Submitted to the EC on 07/09/2014

COMPETITIVENESS AND INNOVATION FRAMEWORK PROGRAMME
ICT Policy Support Programme (ICT PSP)

e-CODEX

e-Justice Communication via Online Data Exchange

ICT PSP call identifier: CIP-ICT-PSP-2009-4

ICT PSP main Theme identifier: CIP ICT PSP 2010 5.2 3: E-JUSTICE SERVICES

Project full title: e-Justice Communication via Online Data Exchange

Grant agreement n°: 270968

D 4.8: Concept for Implementation of WP4 (Update of D4.2)

Deliverable Id :	D 4.8
Deliverable Name :	Concept for Implementation of WP4
Status :	V 2.0
Dissemination Level :	PU
Due date of deliverable :	30.06.2013
Actual submission date :	07.09.2014
Work Package :	WP4
Organisation name of lead partner for this deliverable :	Ministry of Justice, Estonia
Author(s):	Adrian Klar Rudi Teschner Viljar Tina Cyril Murie Lesli Hommik
Partner(s) contributing :	DE, EE, FR, WP1, WP4, WP5, WP7

Abstract:

As part of the e-CODEX project, this deliverable is an update of D4.2 that provides the concept for implementation of WP4. The aim is to provide an up-to-date description of modules and building blocks that have been realised, descriptions of detailed processes and requirements and the mapping of the processes to the functionalities of the solution.

Pending approval by EU commission

History

Version	Date	Changes made	Modified by
0.1	08.03.2013	Aligned document to changes made as the library has been developed.	Adrian Klar
0.5	06.06.2013	Review of document to keep it up-to-date, taking new e-CODEX participants into account. Title changed from “D4.2” to “D4.8”	Adrian Klar
0.8	01.07.2013	Took comments from review into account	Adrian Klar
0.95	24.07.2013	Finalisation	Adrian Klar
1.0	29.07.2013	Changes made to the introduction chapter	Lesli Hommik
1.5	25.06.2014	Updated document	Adrian Klar
1.7	16.07.2014	Comments on the updated document analysed and changes made accordingly.	Adrian Klar
1.8	28.07.2014	History of the document corrected	Lesli Hommik
1.9	26.08.2014	Comments on the updated document (final review) analysed and changes made accordingly. Document finalized for submission to the EC.	Adrian Klar Lesli Hommik
2.0	07.09.2014	Final editorial amendments	WP1

Pending approval by EU commission

Table of Contents

HISTORY **1**

TABLE OF CONTENTS **2**

LIST OF FIGURES **5**

LIST OF TABLES **6**

LIST OF ABBREVIATIONS AND ACRONYMS **7**

EXECUTIVE SUMMARY **9**

1 INTRODUCTION **11**

1.1 SCOPE AND OBJECTIVE OF DELIVERABLE 11

1.2 WP4 GENERAL OBJECTIVES AND VISION 11

1.3 METHODOLOGY OF WORK 11

1.4 RELATIONS TO INTERNAL E-CODEX ENVIRONMENT 11

1.5 RELATIONS TO EXTERNAL E-CODEX ENVIRONMENT 12

1.6 QUALITY MANAGEMENT 12

1.7 RISK MANAGEMENT 13

1.8 STRUCTURE OF THE DOCUMENT 14

2 TECHNICAL ENVIRONMENT **15**

2.1 ARCHITECTURE 15

2.1.1 E-CODEX SERVICE PROVIDER 16

2.1.2 CONNECTOR 16

2.1.3 GATEWAY 16

2.2 KEY COMPONENTS 17

2.2.1 TRUST OK-TOKEN 17

2.2.2 ASIC-S CONTAINER 19

2.2.3 DIGITAL SIGNATURE SERVICES 20

2.2.4 SIGNATURE CERTIFICATE 20

3 FUNCTIONALITIES **21**

3.1 CONNECTOR 21

3.1.1 GENERAL DESCRIPTION 21

3.1.2 CONNECTOR (OUTGOING) 21

3.1.3 CONNECTOR (INCOMING) 22

Pending approval by EU commission

3.2 RECIPIENT 22
3.2.1 MANUAL REVALIDATION 22
4 PROCESSES AND TASKS 23
4.1 SUBMISSION OF DOCUMENTS 23
4.2 SIGNATURE VERIFICATION 28
4.2.1 PRECONDITIONS 28
4.2.2 PROCESS FLOW: SIGNATURE VERIFICATION 28
4.2.3 POST CONDITIONS 34
4.3 CREATE THE "TRUST OK"-TOKEN (PDF & XML) 34
4.3.1 PRECONDITIONS 34
4.3.2 DATA FLOW: CREATION OF THE "TRUST OK"-TOKEN 35
4.3.3 POST CONDITIONS 36
4.4 CREATE CONTENT ARCHIVE 37
4.4.1 PRECONDITIONS 37
4.4.2 DATA FLOW: CREATION OF A CONTENT ARCHIVE 38
4.4.3 PREDEFINED PROCESS: ADD FILE(S) TO ZIP ARCHIVE 39
4.4.4 POST CONDITIONS 41
4.5 CREATE AND SIGN AN ASIC-S SIGNATURE CONTAINER 41
4.5.1 PRECONDITIONS 41
4.5.2 DATA FLOW: CREATION OF ASIC-S SIGNATURE CONTAINER 42
4.5.3 POST CONDITIONS 44
4.6 CERTIFICATE VERIFICATION 45
4.6.1 PRECONDITION: 45
4.6.2 POST CONDITIONS 46
4.7 CREATE VALIDATION REPORT 47
5 DESCRIPTION OF MODULES AND BUILDING BLOCKS TO BE REALISED 48
5.1 JAVA LIBRARY FOR USAGE WITHIN THE CONNECTOR 48
5.1.1 REQUIREMENTS 48
5.1.2 PACKAGING 48
5.1.3 INTERFACES 49
5.1.4 CLASSES AND BASIC INTERFACE IMPLEMENTATIONS 54
5.1.5 ENUMERATIONS 62
5.2 SIGNATURE VERIFICATION 65

Pending approval by EU commission

5.2.1 VERIFICATION AT SENDING SERVICE PROVIDER 65
5.2.2 VERIFICATION AT RECEIVING SERVICE PROVIDER 65
5.2.3 CROSS-BORDER VERIFICATION 66
5.3 SIGNATURE CREATION ON BUSINESS DOCUMENTS 66
5.4 SIGNATURE CREATION ON "TRUST OK"-TOKEN 66
5.4.1 SIGNATURE ON THE PDF VERSION 66
5.4.2 SIGNATURE ON THE XML VERSION 67
5.5 VALIDATION REPORT 70
5.6 "TRUST OK"-TOKEN 71
5.6.1 CONTENT 71
5.6.2 STRUCTURE 72
5.6.3 LINK TO THE VALIDATED DOCUMENTS 80
5.6.4 TRANSMISSION OF THE TOKEN 83
I. REFERENCES 84
A. APPENDIX: VALIDATION REPORT STRUCTURE 85
1 INTRODUCTION 85
1.1 USED TYPE 85
2 VALIDATION REPORT 87
2.1 TIME INFORMATION 87
2.2 SIGNATURE INFORMATION 87

Pending approval by EU commission

List of Figures

Figure 1: e-CODEX Architecture 15
Figure 2: ASiC Container Structure 19
Figure 3: Submission of business documents - Sending side 23
Figure 4: Submission of business documents – Receiving side 26
Figure 5: Basic View on Signature Verification 28
Figure 6: “Check Signature” 29
Figure 7: Closer view on "Signature Integrity Analysis" 30
Figure 8: Closer view on "Certificate Integrity Analysis" 31
Figure 9: Closer view on "Certificate Authority Analysis" 32
Figure 10: Closer view on Certificate Status Analysis 33
Figure 11: Creation of a “Trust Ok”-Token 35
Figure 12: Creation of a Content Archive 38
Figure 13: Closer view on the process “Add file(s) to ZIP archive” 39
Figure 14: Basic Process of ASiC-S signature container creation 42
Figure 15: Closer view on “Valid Archive?” 43
Figure 16: Closer view on “Create signature container” 43
Figure 17: Closer view on “Create XAdES Signature” 44
Figure 18: Closer View on "Certificate Verification” 45
Figure 19: Closer View on "Create Validation Report” 47
Figure 20 Structure of the class "token" 59
Figure 21: Exemplary first page 72
Figure 22: Exemplary second page for a signature-based advanced electronic system 74
Figure 23: Exemplary second page for a authentication-based advanced electronic system 75
Figure 24: Exemplary first page of a national validation report 76
Figure 25: Example for ASiC-S structure applied to a nested container file 80
Figure 26: Example for a valid signature container 81
Figure 27: Signature container within the e-CODEX transport infrastructure 83

Pending approval by EU commission

List of Tables

Table 1: Quality Checklist 12
Table 2: Risks 13
Table 3: Basic View on a XAdES Signature 67
Table 4: Detailed View on the Element "ds:SignedInfo" 68
Table 5 Description "Trust Ok"-Token page 1 72
Table 6 Description "Trust Ok"-Token Assessment of a signature 74
Table 7 Description "Trust Ok"-Token Assessment of an authentication 75
Table 8: Specification of the content archive 81

Pending approval by EU commission

List of Abbreviations and Acronyms

Acronym	*Explanation*
CA	Certification Authority
CAdES	CMS Advanced Electronic Signatures, published by ETSI as TS 101 733
CMS	Cryptographic Message Syntax, see "CAdES"-Description
CRL	Certificate Revocation List, see "RFC 5280" http://www.ietf.org/rfc/rfc5280.txt
DG	Directorate-General
DSS	Digital Signature Standard
eBMS	ebXML Messaging Services
ebXML	Electronic Business using XML
e-CODEX	e-Justice Communication via Online Data Exchange
eCM	e-CODEX Member
EPO	European Payment Order
EQM	External Quality Manager
ETSI	European Telecommunications Standards Institute
GW	Gateway
ICT	Information and communication technologies
ID	Identity
LTV	Long Time Validity
MS	EU Member State
OCSP	Online Certificate Status Protocol, see "RFC 2560" http://www.ietf.org/rfc/rfc2560.txt
PAdES	PDF Advanced Electronic Signature, published by ETSI as TS 102 778
PDF	Portable Document Format
PEPPOL	Pan-European Public Procurement Online http://www.peppol.eu/
PKCS	Public-key cryptography standards
QC	Qualified Certificate
SHA	Secure Hash Algorithm

Pending approval by EU commission

SP	(e-CODEX) Service Provider
SSCD	Secure Signature Creation Device
STORK	Secure Identity across borders linked https://www.eID-stork.eu/
Token	Device that an authorised user of computer services is given to ease authentication
TSL	Trust-Service Status List, published by ETSI as TS 102 231
TSP	Trusted Service Provider
VRE	Validation Report Entry
WP	Work Package
XAdES	XML Advanced Electronic Signatures, published by ETSI as TS 101 903
XML	Extensive Markup Language

Pending approval by EU commission

Executive Summary

Following higher levels of mobility and greater European integration, the number of procedures containing cross-border effects has increased. To meet this demand, more cooperation between the different national judicial systems is needed.

In this context, the goal of e-CODEX is to improve cross-border access for citizens and businesses to legal means in Europe and the interoperability between legal authorities by using instruments of the ICT.

The ambition of e-CODEX is to create a pan-European interoperability layer by connecting already existing national systems to allow communication and data exchange based on the development of common technical approaches and standards.

The project strongly commits itself to adapt and / or adopt the solutions developed by other interoperability projects like PEPPOL, SPOCS, STORK and DG Market's Digital Signature Services Tool.

Whereas the overall scope of the project is bigger, WP4 aims to cover all e-Identity related topics:

- e-Identity management for natural and legal roles, mandates and rights as well as user authentication and authorisation
- Verification and Implementation of e-Signatures.

As several new partners joined e-CODEX due to an extension of the project, an update of the existing deliverables became necessary to provide the new partners with up-to-date information. For this reason, the present document provides an extended and reworked version of the third deliverable D4.2 as it has been written by WP4. Beside several minor changes to the e-Signature related processes, functionalities and software modules that have been applied to the specification at the time of development, this document received major updates within the following topics:

- Updated and more detailed view on both versions, PDF and XML, of the "Trust Ok"-Token.
- Updated descriptions of the implemented software modules as the modules have been improved and changed at the time of development.
- The description of the link between a business document and the "Trust Ok"-Token has been reworked. This link is necessary to provide a solution disabling the usage of a "Trust Ok"-Token for a business document it does not refer to.
- A description of the functionality to support authentication based systems with signatures has been added. These systems do not rely solely on the successful authentication of the user. Instead, the authentication service signs the document and thereby approves that the person that was authenticated and created the document was the claimant that is mentioned within the business document.

Pending approval by EU commission

Beside the detailed description of processes and building blocks this document describes significant key decisions: the usage of DSS, ASiC and the “Trust OK”-Token.

When comparing the e-CODEX use cases to the functionalities provided by other interoperability projects, the result was that most of the missing building blocks can be covered by DG Market’s DSS Tool - a Java based open source software module that can be used to create, extend and validate XAdES, PAdES and CAdES Signatures.

To overcome one of the biggest issues for WP4 that currently not all European citizens are equipped with an electronic signature tool and that advanced electronic signatures are not yet completely interoperable at European level, the concept of the “Trust OK“-Token is introduced providing validation information within a single document.

