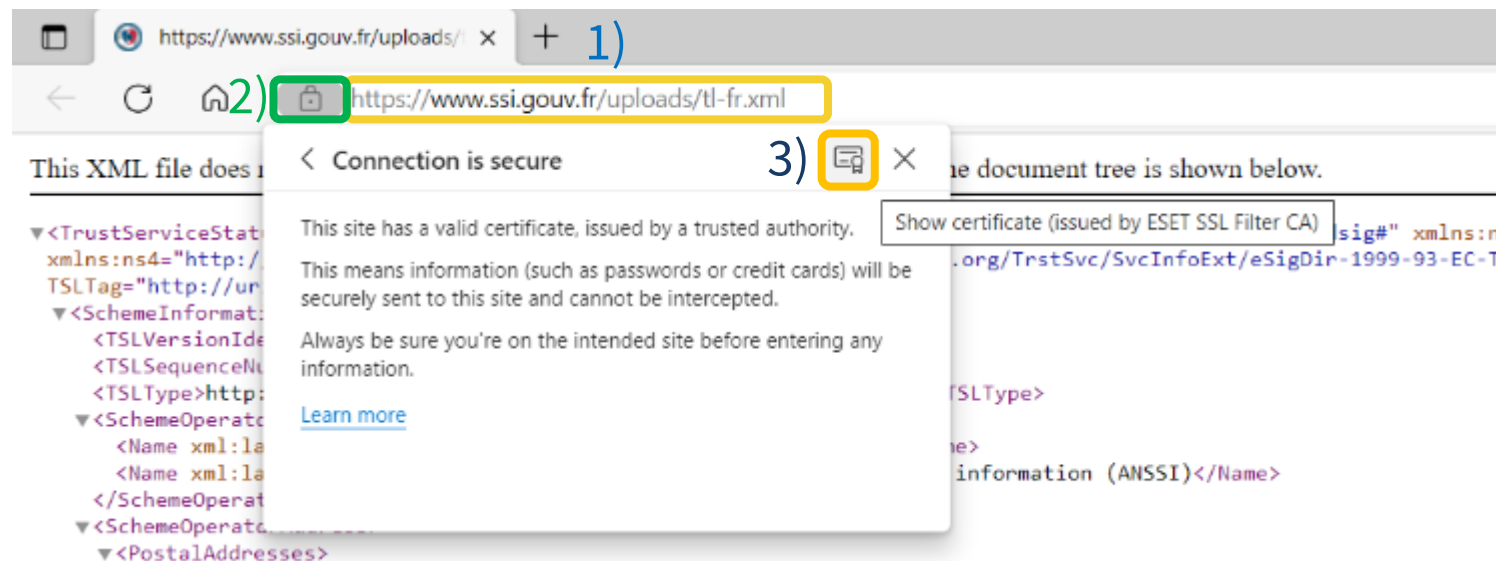
Or by upgrade to DSS 5.12 (provided that TLSv1.3 version is enforced by JVM).

SunCertPathBuilderException: unable to find valid certification path to requested target.

The issue occurs when SSL certificate of the remote host is not trusted by your JVM.

To solve the problem:

- 1) Open the failed URL in browser ;
- 2) Click on “lock” button;
- 3) Open the certificate window;