To realise the “Trust OK”-Token it has been decided that an e-CODEX Service Provider delivering trusted documents has to be characterised as an *advanced electronic system (see section 2.2.1.2)*. The use of an advanced electronic system within e-CODEX will thus guarantee that a trusted document is linked to one particular user, that it is created using means the user can maintain under his sole control and that it has not been changed.

The “Trust OK”-Token basically is a confirmation document signed by the Connector. Within the “Circle of Trust” (see section 2.2.1.1) it is used to assure the authenticity of the delivered documents.

Pending approval by EU commission

1 Introduction

1.1 Scope and Objective of Deliverable

This deliverable provides the concept for implementation of tasks assigned to WP4. The aim is to provide descriptions of modules and building blocks realised, descriptions of detailed processes and requirements and mapping the processes to the functionalities of the solution. This deliverable will reflect the development of necessary modules and building blocks in WP4 for e-CODEX.

1.2 WP4 General Objectives and Vision

The overall objective of WP4 is to establish a model for the use of a European e-Identity framework for data exchange between e-Justice applications and to deal with electronic signatures. Due to the nature of the e-CODEX pilot use cases, WP4 concentrates on electronic signatures. Providing a solution for handling electronic signatures is essential for a successful piloting phase as signatures are especially crucial in the field of justice. The vision of WP4 has been supporting pilots by setting up a solution for e-signature verification in e-CODEX.

1.3 Methodology of Work

The initial deliverable D4.2 was drafted by WP4 author team which consisted of IT-architects and lawyers from Estonia, Germany and France. While writing this deliverable, one workshop was held to discuss open issues and to reach agreements on important issues while other WP4 members were consulted via e-mails. Intensive communication and good cooperation between the authors formed the basis for unity within the work package. The updates of D4.2 which lead to D4.8 were done based on the experience and knowledge gained from the development phase and the information received from new partners in e-CODEX.

1.4 Relations to Internal e-CODEX Environment

This deliverable provides updated information on the concept of implementation of WP4.

Pending approval by EU commission

1.5 Relations to External e-CODEX Environment

In this deliverable, the modules and building blocks have been described in detail and solutions are specified. These specifications could and possibly will affect the piloting phase by being the main basis for the new piloting Member States to also prepare their own signature.

1.6 Quality Management

External quality checks have been performed by the External Quality Manager. Internal quality checks have been done by WP1 team as well as the members of WP4 and several other members of e-CODEX.
The following table gives an overview about the quality checks performed on this deliverable.

Category	Remarks	Checked by
Conformance to e-CODEX template	Firstly done by WP4-leader and also checked by WP1 before submission to EC.	WP4 WP1 EQM
Language & Spelling	Remarks from EQM were taken into account and the deliverable was re-checked by WP4 leader before submission.	WP4 EQM
Delivered on time	Delays in delivery of this deliverable due to limited resources and priorities lying in preparations for piloting.	WP4 WP1 EC
Each technology description contains the correct elements	Checked by IT-architects working on the deliverable.	WP4
Consistency with description in the planning of the project and in other e-CODEX deliverables	Checked by WP4 leader and WP1.	WP4 WP1
Content is fit for purpose	Checked by IT-architects working on the deliverable.	WP4
Content is fit for use	Checked by IT-architects working on the deliverable.	WP4
Commitment within WP	Checked by WP4 leader.	WP4

Table 1: Quality Checklist

Pending approval by EU commission

1.7 Risk Management

The following table gives an overview of the main risks of WP4:

Description	Probability	Impact	Priority	Response	Owner
Partners not contributing in WP4 which causes delays in deliveries.	High	High	Very high	Involvement of new partners in WP4 and enforcement and encouragement of contribution by WP4 leader and the coordinator.	WP4
Problems in making the solution work for every piloting country	High	Medium	High	Close collaboration with piloting countries, including piloting countries in testing and close collaboration with the developer.	WP4
National solutions are not in accordance with the standards and regulations and can't be integrated into the developed solution.	Medium	Medium	Medium	Member States have to modify their national solutions to be in accordance with given standards and regulations.	WP4
The solution created is not suitable for e-CODEX.	Medium	High	High	Experts and good developers have been included in the specification and development phase. Piloting phase will confirm the suitability of the solution and when needed, changes will be made to the solution.	WP4

Table 2: Risks

Pending approval by EU commission

1.8 Structure of the Document

The document is structured as follows:

1. Introduction
2. Technical Environment
   2.1 Architecture
   2.2 Key Components
3. Functionalities
   3.1 Connector
   3.2 Recipient
4. Processes and Tasks
   4.1 Submission of Documents
   4.2 Signature Verification
   4.3 Create the “Trust Ok”-Token (PDF & XML)
   4.4 Create ~~a~~ Content Archive
   4.5 Create and Sign an ASiC-S Signature Container
   4.6 Create Validation Report
5. Modules and Building Blocks
   5.1 Java Library for Usage within the Connector
   5.2 Signature Verification
   5.3 Signature Creation on Business Documents
   5.4 Signature Creation on “Trust Ok”-Token
   5.5 Validation Report
   5.6 “Trust Ok”-Token

Pending approval by EU commission

2 Technical Environment

This chapter gives a general overview about technical parameters that need to be considered as e-CODEX is intended to operate in an environment that is regulated by the European Commission and affected by European and national law.

2.1 Architecture

As stated in the Annex 1 of the Grant Agreement, e-CODEX is an interoperability layer for electronic exchanges in Europe in the field of Justice and should operate within the context of existing solutions of each eCM (e-CODEX Member). It should not be a new centralised approach or duplication of any national solution at the European level.

Within e-CODEX and in cooperation between the technical Work Packages it is agreed that the best way to achieve the goals is to use a gateway-based architecture. The function of the gateway is to act as an access point for each national solution to the interoperability layer developed by e-CODEX. The gateway approach guarantees subsidiarity as it does not overrun the national applications. It converts messages from the national format to a format supported by e-CODEX and *vice versa*.

Figure 1: e-CODEX Architecture

2.1.1 e-CODEX Service Provider

An e-CODEX Service Provider, under the responsibility of a public authority, provides services to users (citizen, judge, lawyer, notaries) or general services (e.g. find a competent court).

An e-CODEX Service Provider can be either a National Service Provider or a European Service Provider like the e-Justice Portal.

Two types of e-CODEX Service Providers exist:

- The first type can be used by a human being, i.e. by a claimant, a defendant, a representative (i.e. lawyer) or a judicial authority, to create PDF / XML documents and send them or to receive documents.
- The second type is an automated e-CODEX Service Provider. It can be used for e-payment, EPO and small claims or to find the competent court.

  The following example describes the latter case:

  The e-CODEX service provider receives a business document containing information from another eCM. With this information, the service provider responds automatically to the business document providing the information about the competent court. As this information has no legal value, there is no need for a “Trust-OK”-Token.

2.1.2 Connector

Under the responsibility of the respective Ministry of Justice, the DG Justice of the European commission or any other competent national authority, the connector contains the functionalities to:

- Add a “Trust OK”-Token to a document generated by an e-CODEX Service Provider.
- Transform the XML document from the national standard to the e-CODEX standard and *vice versa.*
- Retrieve the end user address (e.g. by extracting it from the business XML) for the final national routing.

2.1.3 Gateway

Under the responsibility of the respective Ministry of Justice, the DG Justice of the European Commission or any other competent authority, the gateway has the following functionalities:

- Communicate: Establish a connection to other gateways and connectors
- Send: Format the content of a message to the eBMS3.0 standard.
- Receive: Extract the contents of an eBMS3.0 message.

Pending approval by EU commission

2.2 Key Components

2.2.1 Trust OK-Token

The idea behind the "Trust OK"-Token is to provide the possibility for the receiving party (e.g. a judge) to recognize documents that have been filed by using a trustworthy advanced electronic system based on signature or authentication.

By using the token in accordance with the "Agreement on a Circle of Trust", the receiving party does not need to validate the signature and the certificate itself. Regardless of the signature assessment by the "Trust OK"-Token, the receiving party still has the right and is granted the means to revalidate the signature independently.

The "Trust OK"-Token is human readable and contains either the result of the signature and certificate validation or information regarding the authentication process of the user.

2.2.1.1 Circle of Trust[1]

Several identity providers agree upon that, in terms of identification of persons, they will trust information provided by each one of them in the same way that they trust their own information. That means that if one of them declares that he has properly registered a person, the others will trust in this information. The same principle applies to certification authorities in terms of qualified signatures: they trust each other's certification in the same way they trust their own certification. A closer description of the circle of trust is written down within the document "Agreement on a Circle of trust - adopted by GA on 2013_02_20_v2.pdf".

Pending approval by EU Commission

[1] Description taken from D4.1.1 section 2.1.6

2.2.1.2 Advanced Electronic System

As e-CODEX Service Provider for the civil use cases, e-CODEX will only accept an 'advanced electronic system', an electronic system which meets the following requirements:

(a) the created document is uniquely linked to the user;

(b) the system is capable of identifying the user;

(c) the document is created using means that the user can maintain under his control;

(d) any subsequent change of the data of a created document is detectable;

According to this definition, two approaches are considered to be suitable to be used within e-CODEX: An authentication based- and a signature based advanced electronic system. Where within an authentication based system a user's identity is meant to be validated by a highly secured environment, the signature based system relies on a signature's data to identify the user. A system only using an electronic signature shall use at least an advanced electronic signature that is in accordance with the signature directive[2] to meet the requirement to be an advanced electronic system.

In case of an authentication based system, an authentication service provider is allowed to add a signature to a business document to enhance the security of the document and to approve the successful authentication of the claimant of the business document.

Pending approval by EU commission

[2] http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=CELEX:31999L0093:EN:NOT

2.2.2 ASiC-S Container

The important task to preserve the link between the "Trust OK"-Token and its associated documents and signatures is fulfilled by choosing the ASiC-S container, a data container holding different data objects and associated signatures within a ZIP file.

Figure 2: ASiC Container Structure

By using this container format:

- The relation between the token and its files can be preserved.
- The integrity of the "Trust OK"-Token can be preserved.
- Any change of content is detectable. This includes:
  - Addition of unrelated files.
  - Removal of existing files.
  - Modification of existing files.
- Long-term validity can be achieved by using archival signatures.

The container will be generated and signed within the sending country's connector and validated by the connector on the receiving side.

Details including the specification can be found in section 5.6.3.

Pending approval

2.2.3 Digital Signature Services

DG Market's Digital Signature Services (DSS) libraries provide a solution to create and validate signatures that follow the ETSI standards and thereby should be accepted across Europe. Thereby, the close collaboration with ETSI is reflected in both signature creation and signature verification.

WP4 decided, together with the e-Justice portal, that DSS offers the most appropriate solution for e-CODEX needs.

2.2.4 Signature Certificate

The "Trust OK"-Token requires to be signed to ensure that it has been created by the sending connector and that it has not been altered. For this reason, a signature certificate is located in the connector, not only signing the "Trust Ok"-Token, but signing the whole ASiC-S container as well.

By using consistent connector signatures, it can be avoided that different nationally accepted signature formats are used to sign the token which would negate the advantage of using the "Trust OK"-Token to not having to validate signatures across borders.

This is based on decision 27 of the e-CODEX deliverable D7.3 "High Level Architecture Definition".

Pending approval by EU commission

3 Functionalities

3.1 Connector

3.1.1 General Description

The connector can be located either on the same system as each country's national gateway or on a separate system with a secured connection to the respective national gateway and has the task to convert the message data from national format to e-CODEX format and vice versa. It also validates signatures applied to documents submitted by a Service Provider, generates and validates "Trust OK"-Token and verifies the validity of ASiC-S containers transmitted via e-CODEX.

3.1.2 Connector (outgoing)

The connector (outgoing) is located on the sending side and has the task to convert the message data from national format to e-CODEX format. It also validates documents, signed or not, submitted by the national service provider. The validation results are verified through generation of a signed "Trust OK"-Token.

The connector (outgoing) is able to receive data from different service providers: One can be the e-Justice Portal that basically acts as a European service provider, the others could be the different national e-CODEX service providers. In all cases, the connector will create both the "Trust OK"-Token and the ASiC-S container and thus generates signatures on both of them using DSS and the connector's certificate.

3.1.2.1 e-Justice Portal

The e-Justice Portal decided to use DSS for providing the functionality to sign documents.

By doing so, possible signatures on documents submitted via the portal are automatically restricted to the signatures supported by e-CODEX, simplifying the process of signature validation.

The DSS validation report is available in English and will be included in the "Trust OK"-Token.

3.1.2.2 National e-CODEX Service Provider

Each Member State will have its own national solution to ensure the validity of documents.

Regardless of whether this is achieved by signing the documents or by using an authentication-based advanced electronic system, the national service provider can provide a validation report, either in English and/or in the country's official language, which will be included in the "Trust OK"-Token.

Pending approval by EU commission

3.1.3 Connector (incoming)

The connector (incoming) is located on the receiving side and has the tasks to convert the message data from e-CODEX format to national format and to check the signatures of the ASiC-S Container and the "Trust OK"-Token. If the integrity of all documents is guaranteed, the documents can be passed on to the recipient.

It still has to be decided what procedure applies if the documents cannot be validated successfully. A possible solution would be to add a watermark[3] to the "Trust OK"-Token to indicate that it is defective and continue with the submission of the documents to the recipient.

3.2 Recipient

3.2.1 Manual Revalidation

The recipient (i.e. a judge) should always have the possibility to revalidate documents.