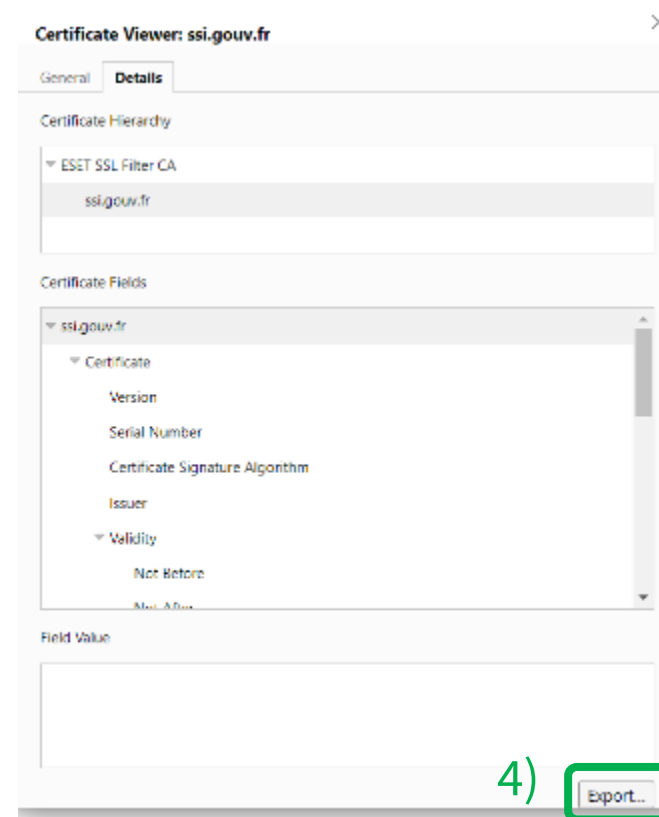


SunCertPathBuilderException: unable to find valid certification path to requested target.

The issue occurs when SSL certificate of the remote host is not trusted by your JVM.

To solve the problem:

- 1) Open the failed URL in browser ;
- 2) Click on “lock” button;
- 3) Open the certificate window;
- 4) Export certificate to a file;



[1](#)[2](#)[3](#)[4](#)[5](#)[6](#)[7](#)[8](#)[9](#)

SunCertPathBuilderException: unable to find valid certification path to requested target.

The issue occurs when SSL certificate of the remote host is not trusted by your JVM.

To solve the problem:

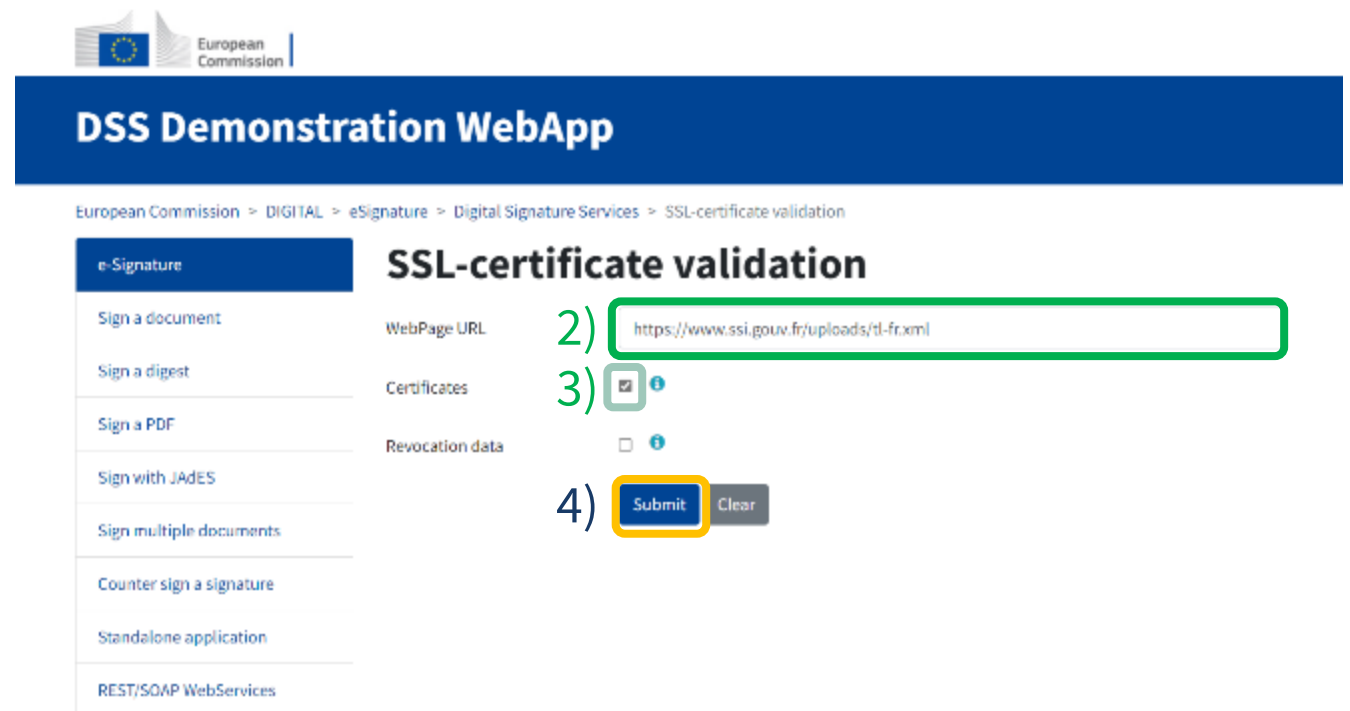
- 1) Open the failed URL in browser ;
- 2) Click on “lock” button;
- 3) Open the certificate window;
- 4) Export certificate to a file;
- 5) Run “Command Prompt” with administrator permission;
- 6) Execute the following like to import certificate to the used JVM keystore:

```
keytool -import -alias newCert -file pathToCert\cert.cer -keystore  
pathToJavaDirectory\lib\security\cacerts -storepass changeit
```

SunCertPathBuilderException: unable to find valid certification path to requested target.

Alternatively, you may export the certificate using [DSS Demonstration WebApp](#):

- 1) Open [SSL certificate validation](#) webpage;
- 2) Copy the failed URL;
- 3) Select “certificates” checkbox;
- 4) Submit the validation;

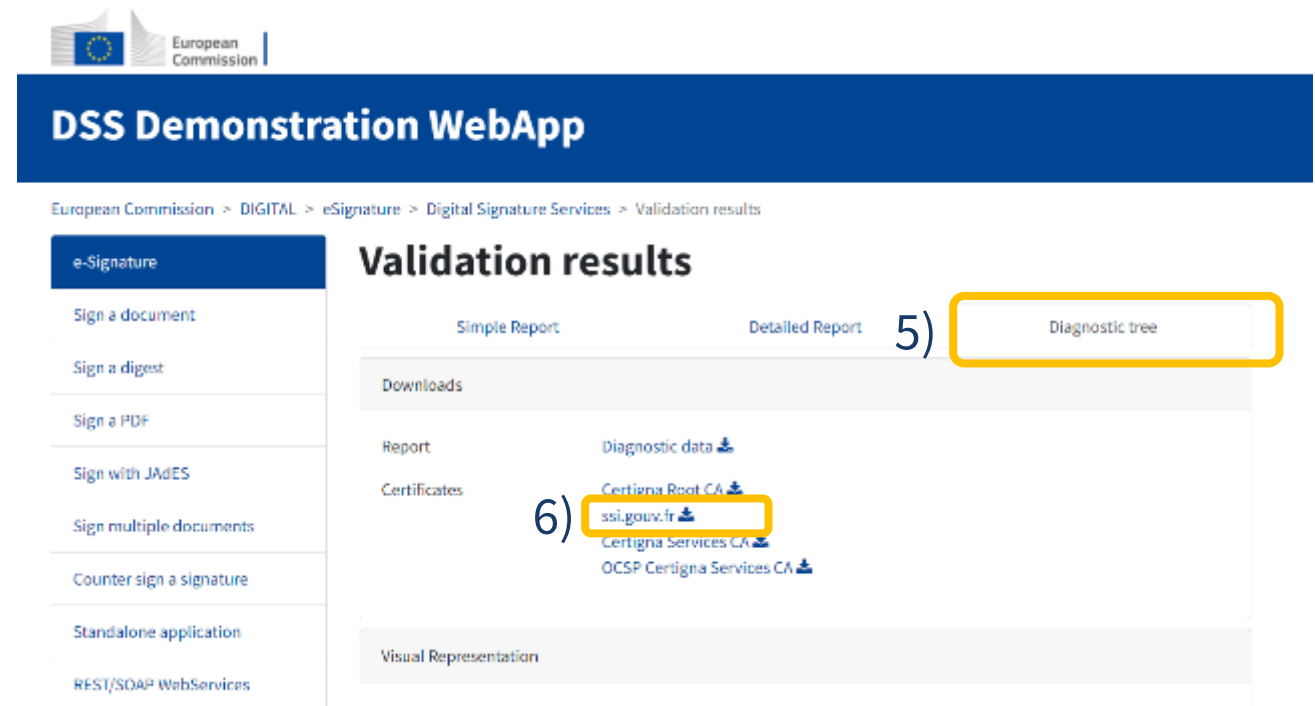


The screenshot shows the 'DSS Demonstration WebApp' interface. The breadcrumb trail is: European Commission > DIGITAL > eSignature > Digital Signature Services > SSL-certificate validation. The main heading is 'SSL-certificate validation'. On the left, there is a sidebar menu with options: Sign a document, Sign a digest, Sign a PDF, Sign with JADES, Sign multiple documents, Counter sign a signature, Standalone application, and REST/SOAP WebServices. The main form area contains three fields: 'WebPage URL' with the value 'https://www.ssi.gouv.fr/uploads/tl-fr.xml' (annotated with '2)'), 