Instead of revalidating the original signatures on a document cross border, revalidating the signature on the "Trust OK"-Token is sufficient as the main purpose of the token is to attest that the documents are legally valid in the sending country.

As the provision of this functionality is not within the scope of e-CODEX, a member state that sees a need to provide this functionality to its end users will have to create a solution to do so itself.

Pending approval by EU commission

[3] A watermark is a visible overlay embedded in the document.

4 Processes and Tasks

4.1 Submission of Documents

A user of e-CODEX submits and receives files though his national system. Such systems, for the purpose of e-CODEX, act as e-CODEX Service Provider.

This chapter describes the complete process of submitting a business document from the Service Provider (SP) in the sender's country up to the delivery to the Service Provider / the user in the receiving country. For a better overview, it is divided into sending and receiving side.

Figure 3: Submission of business documents - Sending side

* The step "Certificate Verification" only takes place when an authentication based system is meant to impose additional security by using the signature of an authentication service provider. This step does NOT describe the certificate validation against a CRL or an OCSP as this is handled within the step "Signature Verification".

1 SP: Authenticate the User:
The Service Provider authenticates the user by means not within the scope of e-CODEX.

2 SP: Create the Business Document:
After successful authentication, the user is able to create a business document based on the data he provides as input.

3 SP: (Optional) Give the User the Ability to Sign the Business Document:
If the Service Provider supports signature, it can grant the user the possibility to sign the business document.

4 SP: Send the Business Document and Attachments to the Connector:
All related documents including the attachments are sent to the connector.

5 Con: Receive the File(s):
The connector receives the files.

6 Con: Signature Verification:
The connector checks the integrity of the business document by checking the applied signatures. For this reason, the connector can rely either on the signature validation of e-CODEX and DSS or on the nationally accepted solution.
The process of signature verification is described in more detail in section 4.2.

7 Con: Certificate Verification:
When an authentication based advanced electronic system is being used, a need comes up in order to secure a user's successful authentication at the time a business document is being created. For this reason, it is possible to sign a file within an authentication based system e.g. by using the signature of an authentication service provider. Within this step the certificate used for this signature can be verified against a list of trusted certificates to verify whether the authentication service provider signing the document is among the trusted service providers. Thereby the list of trusted service providers can be configured for every member state separately. A closer description about the process of certificate verification is described in section 4.6.

8 Con: Create the "Trust OK"-Token:
All relevant information provided by the Service Provider and the results of the signature verification are processed in the generation of the "Trust OK"-Token. Even if the language of the validation report varies due to national implementations, the token itself is only available in English and provided in both supported formats: PDF and XML. See section 4.3 for more information.

9 Con: Create the ASiC-S Signature Container:
The ASiC-S signature container containing the documents and the token is created and signed. See section 4.4 and 4.5 for detailed information.

Pending approval by EU commission

10 Con: (Optional) Verify ASiC-S Container
After creation of the ASiC-S container, the created data can be verified. Thereby it can be verified whether:

- All signatures (ASiC-S container and both versions of the “Trust Ok”-Token) are valid.
- The content of the token is reasonable (e.g. authentication-based token contains authentication data)
- The xml version of the token contains a valid reference to the business document inside the ASiC-S container.

11 Con: Transform the Message to European Format:
Within the connector, the eCM transforms the message to the European format. The definition of the European format is provided by WP5.

12 Con: Send the File(s) to the Gateway:
The connector submits the message to the gateway.

13 SG: Transportation of File(s):
The sending gateway handles the transportation of the message via the e-CODEX network.

Pending approval by EU commission

Figure 4: Submission of business documents – Receiving side

Pending

1 RG: Transportation of File(s):
The receiving gateway receives the message via the e-CODEX network and forwards it to the connector.

2 Con: Receive the File(s):
The connector receives the files.

3 Con: Verify ASiC-S container:
After receiving the ASiC-S container, the created data can be verified. Thereby it can be verified whether:

- All signatures (ASiC-S container and both versions of the "Trust Ok"-Token) are valid. A closer description is written down in section 0.
- The content of the token is reasonable (e.g. authentication-based token contains authentication data)
- The xml version of the token contains a valid reference to the business document inside the ASiC-S container.

4 Con: (Optional) Sign the Signature Container:
After all checks have been completed successfully, the connector signs the signature container to guarantee the integrity of the token and the content archive.

5 Con: Transform to National Format:
The message is transformed from European format to the nationally accepted format.

6 Con: Send the Document(s) to the SP:
The connector forwards the message to the Service Provider.

7 SP: Receive the Documents:
The Service Provider receives the documents and takes care of delivering them to the end user.

8 SP: (Optional) Check the Signature on the Document:
The Service Provider also provides means to let the user revalidate signed documents if he wants to do so by forwarding the files in question to the connector and displaying the results of the revalidation. If desired, this functionality has to be implemented by each member state itself.

Pending approval by EU commission

4.2 Signature Verification

This chapter describes the process of signature verification and the collection of information necessary for the creation of a validation report.

4.2.1 Preconditions

- A file, signed or unsigned, with the need of a signature validation.

4.2.2 Process Flow: Signature Verification

Figure 5: Basic View on Signature Verification

1 Supported Signature available? Depending on whether the file is signed with a signature supported by e-CODEX or not, the process continues as followed:

1.1 Yes:
If the file is signed with a supported signature, the signature will be checked. The file might be signed more than once and every signature should be checked according to the process described in the following chapter.

1.2 No:
If the file is not signed with a supported signature at all, a validation report entry should be generated that states that the file is not signed with a supported signature.

2 The Validation Report is created basing the conclusion on each test result. The process of the report creation is similar to the process described in section 4.6.

4.2.2.1 Predefined Process: Check Signature

Figure 6: "Check Signature"

1 Signature Integrity Analysis:
The first step in analysing a signature is performing the integrity check on the signature itself, see section 4.2.2.2.

2 VRE – Signature Integrity Analysis Result Information:
The results of the signature integrity analysis are added to the report.

3 Certificate Integrity Analysis:
At this point, the certificate analysis described in section 4.2.2.3 is performed.

4 VRE - Certificate Integrity Analysis Result Information:
The results of the certificate analysis are added to the report.

Pending approval by EU commission

4.2.2.2 Predefined Process: Signature Integrity Analysis

Figure 7: Closer view on "Signature Integrity Analysis"

1. Signature Structure Analysis:
   This is the process that analyses the signature to find out which signature standard (XAdES, PAdES) and what profile (BES, etc.) are used and whether the signature is in accordance with the approved standards or not.
2. VRE - Signature Structure Information:
   The results of the signature structure analysis are added to the report.
3. Certificate Access:
   By accessing the signatory's certificate, the public key and information regarding used algorithms can be extracted.
4. VRE - Certificate Information:
   Information about the signatory's certificate and used algorithms are added to the report.
5. Integrity Verification:
   Use the information of the certificate to validate the signature and to verify that the signed content has not been changed.
6. VRE - Integrity Information:
   The results of the complete integrity analysis are added to the report.

Valid for EU commission

4.2.2.3 Predefined Process: Certificate Integrity Analysis

Figure 8: Closer view on "Certificate Integrity Analysis"

1. Certificate Structure Analysis:
   This is the process that analyses the certificate structure to receive detailed certificate information including time stamp information and qualified statements.
2. VRE - Certificate Structure Information:
   The results of the certificate structure analysis are added to the report.
3. Certificate Integrity Analysis:
   As a certificate is signed by the issuing CA, the signature on the certificate needs to be checked to verify its integrity. The necessary steps are described in Figure 7 in steps 3 to 6.
4. VRE - Certificate Integrity Information:
   The results of the certificate integrity analysis are added to the report.
5. Certificate Authority Analysis:
   Not every CA is allowed to issue certificates to create signatures with a legal value. For this reason, the CA has to be validated as described in Figure 9.
6. VRE - Certificate Authority Validation:
   The results of the certificate authority validation are added to the report.
7. Certificate Status Analysis:
   Finally there is the possibility that the certificate is invalid, e.g. if it has been revoked. The certificate status analysis, as it is closer described in section 4.2.2.5, helps to perform this task.
8. VRE - Certificate Status Information:
   The results of the complete certificate status check including OCSP and CRL information are added to the report.

owned by EU commission

4.2.2.4 Predefined Process: Certificate Authority Analysis

Figure 9: Closer view on "Certificate Authority Analysis"

1 Access CA Information:
The information about the issuing CA has to be extracted from the certificate. In a usual X.509 v3 certificate, this can be located within the entry "Issuer" and follows the rules defined in RFC 4519.

2 Extract Country Information:
Usually, the information about the issuing CA contains an attribute about its countryName (e.g. c = DE). This information needs to be extracted to be able to locate the national TSL of the issuing CA.

3 Contact EU TSL:
The European TSL, containing links to every national TSL within European borders, has to be contacted to discover the national TSL containing information about national CAs.

4 Check signature on EU TSL:
As the European TSL has been signed by an official authority to verify its validity, the signature on the European TSL should be validated.

5 Extract National TSL:
Based on the country information of the issuing CA, the responsible national TSL can be extracted from the European TSL.

6 Contact National TSL:
The extracted, national TSL can be contacted to verify the validity of the CA.

7 Check signature on National TSL:
Equal to the EU TSL, every national TSL is signed to ensure that it has not been altered and that it is valid.

8 Validate CA at National TSL:
The CA can be validated at the national TSL. For this reason, at least two actions should be performed:

- Ensure, that the CA can be discovered within the national TSL
- Compare the CA certificate written down in the national TSL with the issuer certificate written down in the signature.

9 VRE - CA Validation Information:
The collected information about the validity of the CA is added to the validation report.

Provided by EU commission

4.2.2.5 Predefined Process: Certificate Status Analysis

Figure 10: Closer view on Certificate Status Analysis

1 Get OCSP Information:
Details for OCSP verification are obtained either from the certificate or from the TSL.

2 OCSP Certificate Status Check:
If an OCSP is available, the OCSP certificate status check is performed.

3 VRE - OCSP (Validation) Information:
The results of the OCSP check is added to the validation report.

4 Status Check returned "revoked" or "good"
If the response of the OCSP contains a valid result, the validation via CRL is not needed.

5 Get CRL Information:
Details for CRL verification are obtained either from the certificate or from the TSL.

6 CRL Certificate Status Check:
If CRL is available, the CRL certificate status check is performed.

7 VRE - CRL (Validation) Information:
The results of the CRL check are added to the validation report.

4.2.3 Post Conditions

- At least one file has been validated.
- The XML version of a validation report, containing the results of each process step, has been generated.

4.3 Create the “Trust Ok”-Token (PDF & XML)

This chapter describes the process of the creation of both the XML- and the PDF version of the “Trust Ok”-Token.

4.3.1 Preconditions

- In case of a signed business document: The signature on the business document has to be verified. The results of this validation have to be accessible for further processing within the national connector.
- If the document was created by an authentication based “advanced electronic system”, details regarding this advanced electronic system and the identity of the user have to be provided.

Pending approval by EU commission

4.3.2 Data flow: Creation of the “Trust Ok”-Token

Figure 11: Creation of a “Trust Ok”-Token

1. Create Empty Token:
   Create both an empty XML- and an empty PDF document to be the basis for the "Trust Ok"-Token.
2. Advanced Electronic System?
   Checks whether the PDF version of the business document has been created by an advanced electronic system.
3. Add Information about Advanced Electronic System to Token:
   Add the information, that the PDF version of the business document has been created by usage of a signature- or authentication based advanced electronic system, to both the XML and the PDF version of the "Trust Ok"-Token.
4. Validation Report?
   Checks whether a validation report is in place.
5. Add Information about Validation to Token
   In case of a validation report being in place: Add the content of the validation report to the token.
6. Signature Based Advanced Electronic System?
   Checks whether a signature is mandatory for the business document. In case of a signature being mandatory, it is necessary that a validation report is in place. In case of authentication based systems, a validation report is optional.
7. Add "Not possible to affirm trust" to Token:
   In case of a signature based system without validation report, the document shall not receive any trust and the result of the technical validation will be set to "FAIL" (The colour will be "red").
8. Add Additional Information to the Token:
   All additional information being of interest for the end user, e.g. conformance of the business documents legal value within the borders of the creating eCM, is added to both versions of the token. A closer view on what Information should be relevant is written down in section 5.6.
9. Create PAdES Signature on PDF- and XAdES Signature on XML-Token:
   Sign the created "Trust Ok"-Token with a signature that is in conformance with the respective standard.

4.3.3 Post Conditions

- The XML- and the PDF version of the "Trust Ok"-Token have been created, including the validation report and the information about the existence of an "advanced electronic system".
- Both versions of the "Trust Ok"-Token have been signed.

Pending approval by EU commission

4.4 Create Content Archive

The usage of ASiC-S dictates, that only one file can be signed by an undefined number of signatures. For this reason, the business document, the "Trust Ok"-Token and all attachments the user decides to submit along with the business document have to be packaged in one single, ZIP based archive before being able to create a valid ASiC-S signature container. As the creation of the ASiC-S signature container will take place in the connector, the ability to create this ZIP archive will be mandatory for the connector as well. The creation of the ASiC-S signature container will be closer described in section 4.5.

4.4.1 Preconditions

- The PDF version of the business document is in place:
  - The integrity and validity of the business document have to be verified, either by validating the signature on the document or the identity of the document creator.
- Both the PDF- and the XML version of the signed "Trust Ok"-Token have to be available.
- Attachments have to be filtered with respect to the file size and supported or unacceptable file formats (e.g. executable files). The restrictions for the file formats are written down in section 5.6.3.2, the limit for the file size is defined in section 3.7 and 3.8 of e-CODEX deliverable D5.3.
- In case of the usage of detached-signatures:
  - The file containing the detached signature has to be in place.