'Certificates' with a checked checkbox (annotated with '3)'), and 'Revocation data' with an unchecked checkbox. At the bottom of the form are 'Submit' and 'Clear' buttons (the 'Submit' button is annotated with '4').

SunCertPathBuilderException: unable to find valid certification path to requested target.

Alternatively, you may export the certificate using [DSS Demonstration WebApp](#):

- 1) Login to [SSL certificate validation](#) webpage;
- 2) Copy the failed URL;
- 3) Select “certificates” checkbox;
- 4) Submit the validation;
- 5) Navigate to “Diagnostic tree” tab;
- 6) Download the target certificate.



Cryptographic constraints failure starting from 01st of January.

DSS aligns cryptographic constraints in line with ETSI TS 119 312 which defines recommended resistance duration parameters for cryptographic algorithms. For example:

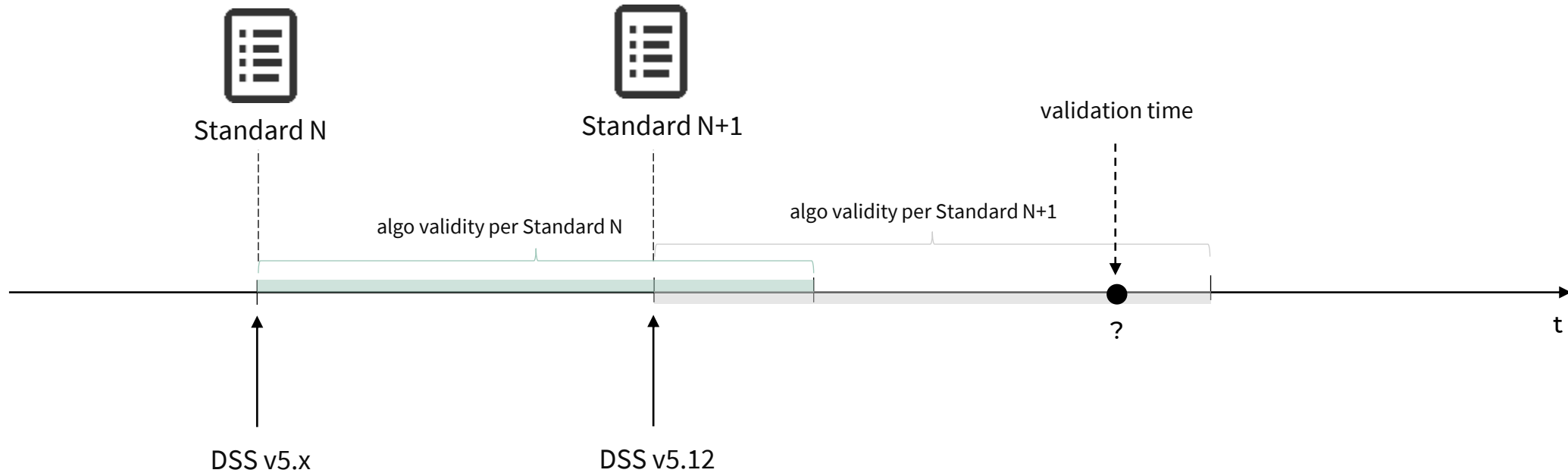
Table 5: Recommended hash functions for a resistance during X years

| Entry name of the hash function | 1 year | 3 years | 6 years |
|---------------------------------|--------|---------|----------|
| SHA-224 | usable | usable | unusable |
| SHA-256 | usable | usable | usable |
| SHA-384 | usable | usable | usable |
| SHA-512 | usable | usable | usable |
| SHA3-256 | usable | usable | usable |
| SHA3-384 | usable | usable | usable |
| SHA3-512 | usable | usable | usable |

For the standard published in 2021 the SHA-224 algorithm is to be valid until end of 2024.

Cryptographic constraints failure starting from 01st of January.

When using an **old version of DSS**, algorithms may expire faster than defined by the updated version of the standard.



Cryptographic constraints failure starting from 01st of January.

When using an **old version of DSS**, algorithms may expire faster than defined by the updated version of the standard.

Signature SIGNATURE_Aleksandr-Beliakov_20220519-1107

| | |
|---------------------------------|--|
| Qualification: | Indeterminate QESig |
| Qualification Details: | The signature/seal is an INDETERMINATE AdES digital signature! |
| Signature format: | PAdES-BASELINE-B |
| Indication: | INDETERMINATE |
| Sub indication: | CRYPTO_CONSTRAINTS_FAILURE_NO_PDE |
| AdES Validation Details: | The certificate validation is not conclusive! The algorithm SHA224 is no longer considered reliable for signature creation! The past signature validation is not conclusive! |
| Certificate Chain: | Aleksandr Beliakov LuxTrust Global Qualified CA 3 LuxTrust S.A. LuxTrust Global Root 2 |
| On claimed time: | 2022-05-19 11:07:26 (UTC) |
| Best signature time: | 2023-03-03 14:04:16 (UTC) |
| Signature position: | 1 out of 1 |
| Signature scope: | Full PDF (FULL) The document ByteRange : [0, 137337, 175227, 568] |

Cryptographic constraints failure starting from 01st of January.

To resolve the issue, you should either:

- Update cryptographic constraints within validation policy:

```
<AlgoExpirationDate Format="yyyy">
  <!-- Digest algorithms -->
  <Algo Date="2005">MD5</Algo> <!-- ETSI TS 102 176-1 (Historical) V2.1.1 -->
  <Algo Date="2009">SHA1</Algo> <!-- ETSI TS 102 176-1 (Historical) V2.0.0 -->
  <Algo Date="2026">SHA224</Algo> <!-- ETSI 119 312 V1.4.2 -->
  <Algo Date="2029">SHA256</Algo> <!-- ETSI 119 312 V1.4.2 -->
  <Algo Date="2029">SHA384</Algo> <!-- ETSI 119 312 V1.4.2 -->
  <Algo Date="2029">SHA512</Algo> <!-- ETSI 119 312 V1.4.2 -->
  <Algo Date="2029">SHA3-256</Algo> <!-- ETSI 119 312 V1.4.2 -->
  <Algo Date="2029">SHA3-384</Algo> <!-- ETSI 119 312 V1.4.2 -->
  <Algo Date="2029">SHA3-512</Algo> <!-- ETSI 119 312 V1.4.2 -->
  ...
</AlgoExpirationDate>
```



1

2

3

4

5

6

7

8

9

Cryptographic constraints failure starting from 01st of January.