Pending approval by EU commission

4.4.2 Data flow: Creation of a Content Archive

Figure 12: Creation of a Content Archive

1. Create a ZIP archive:
   As first step, an empty ZIP based archive has to be created.
2. Add more files?
   Makes sure that every file will be added to the archive.
   2.1. Add file(s) to ZIP archive:
   This step describes the process of adding new files to the archive. This is described in Figure 13: Closer view on the process "Add file(s) to ZIP archive".

Pending approval by EU commission

4.4.3 Predefined Process: Add file(s) to ZIP Archive

Figure 13: Closer view on the process "Add file(s) to ZIP archive"

Pending appr

Kind of file?
Checks which functionality has to be used:

- Business Document

  1.1. Existing Business Document?
  Checks whether a Business Document already is in place or not.

  - Replace Business Document:
    Replaces the current business document in the ZIP archive.
  - Add Business Document:
    Add the business document to the ZIP archive

  1.2. Set Attribute Business Document:
  Set the respective attribute to remember the business document.

- "Trust Ok"-Token

  2.1. Existing "Trust Ok"-Token?
  Checks the attribute that has to be set when the "Trust Ok"-Token has been added to the ZIP archive.

  - Replace "Trust Ok"-Token:
    Replaces the current "Trust Ok"-Token in the ZIP archive.
  - Add "Trust Ok"-Token:
    Add the "Trust Ok"-Token to the ZIP archive

  2.2. Set Attribute "Trust Ok"-Token:
  Set the respective attribute to remember the "Trust Ok"-Token.

- Detached Signature File

  3.1. Existing Detached Signature:
  Checks whether a file with a detached signature already is in place.

  - Add Detached Signature File
    Add the detached signature file to the ZIP archive.
  - Replace Detached Signature File
    Replaces the current detached signature file in the ZIP archive.

  3.2. Set Attribute Detached Signature:
  Set the respective attribute to remember the detached signature file.

- List of Attachments

  4.1. Add all files to ZIP archive:
  Add multiple files to the ZIP archive.

Pending approval by EU commission

4.4.4 Post conditions

A single ZIP archive, containing one main document, one “Trust Ok”-Token, an undefined number of attachments and optionally a detached signature file has been created and is available for further processing. The attributes being used to remember the business document and the “Trust Ok”-Token are only meant to be used for the creation of the ZIP archive and will not be a part of it.

4.5 Create and sign an ASiC-S signature container

The creation of a signature container following the ASiC-S standard has to be done after the creation of a content archive (see section 4.4) and needs to be a part of the connector.

4.5.1 Preconditions

- A single ZIP archive, containing one business document, one signed PDF-“Trust Ok”-Token and an undefined number of attachments has to be available.
  - Optional detached signature file can be in place.
- The signed “Trust Ok”-Token has to be in place in XML format.
- The possibility to create signatures has to be available at the connector. This should be the connector certificate at the connector the ASiC-S container is created at.

Pending approval by EU commission

4.5.2 Data flow: Creation of ASiC-S Signature Container

Figure 14: Basic Process of ASiC-S signature container creation

1. Valid archive?
   This is a validity check for the content archive that will be signed by the ASiC-S signature container. The steps of the validation process are described in Figure 15: Closer view on "Valid Archive?".
2. If the content archive is valid:
   2.1. Create signature container:
   This step prepares the ASiC-S signature container for the signature creation. The processes done for this preparation are described in Figure 16.
   2.2. Create XAdES Signature:
   The content archive is signed by a detached XAdES signature. The process of signature creation is described in Figure 17.
   2.3. Add XAdES signature to signature container:
   The created XAdES signature is added to the ASiC-S signature container.
3. If the content archive is not valid:
   3.1. Generate Error Report:
   In case that the content archive is not valid, an error report is filed.

Pending approval by EU commission

4.5.2.1 Valid Archive?

Figure 15: Closer view on "Valid Archive?"

1. Valid file?
   Checks whether the received file is a ZIP based archive or not.
2. Existing Business Document?
   As a business document is mandatory for e-CODEX, an archive without a business document has to be seen as invalid.
3. Existing "Trust OK"-Token?
   If there is a business document, there has to be a "Trust OK"-Token as well.
4. Set Attribute "Archive" to valid or invalid:
   The attribute "Archive" is only set to valid if the content archive meets all requirements listed above.

4.5.2.2 Predefined Process: Create Signature Container

Figure 16: Closer view on "Create signature container"

1. Create ZIP container:
   The basis for the signature container is a ZIP based container.
2. Add Mimetype to ZIP container:
   The usage of the additional file "mimetype" as it has been described in [ASiC-S] section 5.2.2 is mandatory for this container.
3. Add content archive to ZIP container:
   Finally, the content archive, containing all data that is meant to be signed, has to be added to the signature container.

4.5.2.3 Predefined Process: Create XAdES Signature

Figure 17: Closer view on "Create XAdES Signature"

1. Create XAdES Structure:
   The basic structure of the detached XAdES signature as it has been defined in [XAdES] section 4.4.1 has to be created.

   Note: All information available at the time the structure is created will be added to it. At this time, the XMLDSig-Part of the XAdES signature will be empty!
2. Create signature:
   The XMLDSig-Part of the XAdES signature is created, signing the content archive as detached signature.
3. Add signature to XAdES structure:
   The created XMLDSig-Part is added to the prepared XAdES structure.

4.5.3 Post conditions

A valid ASiC-S signature container has been created:

- The content archive is signed once with a detached XAdES signature that has been created by the connector of its SP of origin.
- The content of the signature container is a ZIP archive containing one business document, one PDF version of the "Trust Ok"-Token, an undefined number of attachments and optionally a file containing a detached signature.
- The file extension of the ASiC-S signature container still is ".zip.asics".

Pending approval by EU commission

4.6 Certificate Verification

This part describes the certificate verification against a TSL of trusted authentication service providers. It only takes place in case of an authentication based system being secured by a signature being present on the business document. This chapter does NOT describe the certificate verification taking place within the signature verification as this already has been described within chapter 4.2.2.5.

4.6.1 Precondition:

- An authentication based system has to be in use. Within this system the authentication service provider has to sign the business document to accredit that the person that created the document was successfully authenticated at the time of document creation and thereby really is the claimant of the document.
- A valid TSL has to be configured, being in accordance with the specification ETSI TS 102 231. This TSL has to contain a list of certificates that belong to trusted authentication service providers.

4.6.2 Data flow: Certificate Verification

Figure 18: Closer View on "Certificate Verification"

1. Signature Present?
   As, for a certificate verification against a TSL, a certificate needs to be in place, the first step is to check whether a signature is in place where the certificate can be extracted from.
2. Authentication Based System?
   If a signature is present, it has to be checked whether the configured system is authentication based. In case of a signature based system, the certificate check is not necessary as the certificate is intended to belong to a user and not to a trusted authentication service provider.
3. Extract Certificate from Signature
   If every precondition is met, the certificate of the signatory is extracted from the signature.

4. Check Certificate against TSL
   The certificate then is checked against a configured TSL. If the certificate is present within the TSL, the signatory of the document has been verified as trusted authentication service provider.
5. Consider Validation Result within Legal Validation
   The result of the validation will be taken into account at the time the legal validation result is created. The fact that the signatory is not a trusted authentication service provider will have a negative effect on the legal validation result of the business document.

4.6.3 Post conditions

The certificate has successfully been validated against the configured TSL.

Pending approval by EU commission

4.7 Create Validation Report

Figure 19: Closer View on "Create Validation Report"

1.- 4. Process Signature Information:
In this step, information regarding the signature itself is processed. This includes the occurrence of a supported signature, its structural information, the used algorithms as well as the results of the integrity check.

5.- 6. Process Certificate Information:
In this step, information regarding the signatory's certificate is processed. This includes the results of the structural analysis, the integrity checks and the OCSP and / or CRL certificate status checks.

7. Process Certificate Authority Information:
All information related to the certification authority that issued the signatory's certificate is processed at this point.

8. Process Certificate Status Information
Information received from either the OCSP or the CRL are processed within this step.

9. Calculate Trust-Level:
The legal- and technical trust level is calculated based on a set of rules that is specified independently for each sending country. This calculation relies on the results of previously performed analytic processes that are compared to the applicable rule set.

Some examples:

- If all required tests are successful, both trust levels are set to green, according to the traffic light principle.
- If it is not possible to conduct some tests successfully, but these are not required in the sender country to have a legally valid document, this can also be considered and the legal trust level could be set to green. Depending on the sending member states needs, the technical trust level could be set to green or yellow.

approval by EU commission

5 Description of modules and building blocks to be realised

5.1 Java Library for Usage within the Connector

The signature verification, creation of the "Trust-Ok"-Token and the creation of the ASiC-S signature container will be a part of the connector that has to be implemented for every e-CODEX gateway. To prevent every eCM to have to create a solution on its own, e-CODEX will create a library to provide these functionalities. This Java library will be provided as single Java archive (jar file) to make its usage as comfortable as possible.

5.1.1 Requirements

To make the usage of the library possible, the following requirements need to be fulfilled:

- Java needs to be installed. (At least Java 1.6)
- Connection to the internet needs to be in place.
- e-CODEX connector certificate needs to be accessible.

5.1.2 Packaging

The functionalities of the library are, from a user's point of view, separated into two important packages:

- eu.ecodex.dss.model
  The package "model" contains all classes that are able to handle and to store the actual information of the library.
- eu.ecodex.dss.service
  This package contains all classes and interfaces that provide the functionality and thereby are necessary to use the library

Pending approval by EU commission

5.1.3 Interfaces

This chapter will provide specifications for interfaces that can be implemented for the usage within e-CODEX. The usage of interfaces allows the library to use various implementations in situations where it might be needed (e.g. it is planned to be possible either to use DSS or a national solution to validate signatures)

5.1.3.1 ECodexContainerService

An implementation of this interface will be the main class of the library providing all necessary methods for the handling of both the ASiC-S container and the "Trust Ok"-Token.

Class Name: ECodexContainerService

Package: eu.ecodex.dss.service

Method Summary	
ECodexContainer	addSignature(ECodexContainer container)
CheckResult	check(ECodexContainer container)
ECodexContainer	create(BusinessContent businessContent, TokenIssuer issuer)
ECodexContainer	receive(InputStream asicInputStream, InputStream tokenStream)
void	setContainerSignatureParameters(SignatureParameters sigParam)
void	setTechnicalValidationService(ECodexTechnicalValidationService validationService)
void	setLegalValidationService(ECodexLegalValidationService validationService)
void	setEnvironmentConfiguration(EnvironmentConfiguration conf)

<u>Method Details:</u>

- addSignature(ECodexContainer container)

  Method to add an additional signature to an existing ASiC-S container.

  Parameters:

  container — An ECodexContainer object as it can be created using the methods "create" and "receive" of an ECodexContainerService

  Returns:

  An ECodexContainer object being signed with an additional signature.

Pending approval by EU commission

- check(ECodexContainer container)

  Verifies the integrity and the content of a given ASiC-S container and the respective “Trust Ok”-Token information.

  Parameters:

container	An ECodexContainer object as it can be created using the methods “create” and “receive” of an ECodexContainerService

  Returns:
  A CheckResult object, containing information about discovered discrepancies within the validation of the ASiC-S container and the “Trust Ok”-Token.

- create(BusinessContent businessContent, TokenIssuer issuer)

  This method is used to create an ASiC-S container and a “Trust Ok”-Token.

  Parameters:

businessContent	An object providing the business content for the ASiC-S container and the “Trust Ok”-Token, e.g. business document and attachments.
issuer	Information about the Issuer of the “Trust Ok”-Token, e.g. the name of the issuer and the country.

  Returns:
  An ECodexContainer object being signed by the connector it has been created in.

- receive(InputStream asicInputStream, InputStream tokenStream)

  “receive” can be used at an receiving connector to create an ECodexContainer object out of received data streams.

  Parameters:

asicInputStream	An input stream for an received ASiC-S container.
tokenStream	The respective XML “Trust Ok”-Token.

  Returns:
  An ECodexContainer object being usable for further processing.

- setContainerSignatureParameters(SignatureParameters sigParam)

  As the DSSECodexContainerService needs to be able to create signatures on both the ASiC-S container and the “Trust Ok”-Token, this method will be used to configure the signature creation functionality.

  Parameters:

sigParam	The signature parameters, e.g. the keystore, the signing certificate and the respective passwords.

Pending approval by EU commission

- setTechnicalValidationService(ECodexTechnicalValidationService validationService)

  For the first part of a business documents validation, a technical validation service needs to be configured to make the technical evaluation for a given document possible.

  Parameters:

validationService	The functionality to do a technical validation for a given business document.

- setLegalValidationService(ECodexLegalValidationService validationService)

  The second part of a business documents validation is its legal assessment. For this reason, a LegalValidationService needs to be set within an ECodexContainerService object.

  Parameters:

validationService	The functionality to do a legal validation for a given business document.

- setEnvironmentConfiguration(EnvironmentConfiguration conf)

  The EnvironmentConfiguration can be used to set various parameters within an ECodexContainerService object. Currently known, possible parameters are e.g. proxy configuration and the certificate store configuration, which checks whether a received ASiC-S container has been created by an authority listed within the certificate store.

  Parameters:

conf	The actual configuration.

Pending approval by EU commission

5.1.3.2 ECodexTechnicalValidationService

The ECodexTechnicalValidationService can be used within an implementation of an ECodexContainerService to create a technical assessment for a given document. Thereby it does not matter whether this assessment is based on authentication data or a signature.

Class Name: ECodexTechnicalValidationService

Package: eu.ecodex.dss.service

Method Summary	
TokenValidation	create(Document businessDocument, Document detachedSignature)
Document	createReportPDF(Token token)
void	setEnvironmentConfiguration(EnvironmentConfiguration conf)

- create(Document businessDocument, Document detachedSignature)

  Usable to verify the technical validity of a document. In case of a signature-based system, this will be a verification of the document's signature. Within an authentication-based system, this will be a validation of given authentication data.

  Parameters:

businessDocument	A business document the validation has to be done for.
detachedSignature	Optional parameter for documents being signed with a detached signature.

  Returns:
  A TokenValidation object, containing the validation data for the given business document in a standardised way.

- createReportPDF(Token token)

  If the "Trust Ok"-Token is meant to provide a PDF version of a validation report (e.g. the DG MARKT DSS validation report as it is created at the time of signature validation), this method can be used to transform existing validation data into a PDF document.

  Parameters:

token	If access to existing "Trust Ok"-Token data is necessary, this attribute is used.

  Returns:
  The PDF version of a validation report.

Pending approval by EU commission

- setEnvironmentConfiguration(EnvironmentConfiguration conf)

  The EnvironmentConfiguration can be used to set various parameters within the ECodexTechnicalValidationService object.

  Parameters:

  conf — The actual configuration.

5.1.3.3 ECodexLegalValidationService

An ECodexLegalValidationService can be used by an ECodexContainerCreationService to create a legal assessment for a business document. Thereby, the legal assessment can be based on the results of an ECodexTechnicalValidationService.

Class Name: ECodexLegalValidationService

Package: eu.ecodex.dss.service

Method Summary	
LegalValidationResult	create(Token token)
Void	setEnvironmentConfiguration(EnvironmentConfiguration conf)

- create(Token token)

  Creates a legal assessment for a given business document. The validation results of a technical validation thereby can be used as basis for the legal validation.

  Parameters:

  token — Technical validation data for a given business document.

  Returns:
  A LegalValidationResult object, containing the legal validation data for a given business document.

Pending approval by EU commission

- setEnvironmentConfiguration(EnvironmentConfiguration conf)

  The EnvironmentConfiguration can be used to set various parameters within the ECodexLegalValidationService object.

  Parameters:

  conf The actual configuration.

5.1.4 Classes and Basic Interface Implementations

This chapter will give an overview of the most important classes of the e-CODEX library. In addition it will give an overview of the basic implementation for the previously described interfaces and a description about how they are meant to be implemented.

5.1.4.1 DSSECodexContainerService

The DSSECodexContainerService is an implementation of the interface "ECodexContainerService" mainly relying on the functionality of DG MARKT DSS to create and verify the necessary signatures.

Class Name: DSSECodexContainerService

Package: eu.ecodex.dss.service.impl.dss

Implemented Interface: eu.ecodex.dss.service.ECodexContainerService

Implementation Details:

- addSignature(ECodexContainer container)

  As DG MARKT DSS supports the handling of ASiC-S containers, the implementation will use the functionalities of DSS libraries to add additional signatures to a given container object.

- check(ECodexContainer container)

  The basic implementation of the method "check" will be used to do the following analysis of received and created ASiC-S containers:
  - Verifies the signature on both the XML and the PDF version of the "Trust Ok"-Token and the ASiC-S container.
  - Checks whether the connector certificate the sending side used is present within the official e-CODEX connector certificate store. The respective certificate store can be configured within the DSSECodexContainerServices environment configuration.
  - Checks whether the "Trust Ok"-Tokens content is plausible, e.g. if authentication data is present within an authentication based "Trust Ok"-Token.

Pending approval by EU commission

- create(BusinessContent businessContent, TokenIssuer issuer)

  Within the method "create", the following functionalities will be provided:
  - Assessment of a given business document, using the legal- and technical validation service of the DSSEcodexContainerService object.
  - Creation of "Trust Ok"-Token (PDF and XML version).
  - Signing of the "Trust Ok"-Token, using the SignatureParameters being set for the DSSEcodexContainerService object.
  - Creation of ASiC-S container, using the SignatureParameters being set for the DSSEcodexContainerService object.

- receive(InputStream asicInputStream, InputStream tokenStream)

  The implementation of the method "receive" will create an ECodexContainer object out of the received data. The content of the "Trust Ok"-Token thereby is received from the XML version of the received "Trust Ok"-Token, taken from the parameter "tokenStream".

5.1.4.2 DSSECodexTechnicalValidationService

This class will be the basic implementation of the interface "ECodexTechnicalValidationService" and can be used for the technical validation within either a signature based system or an authentication based system being secured with a signature. For signature validation it mainly relies on the functionality of DG MARKT DSS.

Class Name: DSSECodexTechnicalValidationService

Package: eu.ecodex.dss.service.impl.dss

Implemented Interface: eu.ecodex.dss.service.ECodexTechnicalValidationService

<u>Implementation Details:</u>

- create(Document businessDocument, Document detachedSignature)

  The method "create" of the basic implementation is called by the basic implementation of the ECodexContainerService and fulfils several tasks:
  1. Validation of a signature on a given business document
     The validation of the business document will be done using the signature validation functionality of DG MARKT DSS, whereby detached signatures shall also be taken into account. The result of this validation (an XML validation report) then can be used as basis for the next task.
  2. To provide a standardised way to use the validation information, the method "create" will match the original XML validation result to the structure of a TokenValidation object. The original XML validation report thereby will not get lost and is meant to be transmitted within the TokenValidation object as well.
  3. In case of an authentication based system with a signature, the used certificate will be verified against a configured list of trusted certificates to make sure that the signatory of the business document is a trusted authentication service provider.
  4. For further processing, the TokenValidation object will be transmitted back to the ECodexContainerService implementation.

Pending approval by EU commission

- createReportPDF(Token token)

  DG MARKT DSS already has a method to create a PDF validation report out of an existing XML validation report as it has been created by DG MARKT DSS. For this reason, the mentioned functionality simply will be used within this method to create the PDF version of the original validation data.

The new desired behaviour, to be able to react on authentication based systems being secured by a signature, made adjustments within the library necessary. Due to this reason, the methods described below have been added to the DSSECodexTechnicalValidationService with the release of version 1.8:

- setAuthenticationCertificateTSL(String authenticationCertificateTSL)
  setAuthenticationCertificateTSL(InputStream authenticationCertificateTSL)
  setAuthenticationCertificateTSL(byte[] authenticationCertificateTSL)

  This method (in one of its versions) can be used to configure the list of trusted certificates to be used for certificate verification. The format of the trusted list being configured thereby has to be in accordance with the ETSI standard for TSLs, TS 102 231.

- isAuthenticationCertificateLOTL(boolean isLOTL)

  Analog to the European TSL, the configured TSL could be a LOTL, a “list of the lists”. This kind of list would not contain entries for trusted certificates. Instead the list would refer to multiple TSLs, each containing a list of trusted certificates. In case of this kind of list being present this method has to be used with its parameter being set to “true”.

- initAuthenticationCertificateVerification()

  After configuration this method has to be used for one time initialisation of the TSL.

Pending approval by EU commission

5.1.4.3 DSSECodexLegalValidationService

This basic implementation of the interface ECodexLegalValidationService provides a legal assessment for a document that will be based on the results of the DSSECodexTechnicalValidationService. The rules for this legal validation are as followed:

- Legal validation is valid if the result of the technical validation was “Successful”
- Legal Validation is invalid if the technical validation result is “Failed” or “Sufficient” or, in case of authentication based systems with additional signature, if the certificate used for the signature is not within the list of trusted authentication service providers.

Class Name: DSSECodexLegalValidationService

Package: eu.ecodex.dss.service.impl.dss

Implemented Interface: eu.ecodex.dss.service.ECodexLegalValidationService

Implementation Details:

- create(Token token)

  Provides the result of the basic legal validation as a LegalValidationResult object.

Pending approval by EU commission

5.1.4.4 ECodexContainer

Class Name: ECodexContainer

Package: eu.ecodex.dss.model

The class ECodexContainer will provide objects containing all information of an ASiC-S container. An ECodexContainer shall contain the following data:

- ASiC-Document

  The ASiC-S signature container as “in memory” document. This container is ready to be transported within the e-CODEX transport infrastructure.

- Business Document

  The actual business document for a given use case and the document the created “Trust Ok”-Token refers to.

- Business Signature

  This attribute is used in case of a detached signature being in place for a given business document. If there is no detached signature, this attribute can be null.

- List of Business Attachments

  Within this attribute, a list of attachments will be provided, containing all attachments that have been added to a business document.

- Token Documents (PDF and XML Version)

  The “in memory” representation of both the PDF- and XML version of the “Trust Ok”-Token.

- Token Information

  The token information will make the information of a “Trust Ok”-Token accessible within a Java environment.

Pending approval by EU commission

5.1.4.5 Token

Class Name: Token

Package: eu.ecodex.dss.model.token

The class “token” will provide the information of a “Trust Ok”-Token to any Java program that needs to work with a token’s data. The structure of the token should be as followed:

Figure 20 Structure of the class "token"

Pending approval

5.1.4.6 MemoryDocument

This class can be used to improve the handling of files. It implements the interface "Document" and is used to provide an "in memory" representation of a given document.

Class Name: MemoryDocument

Package: eu.ecodex.dss.model

Implemented Interface: eu.ecodex.dss.model. Document

Constructor Summary
MemoryDocument (byte[] document)
MemoryDocument (byte[] document, String name)
MemoryDocument (byte[] document, String name, MimeType mimeType)

Constructor Details:

- MemoryDocument (byte[] document)

  Initialises a MemoryDocument object with a nameless document and without any information about the mimetype.

  Parameters:

document	the content of a document

- MemoryDocument (byte[] document, String name)

  Initialises a MemoryDocument object, whereby the mimetype of the object will be set automatically based on an analysis of the filename.

  Parameters:

document	the content of a document
name	the filename of the document. In case of an existing extension, the mimetype of the MemoryDocument object will be set automatically, e.g. the mimetype of the file "Message.xml" will be set to "MimeType.XML". In cases of no or an unknown file extension, the mimetype will be set to "MimeType.BINARY".

Pending approval by EU commission

- MemoryDocument (byte[] document, String name, MimeType mimeType)

  This constructor should usually be used when working with MemoryDocument objects as it makes necessary to give all important information to work with a file.

  Parameters:

document	The content of the document
name	The filename of the document
mimeType	The mime type of the document

Method Summary	
InputStream	openStream()
String	getName()
MimeType	getMimeType()

<u>Method Details:</u>

- openStream()

  Returns a stream to the document.

  Returns:
  A ByteArrayInputStream to the document.

- getName()

  Returns the name.

  Returns:
  The name of the document as String.

- getMimeType()

  Returns the mime type.

  Returns:
  The mime type of the document as MimeType object.

Pending approval by EU commission

5.1.5 Enumerations

5.1.5.1 Advanced Electronic System

The enumeration AdvancedElectronicSystem will be usable to help with the decision whether the advanced electronic system is authentication- or signature based. As the advanced electronic system mostly is used within the information given to a “Trust Ok”-Token, the enumeration concerning the advanced electronic system will be located within the model-package of the token.

Enumeration Name: AdvancedSystemType

Package eu.ecodex.dss.model.token

Name	Translation
AUTHENTICATION_BASED	Authentication-based
SIGNATURE_BASED	Signature-based

Method Summary	
String	getValue()

<u>Method Details:</u>

- getValue()

  Returns the translation to the advanced electronic system the object has been set to.

  Returns:
  The translation for the advanced electronic system, e.g. “Authentication-based” if the AdvancedSystemType has been set to “AUTHENTICATION_BASED”.

Pending approval by EU commission

5.1.5.2 MimeType

The enumeration MimeType will alleviate the handling of mime types within the handling of documents. At least, the following MimeTypes should be supported.

Enumeration Name: MimeType

Package eu.ecodex.dss.model

Name	Translation
BINARY	application/octet-stream
XML	text/xml
PDF	application/pdf

Method Summary	
MimeType	fromFileName(String name)
String	getCode()

<u>Method Details:</u>

- fromFileName(String name)

  This method returns a MimeType based on the extension of a filename.

  Parameters:
  name Name of a document

  Returns:
  The mime type of a filename, e.g. “XML” for the filename “Message.xml”.

- getCode()

  Returns the translation to the mime type the object has been set to.

  Returns:
  The translation for the mime type, e.g. “text/xml” if the mime type has been set to “XML”.

Pending approval by EU commission

5.1.5.3 Configuration

As the library is meant to be usable on different systems, several properties might need to be initialized or configured to guarantee faultless procedures.

5.1.5.4 Certificate

The library will have to create valid signatures for several documents, e.g. the ASiC-S signature container and the "Trust Ok"-Token. For this reason, the solution using this library needs to have access to the connector certificate of the national e-CODEX connector and the respective keystore.

5.1.5.5 Proxy Configuration

The Java library needs to be able to contact TSLs and CAs. For this reason, the library will have to send and receive http- and https messages. As a national solution might use a proxy server, it is necessary to configure a proxy within the class.

5.1.5.6 Logging

As recording system events might become necessary for several participants, the Java library that will be provided by e-CODEX will create several logging messages for important processes, e.g.:

- Sending of data and requests to the internet (e.g. contacting a TSL, certification authority)
- Receiving data from the internet (e.g. receiving data about a TSL, certification authority)
- Creation of the signature container
- Signature validation on the signature container
- …

To create these logging messages, e-CODEX will rely on the "free to use" library "log4j"[5]. Every MS is free to create and configure a "log4j"-logger within the connector to log these messages.

Pending approval by EU commission

[5] http://logging.apache.org/log4j/

5.2 Signature verification

Signature verification can be separated in three topics. One of them is the verification of the business document at the sending Service Provider. This verification is necessary to make the creation of the "Trust Ok"-Token possible as, in case of signed documents, the token is meant to provide information about the signature's validity to the end user. The second topic is the verification at a receiving Service Provider and will mainly be used for the verification of signatures on the ASiC-S signature container. As last and most complicated topic, the cross-border verification of signatures has to be kept in mind. This will be necessary for both the signature verification at cross-border Service Providers (e.g. the e-Justice portal) and the revalidation of signed business documents that have been received by an end user.

5.2.1 Verification at Sending Service Provider

The verification of a signature created at an eCM is under the responsibility of the respective eCM. The outcome of this verification can be transformed to a validation report (See section 5.5). This validation report can be transmitted to the e-CODEX Java library to be included in the "Trust Ok"-Token.

In case of a cross-border Service Provider (e.g. the e-Justice portal), usage of DG MARKT DSS is recommended to verify a signatures validity. As DSS creates a validation report as it is described in section 5.5 and the current version of DSS will be a part of the Java library provided by e-CODEX, there will not be a need to install additional packages. Nonetheless, every cross-border Service Provider is free to decide which solution shall be used for cross-border signature verification.

5.2.2 Verification at Receiving Service Provider

As the verification of the signature on the business document has been done at the sending Service Provider, the only signatures that need to be verified are the signatures on the ASiC-S signature container and on both versions of the "Trust Ok"-Token. This verification will be done by the method "check()" of the class "eu.ecodex.dss.service.impl.dss.DSSECodexContainerService" as it has been described in section 5.1.4.1. The outcome of this verification will be a "CheckResult" object that can be used to decide, whether the ASiC-S signature container can be trusted or not. In case of a trusted ASiC-S signature container, the receiving Service Provider can decide to:

- Add its own signature to the signature container by calling the method "addSignature(ECodexContainer container)" of the class "eu.ecodex.dss.service.impl.dss.DSSECodexContainerService" and forward the whole signature container to the end user.
- Unpackage the signature container by calling the method "getBusinessContent()" of the class "eu.ecodex.dss.model.ECodexContainer" and forward the files. This solution is advised not to be used as the link between the files and the "Trust Ok"-Token gets lost.

Pending approval by EU commission

5.2.3 Cross-Border Verification

The cross-border verification of signatures will be necessary for the first validation, the revalidation of signatures at cross-border Service Providers and the revalidation of signatures for received documents at the receiving Service Provider. For this reason, the basic implementation of the technical validation service relies on the classes of DG MARKT DSS for signature verification. If the receiving eCM plans to allow cross-border signature revalidation at his national service provider, either the e-CODEX library can be used to create a "Trust Ok"-Token on the receiving side or the DG MARKT DSS libraries can be used to create a validation report.

5.3 Signature creation on business documents

The business document of a case is highly recommended to be signed by the user to make changes to the document detectable and to create a trustworthy link between user and document. The possibility to create this signature should be implemented at the Service Provider and should rely on nationally accepted means.

In case of the European solution, the e-Justice portal, e-CODEX recommends the signature creation to be realised by:

- Usage of DSS, provided by DG MARKT, as main solution.
- Providing a possibility to download, sign and upload the document. This solution should be available for the signature solutions not being supported by DSS.

Nevertheless, the final decision about how to implement the necessary cross border signature creation service is up to the e-Justice portal.

5.4 Signature creation on "Trust Ok"-Token

The "Trust Ok"-Token is necessarily signed to ensure:

- The token has been created by an authority that is allowed to create this kind of token, e.g. the connector of a service provider.
- The token has not been altered during its transmission.

As the "Trust Ok"-Token will be created and signed at the connector of the Service Provider, it is necessary to provide a certificate for the automated creation of electronic signatures at every connector.

5.4.1 Signature on the PDF version

The signature on the PDF version of the "Trust Ok"-Token will be created at the connector and has to be mandatory for further processing of the token.

A signature created by a connector shall be in accordance with the standard PAdES and has to be realised with the profile PAdES-BES as a minimum.

Pending approval by EU commission

5.4.2 Signature on the XML version

To be accessible in an easy way, the XML version of the “Trust Ok”-Token is planned to be transported outside of the ASiC-S signature container. To prevent changes on the content of the XML token, it is highly recommended to sign it at the connector of the sending Service Provider. For this reason, an enveloped XAdES signature seems to be most suitable, at least in the profile BES as it is described in XAdES. In addition, the signature should involve information about the PDF version of the main document to provide a trustworthy link to the signed content. This is realised by adding the file name and the respective hash value to the XML version of the “Trust Ok”-Token. The following tables describe the structure of the described solution based on an example.

```
    <?xml version="1.0" encoding="UTF-8" ?>
    <TrustOkToken xmlns:ns2=”http://www.w3.org/2000/09/xmldsig#”>
        +<Issuer>

1)      <Document>
            <Filename>Business Document.pdf</Filename>
            <Type>PDF</Type>
            <ns2:DigestMethod Algorithm=”SHA-256” />
            <ns2:DigestValue>XmSlDfuCVAd2Dc4Das=</ns2:DigestValue>
        </Document>

        +<Validation>

2)      <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
            Id="sigId-ideebe4db20c4c19460298dda2410e3665">

3)          +<ds:SignedInfo>

4)          <ds:SignatureValue
                Id="value-ideebe4db20c4c19460298dda2410e3665">
                Signature
            </ds:SignatureValue>

5)          +<ds:KeyInfo>
6)          +<ds:Object >

        </ds:Signature>
    </TrustOkToken>
```

Table 3: Basic View on a XAdES Signature

1. Document
   Information about the business document the XML “Trust Ok”-Token refers to.
2. ds:Signature
   This is the root element of the XAdES signature as it is described in XAdES.
3. ds:SignedInfo
   The element “ds:SignedInfo” contains information about the signed content and is described in section 4.3 of the XMLDSig specification. This element will be described closer in Table 4 as this is an important element to understand, what has to be signed.

Pending approval by EU commission

4. ds:SignatureValue
   This element contains the actual value of the digital signature and is specified in section 4.2 of XMLDSig.

5. ds:KeyInfo
   The element “ds:KeyInfo” shall contain the signatories certificate to allow the verification of the signature and is described in section 4.4 of XMLDSig. In the case of e-CODEX, this will be the certificate of the sending Service Provider’s Connector.

6. ds:Object
   The element “ds:Object” contains all additional information about the signature that has to be in place to be in accordance with XAdES.

```
<ds:SignedInfo>
    <ds:CanonicalizationMethod
        Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n#" />
    <ds:SignatureMethod
        Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1" />

1)  <ds:Reference
        Id=”xml_ref_id” URI="">
    <ds:Transforms>
    <ds:Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature" />
    <ds:Transform Algorithm="http://www.w3.org/TR/1999/REC-xpath-19991116 ">
        <ds:XPath>not(ancestor-or-self::ds:Signature)</ds:XPath>
    </ds:Transform>
    </ds:Transforms>

    <ds:DigestMethod
        Algorithm="http://www.w3.org/2000/09/xmldsig#sha1" />
    <ds:DigestValue>6K87kQqCq4FrssZcP0mggA/y3yc=</ds:DigestValue>
    </ds:Reference>

2)  <ds:Reference
        Type="http://uri.etsi.org/01903#SignedProperties"
        URI="#xades-ideebe4db20c4c19460298dda2410e3665">
    <ds:Transforms>
    <ds:Transform
        Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315" />
    </ds:Transforms>
    <ds:DigestMethod
        Algorithm="http://www.w3.org/2000/09/xmldsig#sha1" />
    <ds:DigestValue>GhlbctrqHPEzb42l7RNAomXsEU4=</ds:DigestValue>
    </ds:Reference>
</ds:SignedInfo>
```

Table 4: Detailed View on the Element "ds:SignedInfo"

Pending approval by EU commission

1. ds:Reference Id="xml_ref_id" URI=""
   This part of the signature contains the digest of the XML version of the "Trust Ok"-Token. The URI refers to the ID of the signed content (compare to the element "Object" in Table 3)
2. ds:Reference URI="#xades-ideebe4db20c4c19460298dda2410e3665"
   This reference contains the part of the signature, that contains the additional XAdES information. The reference has to be in place to be in accordance with [XAdES] (section 4.4.1).

Pending approval by EU commission

5.5 Validation Report

The validation report contains all validation data that can be collected within the process of signature validation. In case of e-CODEX, this leads to two scenarios:

- Signature validation with nationally accepted solutions

  If the signature validation is done by the nationally accepted solution, e.g. at the sending Service Provider to create a report for the "Trust Ok"-Token, the report will follow the nationally accepted standards.

- Signature validation with cross-border solutions

  In case of the e-Justice portal as a sending participant of e-CODEX and the revalidation of a signature by means of the e-CODEX signature validation mechanism, the signature validation will be realised by DG MARKT DSS. For this reason, the result of a signature validation will be a report as it has been defined within the documentation of this solution[6].

There will be no standardised validation report as the content of the report is mostly under the control of each member state. From a legal point of view, e-CODEX is not allowed either to add information to these reports or to remove information. The only mandatory requirement to the report is that is has to be transmitted as XML.

Pending approval by EU commission

[6] See A Appendix: Validation report structure

5.6 “Trust Ok”-Token

As already stated in section 2.2.1, the idea behind the “Trust OK”-Token is to provide the possibility for the receiving party (e.g. a judge) to recognise documents filed using trustworthy advanced electronic systems based on signature or authentication.

By using the token in accordance to the “Circle of Trust”, the receiving party does not need to validate the documents, applied signatures and used certificates itself and can rely solely on the token. By doing so, it is no longer necessary to validate signatures and certificates cross-border and deal with the different national standards and implementations.

The “Trust-OK”-Token will be a PDF-File generated by the sending Connector to provide a human readable document. It will be easy to understand with the aim to aid the receiving party in the signature verification process.

Additionally, a machine readable form will be generated to support future developments.

5.6.1 Content

To fulfil the requirement, that the “Trust OK”-Token should be easy to understand, the token itself will be divided into three parts:

- The first part consists of basic information necessary for the receiving party to recognise documents as trustworthy which includes information on the advanced electronic system and an evaluation of the legal trust level. The legal trust level is stated either as "successful" or "unsuccessful".
- The second part gives a technical assessment of the documents signature (signature-based advanced electronic system) or the authentication information (authentication-based advanced electronic system). This assessment consists of a standardised summary of the original validation data and a technical trust level based on this summary.
- The third part will be made of the original validation report provided by either the national solution itself or by the DSS validation tool.

The “Trust OK”-Token will contain:

- Information on the used advanced electronic system
- Information on the time the documents have been filed
- Basic Information on applied signatures and used certificates or the identity of the user
- Evaluation of the technical trust level (red / yellow / green)
- Evaluation of the legal trust level (red / green)
- Original validation report provided by the national solution or the DSS validation tool

Pending approval by EU commission

5.6.2 Structure

5.6.2.1 Human Readable Token (PDF)

As mentioned in 5.6.1, the human readable token will consist of three parts.

The first part will be presented on the first page of the actual token. It includes general information on the advanced electronic system and a legal assessment of the business document. In addition, a national disclaimer and a “validation stamp” in the colour of the legal validation result (green/red) are shown in the bottom of the page.

Figure 21: first page ***Exemplary***	General Information	
	Issuing Country	The country the “Trust Ok”-Token has been created in
	Advanced Electronic System	Information about what kind of Advanced Electronic System has been used to create this token (authentication- or signature based)
	Document Information	Name and Mimetype of the file the token refers to
	Time of Issuance	Time the token has been created
	Legal Result	
	Evaluation of the Document	Final result of the legal assessment of the business document

Table 5 Description "Trust Ok"-Token page 1

Pending approval by EU commission

The second page provides a standardised, technical overview of information the original validation report gives. Depending on the Advanced Electronic System (authentication- or signature-based), the information given by the technical overview varies.

Similar to the first page, the bottom of this page consists of a stamp in the colour of the documents technical validation result (green/yellow/red) and a describing text, e.g. providing information about why a document received a yellow technical assessment.

Pending approval by EU commission

Technical Assessment of the Validation Report

General Information

Issuing Country	DE
Advanced Electronic System	Signature-based
Document Information	PDF, "Formsheet_A_sample1-signed.pdf"
Verification Time	2013-02-25 04:08

Signature information

Signing Time	2013-01-30 12:41
Structure Verification	Successful
Signature Verification	Successful
Signature Level	AdES_QC

Certificate information

Issuer	CN=Test Cert. O=Test CA. C=DE
Certificate Verification	Successful
Validity At Signing Time	Successful

Technical Result

Validation of the Document	Successful

The content of the document has not been altered.

The technical validation of the certificate was not possible as the certificate validation service was not available at the time of validation.

VALIDATION OK SUCCESSFUL

Figure 22: Exemplary second page for a signature-based advanced electronic system

General Information	
Issuing Country	The country the “Trust Ok”-Token has been created in
Advanced Electronic System	Information about what kind of Advanced Electronic System has been used to create this token (authentication- or signature based)
Document Information	Name and Mimetype of the file the token refers to
Verification Time	Time the signature has been validated
Signature Information	
Signing Time	Time the signature has been created
Signature Verification	Result of the signatures mathematical verification
Signature Level	Assessment of the signature level, e.g. Qualified, Advanced or Unknown
Certificate Information	
Issuer	Issuer of the certificate
Certificate Verification	Information about whether it was possible to verify the validity of the certificate at an OCSP or an CRL
Validity at Signing Time	Information about whether the certificate was valid at the time the signature has been created
Technical Result	
Validation of the Document	Assessment of the documents validity

Table 6 Description "Trust Ok"-Token Assessment of a signature

Pending approval by EU commission

Technical Assessment of the Validation Report

General Information

Issuing Country	DE
Advanced Electronic System	Authentication-based
Document Information	PDF, "Formsheet_A_sample1-signed.pdf"
Verification Time	2013-02-26 02:00

Authentication Information

Identity Provider	Test Your Identity Inc.
Username Synonym	John Doe
Time of Authentication	2013-02-26 02:00

Technical Result

Validation of the Document	Successful

VALIDATION OK SUCCESSFUL

Figure 23: Exemplary second page for a authentication-based advanced electronic system

General Information	
Issuing Country	The country the "Trust Ok"-Token has been created in
Advanced Electronic System	Information about what kind of Advanced Electronic System has been used to create this token (authentication- or signature based)
Document Information	Name and Mimetype of the file the token refers to
Verification Time	Time the signature has been validated
Authentication Information	
Identity Provider	Name of the Identity Provider the user has been authenticated at
Username Synonym	Username or synonym of the user
Time of Authentication	The time the user has been authenticated at the identity provider
Technical Result	
Validation of the Document	Assessment of the documents validity

Table 7 Description "Trust Ok"-Token Assessment of an authentication

Pending approval by EU commission

The third part of the document consists of the original validation report as it has been created by the issuing member states' validation software.

Time information
Verification Time Tue Jan 15 18:03:23 CET 2013

Signature information 1
✔Signature verification
Signature algorithm : http://www.w3.org/2000/09/xmldsig#rsa-sha1

Certificate Path Revocation Analysis
✘Summary

Certificate Verification

CN=John Doe, O=Trusted Inc., C=DE

Issuer name
CN=Trusted Inc. CA, O=Trusted Inc., C=DE
Serial Number : 14
Validity at signing time : Result[VALID]
Certificate status UNKNOWN

Trusted List Information
✘Service was found
✘Trusted List well-signed

Signature level analysis
Signature format XAdES

✔Signature Level BES
Signing certicate: CN=John Doe, O=Trusted Inc., C=DE

Signing time : Tue Jan 15 18:01:58 CET 2013

Certificates
There is 1 in the signature.

Certificate of CN=John Doe, O=Trusted Inc., C=DE

Version: 1
Subject:
CN=John Doe, O=Trusted Inc., C=DE
Issuer:
CN=Trusted Inc. CA, O=Trusted Inc., C=DE


```
-----BEGIN CERTIFICATE-----
MIICmDCCAgECAQ4wDQYJKoZIhvcNAQEFBQAwgZExCzAJBgNVBAYTAkRFMQwwCgYD
VQQIEwNOUlcxFDASBgNVBAcTC0R1ZXNzZWxkb3JmMQ8wDQYDVQQKEwZJVC5OUlcx
ETAPBgNVBAsTCGVQYXltZW50MRcwFQYDVQQDEw51UGF5QkwgVGVzdCBDQTEhMB8G
CSqGSIb3DQEJARYSZXBheW1bnRAaXQubnJ3LmR1MB4XDTExMTIwODEyMjgyMioX
DTE0MDkwMzEyMjgyMiowgZYxCzAJBgNVBAYTAkRFMQwwCgYDVQQIEwNOUlcxFDAS
BgNVBAcTC0R1ZXNzZWxkb3JmMQ8wDQYDVQQKEwZJVC5OUlcxETAPBgNVBAsTCGVQ
YXltZW50MRwwGgYDVQQDExNNYWdlbnRvIENsaWVudCBDZXJ0MSEwHwYJKoZIhvcN
AQkBFhJlcGF5bWvudEBpdC5ucncuZGUwgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJ
AoGBAPFHTfwDdLm13NiD1bKfW0gZ66bXd0bfn95EzN6ZMhzJ9+yml1TpB1iA54kBF
GQ39oYrZmXrYli1IB/SuDznX3oL0U7rMxUXXfzeNNN8cTFqbNMfABuGa+nBR+a1b
+YDSPY6CwYrjvEwH313ZmxLS1s62+Gc6J3kCw//sgRhCjIu3AgMBAAEwDQYJKoZI
hvcNAQEFBQADgYEARkPzEpSbifKqDGXx3NothZ2rf174jpywmIiEnWtZ07qVnaaQ
5rUqPNB8d6ipSyAgymJLfULL8BxhKInwL+OZH63x3U3yzsPe3NQu/5pc7C3gt2Lg
mhW4j0ucVk4fjxToSY71ZsVOAVovYIMeQkpGvaRGXD2YR8VrL/Nb7MpRqX0=
-----END CERTIFICATE-----
```


Figure 24: Exemplary first page of a national validation report

Pending Submission

5.6.2.2 Machine Readable Token (XML)

This chapter provides the XML schema that defines the structure of the XML version of the “Trust Ok”-Token.

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema
	xmlns:xsd="http://www.w3.org/2001/XMLSchema"
	xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
	elementFormDefault="qualified"
	attributeFormDefault="unqualified">

<xsd:import namespace="http://www.w3.org/2000/09/xmldsig#"
	schemaLocation="http://www.w3.org/2000/09/xmldsig#" />

<xsd:element name="TrustOkToken" type="TokenType" />

<xsd:simpleType name="AdvancedSystemEnum">
	<xsd:restriction base="xsd:string">
		<xsd:enumeration value="Signature-based"/>
		<xsd:enumeration value="Authentication-based"/>
	</xsd:restriction>
</xsd:simpleType>

<xsd:simpleType name="TechnicalTrustLevelEnum">
	<xsd:restriction base="xsd:string">
		<xsd:enumeration value="FAIL"/>
		<xsd:enumeration value="SUFFICIENT"/>
		<xsd:enumeration value="SUCCESSFUL"/>
	</xsd:restriction>
 </xsd:simpleType>

<xsd:simpleType name="LegalTrustLevelEnum">
	<xsd:restriction base="xsd:string">
		<xsd:enumeration value="SUCCESSFUL"/>
		<xsd:enumeration value="NOT_SUCCESSFUL"/>
	</xsd:restriction>
</xsd:simpleType>

<xsd:complexType name="IssuerType">
	<xsd:sequence>
		<xsd:element name="ServiceProvider" type="xsd:string"/>
		<xsd:element name="Country" type="xsd:string"/>
		<xsd:element name="AdvancedElectronicSystem" type="AdvancedSystemEnum"/>
	</xsd:sequence>
</xsd:complexType>

<xsd:complexType name="DocumentType">
	<xsd:sequence>
		<xsd:element name="Filename" type="xsd:string"/>
		<xsd:element name="Type" type="xsd:string"/>
		<xsd:element ref="ds:DigestMethod" />
		<xsd:element ref="ds:DigestValue" />
		<xsd:element name="SignatureFilename" type="xsd:string" minOccurs="0"/>
	</xsd:sequence>
</xsd:complexType>
```

Pending approval by EU commission

```
<xsd:complexType name="SignatureInformationType">
	<xsd:sequence>
		<xsd:element name="SignatureVerification" type="xsd:boolean"/>
		<xsd:element name="StructureVerification" type="xsd:boolean"/>
		<xsd:element name="SignatureFormat" type="xsd:string" minOccurs="0"/>
		<xsd:element name="SignatureLevel" type="xsd:string" minOccurs="0"/>
	</xsd:sequence>
</xsd:complexType>

<xsd:complexType name="CertificateInformationType">
	<xsd:sequence>
		<xsd:element name="Issuer" type="xsd:string" minOccurs="0" />
		<xsd:element name="CertificateVerification" type="xsd:boolean" />
		<xsd:element name="ValidityAtSigningTime" type="xsd:boolean" />
	</xsd:sequence>
</xsd:complexType>

<xsd:complexType name="AuthenticationInformationType">
	<xsd:sequence>
		<xsd:element name="IdentityProvider" type="xsd:string"/>
		<xsd:element name="UsernameSynonym" type="xsd:string"/>
		<xsd:element name="TimeOfAuthentication" type="xsd:dateTime"/>
	</xsd:sequence>
</xsd:complexType>

<xsd:complexType name="SignatureDataType">
	<xsd:sequence minOccurs="0">
		<xsd:element name="SigningTime" type="xsd:dateTime" minOccurs="0"/>
		<xsd:element name="SignatureInformation" type="SignatureInformationType"/>
		<xsd:element name="CertificateInformation" type="CertificateInformationType"/>
	</xsd:sequence>
</xsd:complexType>

<xsd:complexType name="VerificationDataType">
	<xsd:choice>
		<xsd:element name="SignatureData" type="SignatureDataType"/>
		<xsd:element name="AuthenticationData" type="AuthenticationInformationType"/>
	</xsd:choice>
</xsd:complexType>

<xsd:complexType name="TechnicalResultType">
	<xsd:sequence>
		<xsd:element name="TrustLevel" type="TechnicalTrustLevelEnum"/>
		<xsd:element name="Comments" type="xsd:string" minOccurs="0"/>
	</xsd:sequence>
</xsd:complexType>

<xsd:complexType name="LegalResultType">
	<xsd:sequence>
		<xsd:element name="TrustLevel" type="LegalTrustLevelEnum"/>
		<xsd:element name="Disclaimer" type="xsd:string" minOccurs="0"/>
	</xsd:sequence>
</xsd:complexType>
```

Pending approval by EU commission

```
<xsd:complexType name="SourceType">
	<xsd:sequence>
		<xsd:any minOccurs="0" namespace="##any" maxOccurs="unbounded" processContents="lax"/>
	</xsd:sequence>
</xsd:complexType>

<xsd:complexType name="ValidationType">
	<xsd:sequence>
		<xsd:element name="VerificationTime" type="xsd:dateTime"/>
		<xsd:element name="VerificationData" type="VerificationDataType"/>
		<xsd:element name="TechnicalResult" type="TechnicalResultType"/>
		<xsd:element name="LegalResult" type="LegalResultType"/>
		<xsd:element name="OriginalValidationReport" type="SourceType"/>
	</xsd:sequence>
</xsd:complexType>

<xsd:complexType name="TokenType">
	<xsd:sequence>
		<xsd:element name="Issuer" type="IssuerType"/>
		<xsd:element name="Document" type="DocumentType"/>
		<xsd:element name="Validation" type="ValidationType"/>
		<xsd:element ref="ds:Signature" minOccurs="0"/>
	</xsd:sequence>
</xsd:complexType>

</xsd:schema>
```

Pending approval by EU commission

5.6.3 Link to the Validated Documents

To create a secure link between “Trust Ok”-token and transmitted documents, the standard ASiC-S is applicable for e-CODEX needs. This standard has the positive side effect, that neither transmitted documents, be they signed or not, nor the token can be altered as all documents are secured by the signature of the ASiC-S container.

5.6.3.1 Description

The ASiC-S format provides the possibility to create a signature container for a single file with an undefined number of signatures. To make this solution suitable for e-CODEX needs, it is possible to create a ZIP file containing all files that have to be transported. Afterwards, this ZIP file can be signed by both the sending and the receiving gateway.

Figure 25: Example for ASiC-S structure applied to a nested container file[7]

[7] Source: http://www.etsi.org/deliver/etsi_ts/102900_102999/102918/01.01.01_60/ts_102918v010101p.pdf

5.6.3.2 Technical Specification – ASiC-S Signature Container

As described before, all files that are meant to be signed and transported by the ASiC-S signature container need to be packaged in a single ZIP archive with the following recommendations:

Name:	Signed_Content.zip
Format:	ZIP archive
Content:	• 1 Main Document (Allowed Formats: PDF) • 0 – 1 Detached Signature • 1 "Trust Ok"-Token (Allowed Format: PDF) • 0 – X Attachments (Format restriction: No executable files)

Table 8: Specification of the content archive

If the Service Provider discovers the usage of a detached signature on an attachment, it should be possible to package the signed file and the respective signature within an additional ZIP archive.

The next step in the creation of the ASiC-S signature container will be the creation and preparation of the container itself. Within this process, several rules have to be followed:

- The basis for the ASiC-S signature container should be a ZIP based container.
- The first file within the ASiC-S signature container has to be the file "mimetype".
  - The content of the file has to be "application/vnd.etsi.asic-s+zip".
- The folder "META-INF" has to be in place.
  - The folder "META-INF" has to contain the file "signatures.xml".
- The name of the ASiC-S signature container should be "e-CODEX_Signature_Container".
- To follow the standard, the file extension for transportation should be ".zip.asics". This could be changed to ".zip" at the receiving Service Provider to make it more user friendly to the end user.
- The file "Signed_Content.zip" has to be on the root level of the container.
- No additional files or folders are allowed. Neither on the root level of the ASiC-S container nor in the folder "META-INF".

Figure 26: Example for a valid signature container

Pending approval by EU commission

5.6.3.3 Technical Specification – signatures.xml[8]

The file "signatures.xml" has to be in accordance with the rules described in [ASiC-S]. Within Annex 3 of the ASiC specification, the following namespace declarations apply for the XML Schema definitions:

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema
	targetNamespace="http://uri.etsi.org/02918/v1.1.1#"
	xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
	xmlns="http://uri.etsi.org/02918/v1.1.1#"
	xmlns:xsd="http://www.w3.org/2001/XMLSchema"
	elementFormDefault="qualified"
	attributeFormDefault="unqualified">
<xsd:import
	namespace="http://www.w3.org/2000/09/xmldsig#"
	schemaLocation="http://www.w3.org/TR/2002/REC-xmldsig-core-20020212/xmldsig-core-schema.xsd"/>
```

The XML Schema itself is described in Annex 5 of the ASiC specification, whereby the element <XAdESSignatures> is meant to be the root element of the XML:

```
<xsd:complexType name="XAdESSignaturesType">
	<xsd:sequence>
		<xsd:element ref="ds:Signature" maxOccurs="unbounded"/>
	</xsd:sequence>
</xsd:complexType>

<xsd:element name="XAdESSignatures" type="XAdESSignaturesType">
	<xsd:annotation>
		<xsd:documentation>Schema for parallel detached XAdES Signatures </xsd:documentation>
	</xsd:annotation>
</xsd:element>
```

Pending approval by EU commission

[8] Sources: TS 102918 v1.1.1 (ASiC), TS 103174 v1.2.1 (ASiC) and TS 101903 v1.3.2 (XAdES)

5.6.4 Transmission of the token

The signature container described in section 5.6 can be attached to the XML version of the main document creating the structure described in "Figure 27: Signature container within the e-CODEX transport infrastructure". This structure can easily be transmitted via the transport infrastructure provided by WP5.

Figure 27: Signature container within the e-CODEX transport infrastructure

I. References

ASiC-S TS 102918 v1.1.1	http://www.etsi.org/deliver/etsi_ts/102900_102999/102918/01.01.01_60/ts_102918v010101p.pdf
ASiC-S TS 103174 v1.2.1	http://www.etsi.org/deliver/etsi_ts/103100_103199/103174/01.02.01_60/ts_103174v010201p.pdf
XAdES TS 101903 v1.3.2	http://uri.etsi.org/01903/v1.3.2/ts_101903v010302p.pdf
PAdES TS 103172 v1.1.1	http://www.etsi.org/deliver/etsi_ts/103100_103199/103172/01.01.01_60/ts_103172v010101p.pdf
TSL TS 102 231 v3.1.2	http://www.etsi.org/deliver/etsi_ts/102200_102299/102231/03.01.02_60/ts_102231v030102p.pdf
RFC 4519	http://www.rfc-editor.org/rfc/rfc4519.txt
XMLDSig	http://www.w3.org/TR/xmldsig-core/

Pending approval by EU commission

A. Appendix: Validation report structure[9]

1 Introduction

This section describes the fields provided in the validation report. The content of these fields are recovered from the verification performed during the validation algorithm.

1.1 Used type

To facilitate the readability of the validation report structure, the following information type has been used along the description.

Result type

The "Result type" is used to indicate the success or the failure of individual verification steps. It is defined as follows:

- **Result:** indicates whether the result of the verification step is:
  - Valid;
  - Invalid;
  - Undetermined;
- **Description:** provides additional information about the result of the verification step. In particular it provides explanation in case of invalid or undetermined result.

Revocation verification type

The "Revocation verification type" is used to indicate the success or the failure of the verification of revocation data. It is defined as follows:

- **Status:** contains the status of the certificate, specifies if the certificate is revoked or not;
- **Revocation date:** if the certificate is revoked, this element contains the date and time of revocation;
- **Issuer**: if the certificate is revoked, specifies the name of the issuer of the revocation data;
- **Issuing time**: if the certificate is revoked, specifies the date and time at which the revocation data has been issued.

Time stamp verification type

The "Time stamp verification type" is used to indicate the success or the failure of the verification of a time stamp. It is defined as follows:

- **Format** (Result type): indicates whether the format of the time stamp is ok or not;

[9] This chapter (A Appendix: Validation report structure) is taken from "DSS-DM-Design Model-v1.00.doc" by DG MARKT. The complete documentation is available for download under https://joinup.ec.europa.eu/software/sd-dss/release/all.

Pending approval by EU commission

- **Time stamp content**: contains the content of the time stamp:
  - Serial number;
  - Creation time;
  - Issuer name;
- **Signature verification**: contains information about the validity of the signature of the time stamp:
  - Signature verification (Result type): specifies if the signature is mathematically correct or not;
  - Signature algorithm: provides the name of the algorithm applied for the signature.
- **Certificates path verification** (Result type): indicates whether the certificate path of the time stamp has been verified up to a trusted list.

Pending approval by EU commission

2 Validation report

The validation report will contain the results of the analysis steps performed during the overall process of signature validation. Its content will be used to evaluate the trustability of a document.

The validation report itself will be generated by the DSS tool and will contain the following information.

2.1 Time information

Verification time

Specifies the date and time in which the validation has been launched.

2.2 Signature information

For each signature, the verification algorithm provides the following information in the validation report.

Signature structure verification (Result type**)**

Specifies if the structure of the signature is syntactically correct or not, according to the signature format (CAdES, XAdES or PAdES).

Signature verification

Contains information about the validity of the signature.

- Signature verification (Result type): specifies if the signature is mathematically correct or not;
- Signature algorithm: provides the name of the algorithm applied for the signature.

2.2.1 Certification path and revocation data analysis

Summary (Result type)

Contains the result of the certification path validation.

Certificate verification

For every certificate from the certification path, the following information is provided:

- **Issuer name**: specifies the name of the certificate's issuer;
- **Serial number:** specifies the serial number of the certificate;
- **Subject:** specifies the distinguished name of the certificate;
- **Validity period verification at signing time (**Result type**):** specifies whether or not the signing time fits with the validity period of the certificate;
- **Signature verification:** contains information about the validity of the signature:
  - **Signature verification** (Result type): specifies if the signature is mathematically correct or not;
  - **Signature algorithm**: provides the name of the algorithm applied for the signature.
- **Certificate status (**Revocation verification type**):** contains information about the result of the certificate revocation verification.

Pending approval by EU commission

Trusted list information

- **Service was found:** specifies if a corresponding service has been found in a trusted list during the validation algorithm;
- **Trusted List well signed:** specifies if the corresponding Trusted List was well signed (a signature is present and the signing certificate was referenced in the LOTL).
- **TSP information:** if a service has been found, provides the information about the provider of the corresponding service:
  - **Name**: specifies the name of the service provider, extracted from the trusted list;
  - **Trade name**: specifies the trade name of the service provider, extracted from the trusted list;
  - **Postal address**: specifies the postal address of the service provider, extracted from the trusted list;
  - **Electronic address**: specifies the electronic address of the service provider, extracted from the trusted list;
- **Service information:** if a service has been found, provides the information about this service:
  - **Service type identifier**: specifies the service type identifier of the corresponding service, extracted from the trusted list;
  - **Name**: specifies the name of the corresponding service, extracted from the trusted list;
  - **Status information**: contains the information about the status of the service:
    - **Current status**: specifies the current status of the corresponding service, extracted from the trusted list;
    - **Current status starting date**: specifies the current status starting date of the corresponding service, extracted from the trusted list;
    - **Status at reference time**: specifies the status of the service at the reference time of the validation algorithm, extracted from the trusted list;
    - **Status starting date at reference time**: specifies the status starting date of the corresponding service at the reference time of the validation algorithm, extracted from the trusted list;
  - **Additional service information**: if present this element contains:
    - **Additional information URI**: specifies the URI identifying the additional information, extracted from the trusted list;
    - **Classification**: if present, this element contains the service information classification, extracted from the trusted list.
  - **Qualification**: if present, specifies the qualifier properties related to the certificate, extracted from the trusted list.
- **Taken over by information:** if present, contains the information about the service provider which has the new responsibility of the service:
  - **Information URI**: specifies the URI pointing towards a descriptive text to inform about which entity is currently responsible for the service, extracted from the trusted list;
  - **TSP name**: specifies the name of the currently responsible service provider, extracted from the trusted list;
  - **Scheme operator name**: specifies the name of the Member State's body in charge of the corresponding trusted list, extracted from the trusted list;
  - **Scheme territory**: specifies the country in which the scheme is established, extracted from the trusted list.

Pending approval by EU commission

2.2.2 Signature level analysis

Signature format

Specifies the format of the signature:

- **XAdES;**
- **CAdES;**
- **PAdES.**

BES level

Contains the information related to the BES level of a signature:

- **BES level verification (**Result type**):** specifies whether the signature has reached the BES level or not;
- **Certificates:** contains the list of X509 certificates;
- **Signing certificate (**Result type**):** specifies if the "signing certificate" field references the signing certificate value contained in the signature;
- **Signing time:** Provides information about the signing time of the document.
- **Mime type:** specifies the data format of the signed document;
- **Location:** if present, contains information about the place where the signature was generated;
- **Signer role:** if present, specifies the claimed role of the signer;
- **Commitment type indication:** if present, contains an indication of the type of commitment implied by the signature;
- **Counter signature:** if present, contains the result of the verification of a counter signature:
  - **Signature verification (Result type)**: specifies if the signature is mathematically correct or not;
  - **Signature algorithm**: provides the name of the algorithm applied for the signature.

EPES Level

Contains, if present, the information related to the EPES level of a signature:

- **EPES level verification (**Result type**):** specifies whether the signature has reached the EPES level or not;
- **Signature policy identifier:** specifies the policy under which the signature has been produced.

T Level

Contains, if present, the information related to the T level of a signature:

- **T level verification (**Result type**):** specifies whether the signature has reached the T level or not;
- **Signature time stamp (**Time stamp verification type**):** specifies the result of the verification of the signature time stamp.

Pending approval by EU commission

C Level[10]

Contains, if present, the information related to the C level of a signature:

- **C level verification (**Result type**):** specifies whether the signature has reached the C level or not;
- **Certificate references verification (**Result type**):** Specifies whether the present certificate references match the certificates from the certification path or not;
- **Revocation references verification (**Result type**):** Specifies whether the present revocation references match the revocation data from the certification path or not.

X Level

Contains, if present, the information related to the X level of a signature:

- **X level verification (**Result type**):** specifies whether the signature has reached the X level or not;
- **Signature and references time stamp (**Time stamp verification type**):** for each time stamp, specifies the result of the verification of the signature and references time stamp;
- **References time stamp (**Time stamp verification type**):** for each time stamp, specifies the result of the verification of the references time stamp.

X-L Level[11]

Contains, if present, the information related to the X level of a signature:

- **X-L level verification (**Result type**):** specifies whether the signature has reached the X-L level or not;
- **Certificate values verification (**Result type**):** specifies whether or not the present certificate values match the certificate references present in the C Level;
- **Revocation values verification (**Result type**):** specifies whether or not the present revocation values match the revocation references present in the C Level.

A Level

Contains, if present, the information related to the A level of a signature:

- **A level verification (**Result type**):** specifies whether the signature has reached the A level or not;
- **Archive time stamp verification (**Time stamp verification type**):** specifies the result of the verification of the archive time stamp.

Pending approval by EU commission

[10] The certificate and revocation references are not displayed in this section of the report as they have already been displayed in the certification path and revocation data analysis.

[11] The certificate and revocation values are not displayed in this section of the report as they have already been displayed in the certification path and revocation data analysis.

LTV Level

Contains, if present, the information related to the LTV level of a signature:

- **LTV level verification (**Result type**):** specifies whether the signature has reached the LTV level or not;
- **Document security store verification:**
  - Certificate references verification (Result type): specifies whether the certificate references present in the security store match the certificates of the certification path or not;
  - Revocation references verification (Result type): specifies whether the revocation references present in the security store match the revocation data of the certification path or not;
  - Revocation values verification (Result type): specifies whether the revocation values present in the security store match the revocation data of the certification path or not.
- **Document time stamp verification (**Time stamp verification type**):** specifies the result of the verification of the archive time stamp.

2.2.3 Qualification extension analysis

Qualifications verification

If the qualification elements are present, this section contains the qualification information:

- **QCWithSSCD:**
  - Match criteria (Result type): specifies whether the matching criteria for QCWithSSCD are reached by the signing certificate or not.
- **QCNoSSCD:**
  - Match criteria (Result type): specifies whether the matching criteria for QCNoSSCD are reached by the signing certificate or not.
- **QCSSCDStatusAsInCert:**
  - Match criteria (Result type): specifies whether the matching criteria for QCSSCDStatusAsInCert are reached by the signing certificate or not.
- **QCForLegalPerson:**
  - Match criteria (Result type): specifies whether the matching criteria for QCForLegalPerson are reached by the signing certificate or not.

2.2.4 Certificate content analysis

QC statement verification

Contains the result of the QC statement verification:

- **QCP presence:** specifies if the ETSI defined QCP OID is present;
- **QCP+ presence:** specifies if the ETSI defined QCP+ OID is present;
- **QcCompliance presence:** specifies if the QcCompliance statement is present;
- **QcSSCD presence:** specifies if the QcSSCD statement is present.

Pending approval by EU commission

2.2.5 Final conclusion

Signature qualification

Specifies one of the following properties:

- The signature is QES;
- The signature is AdESQC;
- The signature is AdES.

Pending approval by EU commission