To resolve the issue, you should either:

- Update cryptographic constraints within validation policy; or
- Update to the latest version of DSS.

Cryptographic constraints failure starting from 01st of January.

To improve migration of cryptographic algorithms in future versions **DSS 5.12** introduces new attributes in validation policy, namely:

- **UpdateDate** defines the time of the latest update of cryptographic constraints;
- **LevelAfterUpdate** defines validation level for failed algorithms with expiration date after the **UpdateDate**.

```
<AlgoExpirationDate Level="FAIL" Format="yyyy" UpdateDate="2022"
                    LevelAfterUpdate="WARN">
  <!-- Digest algorithms -->
  <Algo Date="2005">MD5</Algo> <!-- ETSI TS 102 176-1 (Historical) V2.1.1 -->
  <Algo Date="2009">SHA1</Algo> <!-- ETSI TS 102 176-1 (Historical) V2.0.0 -->
  <Algo Date="2026">SHA224</Algo> <!-- ETSI 119 312 V1.4.2 -->
  <Algo Date="2029">SHA256</Algo> <!-- ETSI 119 312 V1.4.2 -->
  <Algo Date="2029">SHA384</Algo> <!-- ETSI 119 312 V1.4.2 -->
  <Algo Date="2029">SHA512</Algo> <!-- ETSI 119 312 V1.4.2 -->
  ...
</AlgoExpirationDate>
```

The behavior in DSS 5.12 in case of expiration of cryptographic algorithm after update:

Signature SIGNATURE_Aleksandr-Beliakov_20220519-1107

| | |
|---------------------------------|--|
| Qualification: | QESig |
| Signature format: | PAdES-BASELINE-B |
| Indication: | TOTAL_PASSED |
| AdES Validation Details: | The algorithm SHA224 is no longer considered reliable for signature creation! |
| Certificate Chain: | Aleksandr Beliakov LuxTrust Global Qualified CA 3 LuxTrust S.A. LuxTrust Global Root 2 |
| On claimed time: | 2022-05-19 11:07:26 (UTC) |
| Best signature time: | 2023-03-23 14:41:55 (UTC) |
| Signature position: | 1 out of 1 |
| Signature scope: | Full PDF (FULL) The document ByteRange : [0, 137337, 175227, 568] |

PAdES validation takes more time

Starting from version 5.9, DSS performs supplemental validation of a document:

- Visual comparison of document page content (by screenshot generation);
- Comparison of internal PDF objects;
- Other complementary checks.

The validation **can be disabled** for performance reasons.

Please see [7.6.1.3. Disabling PDF comparison security checks](#).

[1](#)[2](#)[3](#)[4](#)[5](#)[6](#)[7](#)[8](#)[9](#)

Performance downgrade in TL Validation Job loading (starting from DSS 5.12).

The issue is caused by a **BouncyCastle dependency update** enforcing additional validation rounds on RSA keys starting from version **1.72**.

Starting from **BouncyCastle 1.73** it is possible to disable validation by providing the following system property:

```
System.setProperty("org.bouncycastle.rsa.max_mr_tests", "0");
```



1

2

3

4

5

6

7

8

9

An upgrade to JAXB 3 (with Java 8 support) and Jakarta namespaces is planned for DSS version 6.x.

Migration to 6.x will be possible only with update of all dependencies to Jakarta namespaces.

Support of **DSS 5.x** with JAXB 2 is planned for version 5.13 with a minimal support after (i.e. bug fixes, etc.).



1

2

3

4

5

6

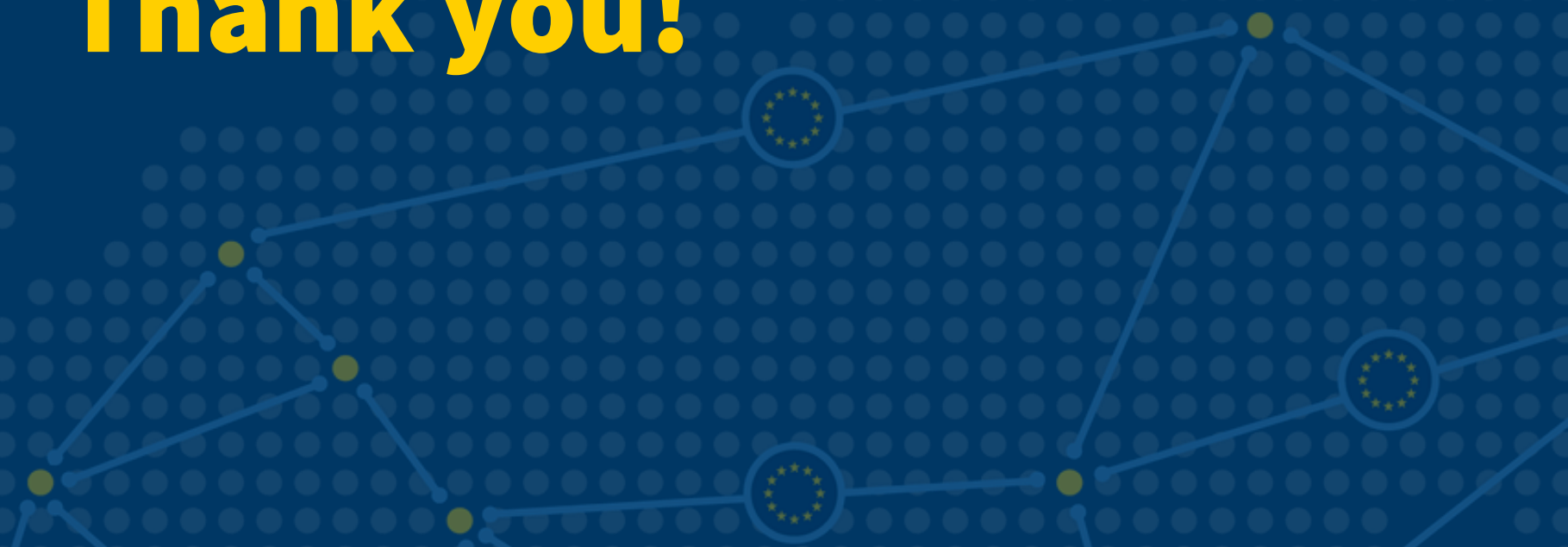
7

8

9

- 1) DSS 5.12 version:
 - Available on Maven Central;
 - new features.
- 2) Migration to the new version:
 - common difficulties;
 - code changes.
- 3) Future versions of DSS:
 - migration to Jakarta.

Thank you!



9

Closing remarks

Apostolos Tolis APLADAS



*Back to the table
of contents*

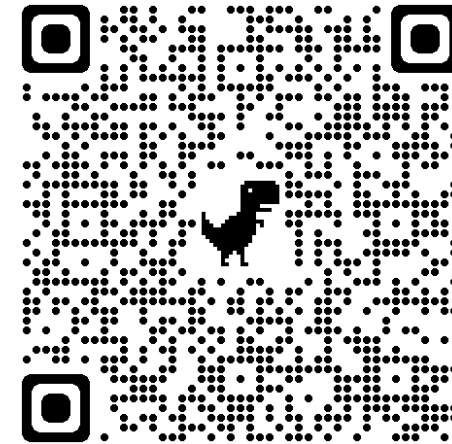
What's next?

Discover how our international collaboration with third countries accelerate the mutual recognition of trust services across the EU borders.

During this event, you can explore the experiences of the Commission, third countries, private market and industry experts via interesting presentations and panel discussions.

EC-3rd Countries Trust Services Forum

June 12th - 09:00 - 17:30 (CET)
Hybrid event, Brussels



Scan the QR Code or [visit the event page](#) to register for the event

Get in touch

For more information, please consult the [eSignature website](#).

[Sign up](#) to receive updates about future milestones or upcoming events:

