TR 119 100 V0.0.2 (2013-09)

Business Driven Guidance for Signature Creation and Validation

Draft 0.0.2

STABLE DRAFT FOR PUBLIC REVIEW UNTIL 15 NOVEMBER 2013

Download the template for comments:

http://docbox.etsi.org/ESI/Open/Latest Drafts/Template-for-comments.doc

Send comments to E-SIGNATURES COMMENTS@LIST.ETSI.ORG

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

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

http://pda.etsi.org/pda/queryform.asp

Reference
DTR/ESI-0019100

Keywords
e-Signatures, e-commerce, trust service

ETSI

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

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

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

Contents

Intellectual Property Rights

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

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

Foreword

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

Introduction

The European “Rationalised structure for Electronic Signature Standardisation”, ETSI TR 119 000 [i.1], describes the structure of a general framework for electronic signatures standardisation outlining existing and potential standards related to the implementation of electronic signatures and the provision of related trust services by trust service providers. This framework identifies six areas of standardisation with a list of existing and potential future standards in each area.

TR 119 000 includes a set of guidance documents to assist business stakeholders, users and their suppliers in mapping or deriving from their business driven requirements the appropriate selection of electronic signature standards and their options. Each guide addresses a particular area as identified in the aforementioned Rationalised Framework. A complete e-signatures solution will need to address requirements in most of these areas.

This series is based on the process of selecting business scoping parameters for each area of standardisation based on an analysis of the business requirements. The selection of these scoping parameters is based on a process involving an analysis of the business requirements and associated risks leading to an identification of the policy and security requirements and to an analysis of the resulting business scoping parameters from which the appropriate standards and options can be selected. From the requirements expressed in terms of business scoping parameters for an area, each guidance document provides assistance in selecting the appropriate standards and their options for that area. Where standards and their options within one area make use of another area this is stated in terms of scoping parameters of that other area.

This general process of the selection of standards and options is described further in TR 119 000 clause [i.1] 4.2.6.

The present document, addressing area 1 of the Rationalised Framework [i.1], proposes a business driven guided process for implementing generation and validation of electronic signatures in business electronic processes.

1 Scope

The scope of the present document is to propose a **business driven guided process for implementing generation and validation of electronic signatures in business' electronic processes**. The prerequisite of this guided process is the existence of a complete and detailed business analysis and risk analysis of the business' electronic processes (e-processes) in which electronic signatures are aimed at being implemented. Starting from this analysis, which in complex processes may be consolidated in a modelled description of such concerned business e-processes, stakeholders are guided for properly specifying all the relevant parameters (hereafter "business scoping parameters" – BSP's) to be taken into account when implementing the creation and the validation of electronic. Finally, stakeholders are guided for making the best choice among the wide offer of standards from the Rationalised Framework of European Standards for Electronic Signatures (RF henceforth) in order to ensure the best implementation of electronic signatures within the of addressed application / business e-processes.

The guided implementation process proposed by this guide is defined in a way that enables stakeholders to identify their requirements in a commonly understood way and facilitates the identification of the solutions to meet those requirements. This is so because the guide explicitly takes into account:

- parameters directly dependant on the specific application or business process,
- parameters derived from the regulatory/legal framework where the business must be conducted,
- parameters inherent to the different types of signing entities, as well as
- other aspects that do not fall within the above three listed categories but are important to be addressed when implementing electronic signatures.

The purported audience of this document is wide and includes different readers' profiles:

1) Business managers facing the integration of electronic signatures in their business electronic processes will find here an understandable explanation on a suitable approach for implementing electronic signatures and the selection of the relevant standards in order to meet their needs.

2) Application architects who will find here material that will guide them throughout the difficult process of designing a system that fully and properly satisfies all the business and legal/regulatory requirements specific to electronic signatures, and who will gain a better understanding on how to select the proper standards to be implemented and/or used.

3) Developers of the systems who will find in this document an understanding of the business driven approach underlying the decisions made by the business managers and application architects on the scoping parameters to be used when creating and validating electronic signatures in the concerned business processes, as well as a proper knowledge of the standards that exist in the field and that they must know in detail for a proper development.

4) Signature policy issuers who will find in this document a guidance on the decision-making process for specifying the constraints to be imposed when creating, preserving/updating and validating electronic signatures within a specific context.

NOTE: A signature policy document is a declaration of the practices and rules (to be) used when creating, preserving and validating electronic signatures in a specific context (e.g. business process) and is usually a document resulting from the execution of the guided implementation approach described in the present document. It is recommended to use he standardised table of contents provided in ETSI EN 319 172 [i.10] as a way to document the various decisions taken while executing the business driven electronic signature implementation process for which guidance is provided in the present document. At the end of this iterative process, it will help to finalise and formalise the declaration of the practices and rules (to be) used when creating, preserving and validating electronic signatures in the concerned specific context (e.g. business process) into such a standardised signature policy document.

Clause 4 contains an introduction to the guided implementation process, including advices on how to read the present document based on the reader's profile, and an overview of the guided implementation process and its phases highlighting the rationale behind each one.

Clause 5 presents the first phase of the guided implementation process, emphasizing the imperative need of developing a proper and as much complete as possible business analysis of the business requirements driving the need for implementing electronic signatures, as a way to ensure that all the details relating to crucial aspects of the involved business processes are actually well captured and that the implementation of electronic signatures does not miss any of them. It also emphasizes the need of conducting a risk analysis, as a way of getting the needed information from which policy and security requirements are identified, so that once they are satisfied, stakeholders are sure that the implementation of electronic signature is done in such a way that it actually counters the identified risks.

Clause 6 presents the second phase of the guided implementation process, namely the proper management of the complete set of requirements imposed by different sources.

Clause 7 presents the third phase of the guided implementation process. It provides material that guides the readers to properly identify and understand the relevant business scoping parameters coming from different sources.

Clause 8 presents the fourth phase of the guided implementation process. It aims, in essence, at guiding the readers in deciding the technical means to be used for implementing electronic signatures in a way that fulfils the entire business context related requirements identified in the previous phases, and what standards are best suited for this. As such, this clause is specifically addressed to readers with a technical profile more than to readers with a management oriented profile.

Clause 9 provides some hints of a set of tools related with testing interoperability and conformance, which implementers may use for assessing the conformance of their implementations to the referenced standards and also their interoperability with other implementers' tools.

Clause 10 provides some hints on the evaluation process to which very likely the implementations need to pass by regulatory legal or quality assurance imperative.

Clause 11, as a way of corollary of this guide, summarizes the relationships existing between each step of the proposed guided implementation process and different documents present within the Standardisation Framework [i.1].

2 References

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

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

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

2.1 Normative references

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

Not applicable.

2.2 Informative references

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

EXAMPLE:

[i 1] ETSI TR 119 000: "Rationalised Framework for Electronic Signature Standardisation"

[i.2] ETSI EN 319 122: "CMS Advanced Electronic Signatures (CAdES)"

[i.3] ETSI EN 319 132: "XML Advanced Electronic Signatures (XAdES)" [i.4] ETSI EN 319 142: “PDF Advanced Electronic Signatures (PAdES)”

[i.5] ETSI EN 319 152: “Advanced Electronic Signatures in Mobile Environments”

[i.6] ETSI EN 319 162: “Associated Signature Containers (ASiC)”

[i.7] ETSI EN 319 102: “Procedures for Signature Creation and Validation”

[i.8] EN 319 101: “Policy & Security Requirements for Signature Creation Applications and Signature Validation Applications”

[i.9] ETSI EN 419 111: “Protection Profiles for Signature Creation & Validation Applications”

[i.10] ETSI EN 319 172: “Signature Policies”.

[i.11] ETSI EN 319 103: “Conformity Assessment for Signature Creation & Validation Applications (& Procedures)”

[i.12] ETSI TS 119 104: “General Requirements for testing Compliance & Interoperability of Signature Creation and Validation”

[i.13] ETSI TS 119 124: “CAdES Testing Compliance and Interoperability”

[i.14] ETSI TS 119 134: “XAdES Testing Compliance and Interoperability”

[i.15] ETSI TS 119 144: “PAdES Testing Compliance and Interoperability”

[i.16] ETSI TS 119 154: “Testing Compliance and Interoperability of AdES in Mobile Environments”

[i.17] ETSI TS 119 164: “ASiC Testing Compliance and Interoperability”

[i.18] ETSI TS 119 174: “Testing Compliance and Interoperability of Signature Policies”

[i.19] ETSI TR 102 045: “Signature Policy for extended business model”

[i.20] CROBIES WP 5-1: “ Guidelines and guidance for cross-border and interoperable implementation of electronic signatures. WP 5-1”

[i.21] ETSI TR 119 200: “Business Driven Guidance for Signature Creation and Other Related Devices”

[i.22] ETSI TR 119 300: “Business Driven Guidance for Cryptographic Suites”

[i.23] ETSI TS 119 312: “Cryptographic Suites for Secure Electronic Signatures”

[i.24] ETSI EN 319 602: “Trust Service Status Lists Format”

[i.25] ETSI EN 319 612: “Trusted Lists Format”.

[i.26] IETF RFC 5280: "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile".

[i.27] ETSI TS 119 001: "Electronic Signature Infrastructure; Definitions and abbreviations.

3 Definitions and abbreviations

3.1 Definitions

For the purposes of the present document, definitions in TS 119 001 [i.27] apply with in particular the following definitions being imported in the present document for the sake of reader's convenience:

business scoping parameter: is a specific parameter scoped in the light of the business process(es) where electronic signatures or trust services are going to be implemented, which implementers need to take into consideration for appropriately addressing the related business requirements in their implementation.

enveloping (electronic) signature: respect the signed data object, is an electronic signature that embeds this signed data object.

enveloped (electronic) signature: respect the signed data object, is an electronic signature that is embedded within this signed data object.

detached (electronic) signature: respect the signed data object, is an electronic signature that is neither enveloping nor enveloped with respect this signed data object.

NOTE: This may contain additional information.

3.2 Abbreviations

TSP	Trust Service provider
AdES	Advanced Electronic Signature
BPMN	Business Model and Notation
BSP	Business scoping parameter
CAdES	CMS Advanced Electronic Signature
DA	Driving Application
ISMS	Information Security Management System
PAdES	PDF Advanced Electronic Signature
PAdES-2	PAdES signatures conformant to PAdES Part 2
PAdES-3	PAdES signatures conformant to PAdES Part 3
PAdES-LTV	PAdES signatures conformant to PAdES Part 4
PAdES-5	PAdES signatures conformant to PAdES part 5
PAdES-5-XML	PAdES signatures conformant to PAdES part 5 Profiles for " XAdES Signatures of XML documents embedded in PDF containers"
PAdES-5-XFA	PAdES signatures conformant to PAdES part 5 Profiles for "XAdES signatures on XFA Forms"
PAdES-NoXML	PAdES signature conformant to PAdES parts 2, 3 or 4.
SCA	Signature Creation Application.
SCDev	Signature Creation Device
SSCD	Secure Signature Creation Device
SVA	Signature Validation Application.
XAdES	XML Advanced Electronic Signature
ASiC	Associated Signature Containers
TL	Trusted List
TSL	Trust Service Status List
UML	Unified Modelling Language

4 Introduction to the guided implementation process

The present document is one of a series of guidance documents on selection standards and options for implementing electronic signatures and/or trust services. All these documents share a general approach, suitably profiled and developed by each one. This general approach starts from a pre-required analysis of the business requirements and involves the analysis of business scoping parameters specific to each area of standardisation. These scoping parameters are essential elements to be addressed and for which business driven choices need to be made facilitating the selection of the appropriate standards and their options in a way which meets, as far as possible, the business requirements.

The present document proposes a business driven guided process for implementing generation and validation of electronic signatures in business electronic processes.

4.1 How to use this document

The present document specifically addresses the implementation of electronic signatures, in particular generation and validation of electronic signatures. Any other aspect within other areas related to the implementation of electronic signatures (like cryptographic devices, cryptographic suites, supporting TSPs, etc.) is out of its scope. Nevertheless, it addresses readers to the suitable guidance documents within the Rationalised Framework that deal with other areas.

The present clause provides some suggestions on how to read the present document depending on the reader's profile (business managers, application architects, developers, and signature policy issuers).

1) Business managers should read until clause 7 included. These clauses are the part of the process that aims at describing at a high level the conditions and rules under which electronic signatures will be used within a business or application domain and process. These clauses focus on areas that are familiar to business managers, i.e. business processes modelling, risk assessment, business requirements, regulatory/legal framework requirements, policy and security requirements, business rules and Business scoping parameters, which will jointly condition the actual implementation of electronic signatures within the business.

2) Application architects and developers should read the whole document. They will find within clause 8 material specifically addressed to technical profiles providing guidance on how to use the standards within the area 1 of the Rationalised Framework for implementing generation and validation of electronic signatures in a way that fulfils the requirements covered during the previous phases of the guided approach.

3) Signature policy issuers should read the whole document. A signature policy document is a declaration of the practices and rules (to be) used when creating, preserving and validating electronic signatures in a specific context (e.g. business process) and is usually a document resulting from the execution of the implementation process described in the present document. It is recommended to use the standardised table of contents provided in ETSI EN 319 172 [i.10] to document the various decisions taken while executing the business driven electronic signature implementation process for which guidance is provided in the present document. At the end of this iterative process, this will help to finalise and formalise the declaration of the practices and rules (to be) used when creating, preserving and validating electronic signatures in the concerned specific context (e.g. business process) into such a standardised signature policy document.

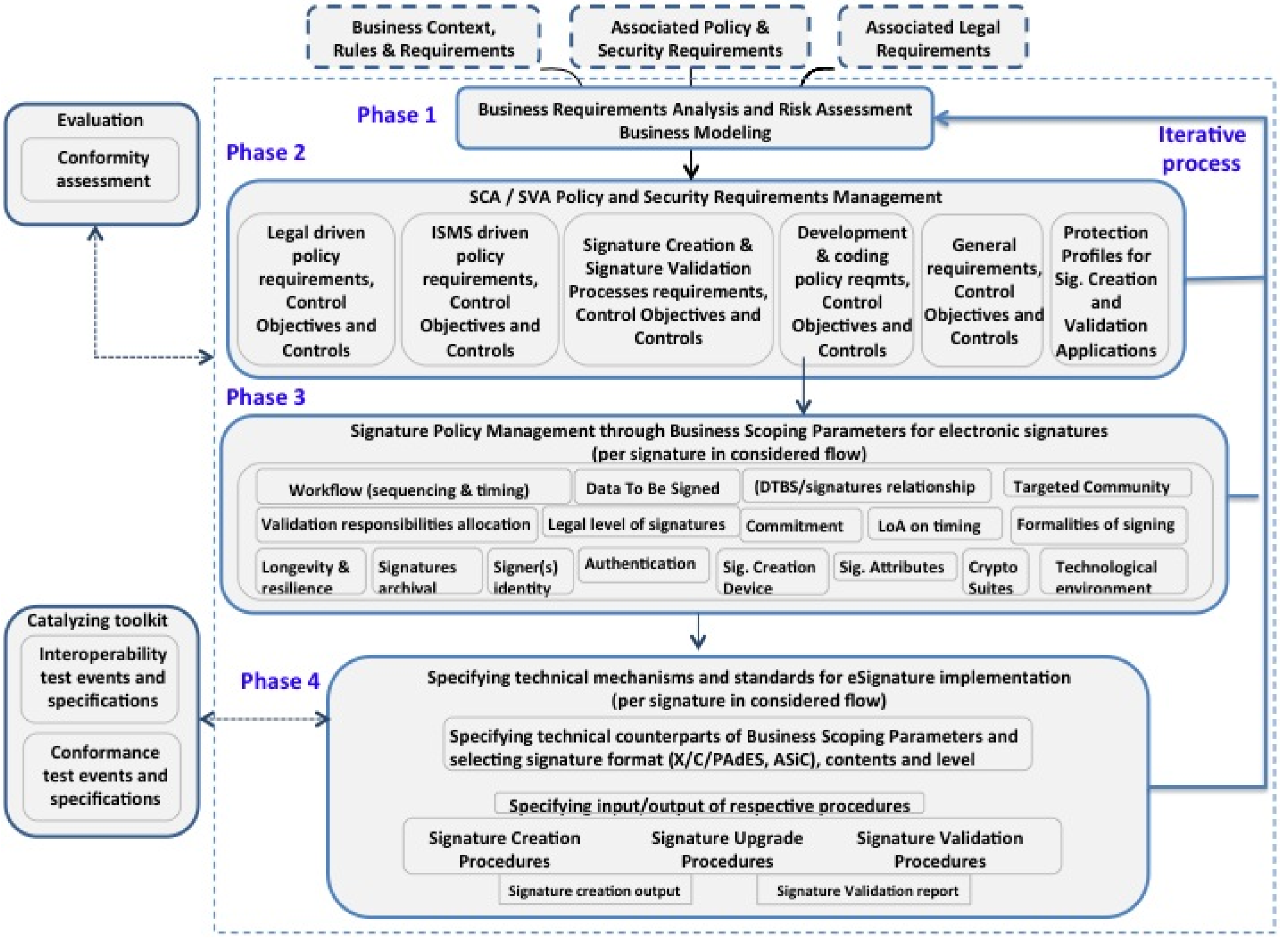
4.2 An overview of the guided implementation process

The present clause aims at providing a summary of the guided implementation process proposed within this document and also at briefly uncovering its relationships with other relevant guidance documents within the Rationalised Framework [i 1].

The figure below graphically summarizes the most relevant phases of the guided implementation process. It also shows two relevant elements, which may have a great impact, despite the fact that they cannot be considered, strictly speaking, as being part of the process. These two elements deserve some words at the end of the present clause.

The proposed guided implementation process is likely to be iterative by nature, as indicated by the arrow that goes back from the last phase to the beginning. The present document does not make any consideration about the degree of completion of the different phases in each iteration, which is entirely left to the implementers.

Figure 1: Iterative process for implementing generation and validation of electronic signatures.


Business Context, Rules & Requirements
Associated Policy & Security Requirements
Associated Legal Requirements
Evaluation
Conformity assessment
Phase 1
Business Requirements Analysis and Risk Assessment Business Modeling
Iterative process
Phase 2
SCA / SVA Policy and Security Requirements Management
Legal driven policy requirements, Control Objectives and Controls
ISMS driven policy requirements, Control Objectives and Controls
Signature Creation & Signature Validation Processes requirements, Control Objectives and Controls
Development & coding policy reqmts, Control Objectives and Controls
General requirements, Control Objectives and Controls
Protection Profiles for Sig. Creation and Validation Applications
Phase 3
Signature Policy Management through Business Scoping Parameters for electronic signatures (per signature in considered flow)
Workflow (sequencing & timing)
Data To Be Signed
(DTBS/signatures relationship
Targeted Community
Validation responsibilities allocation
Legal level of signatures
Commitment
LoA on timing
Formalities of signing
Longevity & resilience
Signatures archival
Signer(s) identity
Authentication
Sig. Creation Device
Sig. Attributes
Crypto Suites
Technological environment
Catalyzing toolkit
Interoperability test events and specifications
Conformance test events and specifications
Phase 4
Specifying technical mechanisms and standards for eSignature implementation (per signature in considered flow)
Specifying technical counterparts of Business Scoping Parameters and selecting signature format (X/C/PAdES, ASiC), contents and level
Specifying input/output of respective procedures
Signature Creation Procedures
Signature Upgrade Procedures
Signature Validation Procedures
Signature creation output
Signature Validation report

As a pre-requisite to the present guided implementation process, implementation of electronic signatures should start with a proper, complete and as detailed as possible analysis of the business processes (description and modelling of complex business electronic processes) within which one or more electronic signatures need to be implemented. This aims to ensure that all the details related to crucial aspects of the business electronic process are actually well captured and that the implementation of electronic signatures does not miss any of them. It also includes a risk assessment, as a way of getting the needed information from which policy and security requirements are identified, so that once they are satisfied, stakeholders are sure that the implementation of electronic signature is done in such a way that it actually counters the identified risks. This document, however does not aim at providing a complete guide on these topics but at making readers aware of their relevance.

The second phase aims at elaborating the different sources of policy requirements and security requirements into controls' objectives, and controls to be implemented in the system. The present document does not aim at providing a complete guide on these topics; instead it makes readers aware of their existence and relevance and refers to ETSI EN 319 101 [i.8] that properly deal with these issues.

The third phase of the process aims, in essence, at properly addressing and analysing the essential business scoping parameters in the light of the context where is conducted the business in which electronic signatures have to be implemented. They will condition the whole implementation lifecycle from its inception to its deployment and maintenance. These parameters may actually come, from different sources:

- From the business e-process itself. These are business scoping parameters inherent to the particularities of the business electronic process in which electronic signatures have to be implemented. They are related to:
  - the data to be signed,
  - the relationship between the signatures and the data objects to be signed,
  - the workflow of the documents and signed documents that is required by the business e-process,
  - the requirements on the timing and sequencing of signatures generation and proof of timely generation,

    - the need that signatures have a certain degree of longevity and resilience to change,
    - the archival requirements imposed by the business e-process,
    - the specific community where the electronic signatures will be exchanged,
    - the fact that the business e-process might envisage the generation / validation of electronic signatures within mobile environment,
    - requirements established by the business e-process on privileges that a signer has to detent, and
    - the allocation of signature validation responsibilities, done by the business e-process.

- From the legal and/or regulatory framework where the business process is conducted. These are business scoping parameters not inherent to the particularities of the business process but consequence of the legal and/or regulatory framework where it is conducted. Lack of consideration of these parameters when defining the strategy for implementing electronic signatures would likely lead to implementations that do not properly satisfy what is established by the applicable legal and/or regulatory framework with all the negative consequences that this would bring. These Business scoping parameters include: the quality level that the legal/regulatory framework impose to certain signatures of certain business processes, parameters derived from what the legal/regulatory framework establishes with regards to the scope and purposes of signatures, parameters related to the formalities of signing, and those that come from requirements on the longevity and resilience to change of signatures.
- From the actor that actually generates the signature. These are business scoping parameters inherent to the actor, including his type (i.e. whether it is a natural person or a legal person), the type of the signing certificate owned by the signer owned by the signer, and the signer device.
- Other. These are business scoping parameters coming from a variety of sources. Some of them might require the introduction of additional information within the signatures not already introduced. Other might require restricting the cryptographic suites.

The three aforementioned phases collectively aim at describing the conditions under which electronic signatures will be used within a business or application domain and process, including the identification of the resulting electronic signatures flow that has to be considered in the context of:

- a specific business application domain and/or process, with its own context and requirements;
- its associated set of policies (e.g. corporate IT and security policies) including any existing signature policy to which the to be designed signature policy is subordinate;
- its associated legal requirements, and
- the associated risk assessment identifying risks for which electronic signatures can be a mitigation tool but also risks induced by the use of electronic signatures themselves in the business or application process.

The fourth phase of the process aims, in essence, at deciding at the technical level the means to be used for fulfilling all the business context related requirements that come from the business scoping parameteres identified in the previous phase, and what standards within the Rationalized Framework are best suited for this. More specifically in this phasp implementers will find guiding material that will help them in deciding:

- The formats, contents, forms, and levels of the electronic signatures.
- The technical procedures for generating, upgrading and validating electronic signatures.
- The protection profiles which their applications generating and/or validating electronic signatures will be compliant with.

The standardised table of contents for signature policy documents provided in ETSI EN 319 172 [i.10] is recommended to be used as a way to document the various decisions taken while executing the business driven electronic signature implementation process for which guidance is provided in the present document. At the end of this iterative process, it would help to finalise and formalise the declaration of the practices and rules (to be) used when creating, preserving and validating electronic signatures in the concerned specific context (e.g. business process) into such a standardised signature policy document, if required.

Implementers may also use a set of available catalysing tools for assessing the conformance of their implementations to referenced standards (and consequently speeding up their production). This includes technical specifications for conformance testing and interoperability testing, and events for testing interoperability and conformance. This usage is shown in the figure 1 as a bidirectional dotted line connecting this phase with the round rectangle showing these tools. These tools are presented in clause 9.

Finally, readers of the present document should also take into account that it is quite likely that the applications to be put in place need to pass an evaluation process in order to be compliant with the regulatory/legal framework in force for the business context. The figure 1 shows this fact as a bidirectional dotted line connecting the round rectangle showing the evaluation with the dotted square enclosing the process itself. Some hints on the evaluation process are given in clause 10.

5 Analysing the Business Requirements

An accurate and complete business analysis, covering the entirety of the electronic business processes conducted, is essential for implementing electronic signatures. Without such analysis is highly unlikely that the implemented solution effectively supports the electronic business as it would be expected by its business managers and sponsors.

As mentioned before, it is not necessary to wait until the completion of the business analysis to start with the next tasks. This analysis, very likely, will be distributed among different iterations. However, it is required to have completed it at the end of all the iterations, in order to ensure that the whole set of requirements have actually been captured. It is recommended that in a business with a certain degree of complexity this analysis include the production of a business model, as a way of capturing all its relevant aspects.

The present document does not provide any further recommendations neither on the techniques used for analysing the business nor on how to distribute their completion throughout the different process iterations, as these issues are not within its scope.

The present document does not provide further recommendations neither on the techniques used for modelling the business nor on how to distribute its production throughout the different process iterations, as these issues are not within its scope. However, it signals the existence of tools for building these models that implementers may take into account, namely the Unified Modelling Language (UML) and some extensions specifically devoted to build up businesses models, or Business Process Management and Notation (BPMN).

It is strongly recommended to conduct a risk assessment with regards to the usage of electronic signatures as part of a business electronic process scenario. It aims at identifying risks for which electronic signatures can be a mitigation tool but also risks induced by the use of electronic signatures themselves in the business or application process. Implementers should also identify the relevant outputs of such assessment to be considered as input to the next phase, i.e. the establishment of the Policy and Security Requirements for electronic signatures generation and validation applications, as well as for the business rules to be accomplished by the implementation of electronic signatures.

It is out of the scope of the present document to provide any further recommendation on risk analysis methodologies.

6 Managing the Policy and Security Requirements

The second phase of the proposed guided implementation process is the management of the policy and security requirements that applies to the business electronic process and to the aimed integration of electronic signatures within. This management includes the following tasks:

1) Identification of the relevant requirements imposed by different sources (among which the different policies in force within the business context).

2) Specification of the objectives to be achieved by the controls to put in place for satisfying the identified requirements.

3) Selection of the controls for achieving the aforementioned objectives

While identifying the relevant requirements, implementers should take into account all their possible sources. Below follows the list of these potential sources of requirements:

1) Policies within the applicable regulatory or/and legal Framework.

2) Policies concerned with the information security management of information technology risks (e.g. ISMS policies).

3) Specific processes for generating, upgrading and validating electronic signatures.

4) Development and coding of applications dealing with the generation, upgrade and / or validation of electronic signatures.

A complete set of these requirements is required as a precondition for the implementation of a solution that effectively supports the electronic business modelled.

The completion of this phase may be distributed among several iterations, and it may receive feedback from results and findings of ulterior phase.

Implementers are strongly advised to perform this task as specified by the EN 319 101 [i.8] “Policy & Security Requirements for Signature Creation Applications and Signature Validation Applications” [i.8]. This European Standard provides general security and policy requirements that should be considered when implementing Signature Creation Applications (SCA) and Signature Validation Applications (SVA).

7 Business scoping parameters for this Area

The present clause provides details of the third phase of the proposed guided implementation process, which aims at properly addressing and analysing essential business scoping parameters in the light of the results of the two previous phases with regards to the specific business aspects and requirements of the business process where the electronic signatures have to be implemented.

The business scoping parameters to be taken into account when implementing creation and validation of electronic signatures are grouped as follows and discussed in the next sub-clauses:

- parameters mainly related with the specific application or business electronic process,
- parameters mainly related with the regulatory/legal framework where the business must be conducted,
- parameters mainly related with the different types of signing entities, as well as
- other aspects that do not fall within the above three listed categories but are important to be addressed when implementing electronic signatures.

7.1 Business scoping parameters mainly related with the business process

When attempting to implement electronic signatures in a business context, a number of business scoping parameters purely inherent to this context need to be taken into account, otherwise the risk of deploying a system that does not properly support the business in one way or the other is extremely high. These business scoping parameters will condition the whole system lifecycle from its inception to its deployment and maintenance. They, in consequence, will highly impact in the selection of the right standards that deal with the direct management of electronic signatures, namely with: their generation, their formats, their contents, their relative placement and relationship, their placement with respect to the signed data object(s), their resilience to time (longevity) or to cryptanalysis advances, and their validation.

This clause enumerates and provides details of the business scoping parameters mainly related with the business process itself that have a direct impact in the selection of standards.

7.1.1 BSP (a): Workflow (sequencing and timing) of electronic signatures

It is not unusual that business processes deal with workflows where different documents are generated and signed (by one or several signatories) in different time instants and in a specific order that may or may not be changed. These inherent parameters of the workflow also have an impact in the selection of the suitable standards, and in consequence, implementers should take them into account. Below follow the most relevant ones:

- Whether the time when a signature was applied is relevant or not. For a deeper discussion see clause 7.1.1.2.
- For the not unusual situations where there are data objects that have to be signed by more than one signatory, implementers should take into account the following aspects:
    - Whether the order in which the signatures are applied is relevant or not. For a deeper discussion see clause 7.1.1.2.
    - Whether all the signatures sign the same (the data object to be signed) or something different (the data object to be signed and one or more signatures previously applied to it, or even only one or more previously applied signatures). For a deeper discussion see clause 7.1.1.1.

7.1.1.1 Multiple signatures

It is not unusual in business contexts that one data object requires more than one signature for having the required effect. In certain occasions this is actually required by the Legal or Regulatory Framework. When facing these situations, implementers should differentiate between:

- Parallel signatures. These are signatures applied exactly to the same data object(s). They are mutually independent. Implementers should, in the cases where this type of signatures is required, identify what parallel signatures are required by the business process and/or its regulatory or legal framework, and where they have to appear, for giving the signed data object(s) its full effect.
- Serial signatures. These are signatures applied to different data object(s) and whose order of generation is relevant. Implementers should, in the cases where this type of signatures is required, identify what serial signatures are required and what data object(s) each one should apply to. Implementers should clearly identify the order in which the different signatures have to be computed and where these signatures have to appear (sequencing of signatures is discussed within clause 7.1.1.2).
- Counter-signatures. These are a special type of serial signatures, used in business processes that establish that a certain signature does not have any effect unless it is signed in turn by another signature, usually generated by a certain entity entitled for conferring such an effect to the first one. When such type of signatures appear in the workflow, implementers should take into account:
    - The relative position of countersignature and countersigned signature. Most of signature formats allow embedding the countersignature within the countersigned signature. However, some formats also allow keeping them physically detached and still indicating that a certain signature is actually a countersignature of another signature.
    - The actual meaning of a signature's countersignature, as this could impact the type of commitment endorsed by the counter-signatory (see clause 7.2.2).
    - Whether there is the requirement of validating the to-be-countersigned signature before generating the countersignature.
    - Whether the counter-signatory is required by the business process to countersign only the previously existing signature(s), or sign these ones and the signed data object(s), or even to add additional data object(s) and also sign it (them).

Implementers should also take into account that complex business processes would likely require to manage combinations of the different signature types aforementioned. A clear differentiation of the signatures types in each combination is crucial for properly selecting the most suitable standards and mechanisms.

Implementers should also identify whether the business process is actually demanding bulk signing, i.e., generate a significantly high number of serial signatures, as this may have an impact on, among other things, requirements for using devices specially designed for these purposes (e.g. hardware security modules).

7.1.1.2. Timing and sequencing

Implementers should identify those constraints on the timing and sequence of signatures generation imposed by the business process and /or its regulatory or legal framework for giving to the documents and signatures its full effect.

These constraints may, depending on the business process, be of very different nature: a mere specification of a deadline for the generation of each signature, a mere specification of the order in which documents and / or signatures have to be generated, detailed ranges of allowed time periods between the occurrence of the aforementioned events, specification of the order in which the signatures have to be validated, etc.

Implementers should also take into account the actual scope of these constraints, as they could apply to individual signatures, individual documents, multiple signatures, or multiple documents, depending of the workflow defined for the business process.

Special care should be paid when the business process and/or its regulatory or legal framework requires capability to prove that certain documents and/or signatures had been generated before a certain given time instant, as the satisfaction of this constraint would lead to use time-stamping or time-marking techniques, significantly impacting the system being built. Should this be the case, implementers should carefully considering the level of assurance of the timing evidences (see clause 7.2.3).

Finally, implementers should also take into account any specific relationships that may appear between constraints in the sequencing of the generation of each signature and constraints established on potential roles/attributes to be held by its corresponding signer (see BSP (l) Identity (and roles/attributes) of the signer).

7.1.2 BSP (b): Data Object(s) to be signed

Implementers of electronic signatures in an application / business processes should clearly identify all the relevant aspects concerning to the data object(s) that have to be signed. These aspects include:

- The nature and the format of the data to be signed (e.g. binary, structured data, xml, PDF document, editable documents such as Word or ODF, multimedia packages, images, etc.). The type of format for the data object to sign may also be influenced by business risks or legal provisions, for example, when a specific provision is imposed on the formalities of signing (e.g. what you see is what you sign, see BSP(i)).

NOTE: At present, electronic signatures may be generated following XML, ASN.1 or PDF syntax. It is quite obvious to conclude that where the data to be signed are specified in one of the aforementioned syntaxes, a reasonable initial choice would be to select the electronic signature defined for that syntax, unless other business parameters clearly recommend to use another one.

- In those cases where the data object involved in a signing process is structured, it is worth to identify whether the whole data object or only certain part(s) have to be signed, as this is strongly related to the features offered by the different electronic signature formats and would impact the final choice.

7.1.3 BSP (c): Relationships of signatures with signed data object(s) and signature(s)

As mentioned before, implementers of electronic signatures in an application / business processes should also pay attention to the relationships between each signature and its corresponding signed data object(s) and other signatures in the workflow. More specifically, they should consider:

- The number of the data objects that one signature actually signs. While all the signature formats are able to deal with one data object without any additional manipulation, the generation of a signature covering more than one object requires the application of different techniques depending on the signature format ranging from manipulating the data objects to be signed, to just take advantage of native mechanisms within the signature format for dealing with this kind of situations.
- In special cases like bulk signatures (i.e. situations where there is a high number of data objects collectively signed by one signature) implementers should pay attention to the benefits of using referencing mechanisms

(like using signed ds:Manifest within XAdES signatures) which, in case of failure in the checks performed on some of the signed data objects, still would allow to affirm that the signature on the rest of the signed data objects is OK.

- The recommended (as per the application/business processes) relative position of the signed data object and its signature. Three different situations may appear:

  - The signature is part of the data object that it signs (enveloped signature henceforth)
  - The signature actually envelops the data object that it signs (enveloping signature henceforth)
  - Signature and signed data object are detached (detached signature henceforth)

  Also here the features offered by the different signature formats vary from one to the other, ranging from formats that by its own nature only cover one of the former situations, to formats that incorporate mechanisms for dealing with all of them.

  When one signature has to sign different data objects, the situation might become more complicated, as theoretically the application / business processes might require that the signature envelops some of the signed data object, and simultaneously be enveloped by another one and even be detached from others signed data objects. Although these so highly complex situations are not likely to be frequent, they should not be discarded by principle.

7.1.4 BSP (d): Targeted community

Implementers should clearly identify the community each document and its (their) signature(s) is (are) addressed to. Once this has been done, the implementers should identify any specific community rules in place. These rules could, for instance, state the conditions under which a certain signature may be relied upon, or include provisions relating to the intended effectiveness of signatures, where multiple signatures are required. These rules could greatly impact not only the formats of the signatures and their relationships with the signed documents, but also the specific standards and/or profiles to be used.

7.1.5 BSP (e): Allocation of responsibility of signatures validation and upgrade

When analysing the management of electronic signatures within business processes, implementers should pay attention to the allocation of the responsibility of validating such electronic signatures. Implementers should clearly distribute this responsibility among the following entities, according to the specificities of the business process:

1) Party relying on the signature. Although this is a common allocation, implementers should not assume that this would always be most suitable one. In certain occasions it would merely be impractical or even too expensive. In consequence in certain scenarios it could be better to assign this responsibility to a subset of parties taking part of the transaction.

2) Electronic Signature Validation Trusted Services. This alternative would release the different relying parties of all the complexities associated with the validation of electronic signatures and allocate them to specialized services conveniently supervised and/or accredited, ensuring the suitable level of trust in the validations performed.

3) Business processes where countersignatures are generated, could impose that counter-signing parties are required to perform a validation of the signatures to be counter-signed before actually countersigning them, as part of the data flow.

These three types of allocations are not necessarily exclusive, being it possible that some of them coexist within complex business processes.

Upgrading electronic signatures is a co-lateral process to the validation of electronic signatures. This is the process by which certain material (e.g. time-stamps, validation data and even archival-related material) is incorporated to the electronic signatures for making them more resilient to change or for enlarging their longevity. Implementers should, in consequence, also identify requirements for upgrading electronic signatures as they are validated and progress in the business process data flow.

7.2 Business scoping parameters mainly influenced by legal/regulatory framework where the business process is conducted

The following BSPs may not strictly be influenced by legal provisions only but may also be driven by business considerations inherent to the concerned business process and its expectations with regards to the type of evidences resulting from the implementation of electronic signatures.

7.2.1 BSP (f): Legal level of the signatures

For each signature identified in the concerned workflow, implementers should specify the signature legal level required in the context of the business process and the associated legal/regulatory requirements.

This parameter has an impact on the level of assurance on the authentication (i.e. the certification of the identification) of the actor generating an electronic signature, on the class and policy requirements on the TSP providing such level of assurance, on the class of signature creation device used by such actors, on the use of a specific trust model for TSP issuing certificates (e.g. Trusted Lists, specific Trust Anchors in PKI hierarchy, use of CA certificate stores).

NOTE: The following levels are identified in accordance with Directive 1999/93/EC, CD 2009/767/EC and CD 2011/130/EU: qualified electronic signatures (QES), advanced electronic signatures supported by a qualified certificate ($AdES_{QC}$), and advanced electronic signatures (AdES).

7.2.2 BSP (g): Commitment assumed by signer

Implementers should identify and describe the expected purpose of each signature and hence the meaning and the precise nature of the responsibility assumed by signing, or in other words the type of commitment for each electronic signature in the considered business scenario and identified electronic signature(s) flow. The description of such electronic signature commitment types may be useful for avoiding potential ambiguity due to the fact that electronic signatures may not provide equivalent contextual information as in the paper world leading to uncertainty about the signer’s intention.

In particular, there is a need to be able to distinguish between:

- electronic signatures intended for data authentication purposes only,

NOTE: The generation of electronic signature for which the expression of the intention to sign is limited to ensure the authentication of the data to which it is associated (signed data object(s)) will serve the same purpose towards natural person signers while being electronic signatures in essence: electronic signatures created as the equivalent of a handwritten signature but not to indicate a will or intention to be legally bound by the content of the data which is signed (this could be an intention to sign a draft, an acknowledgement of receipt, or to indicate authorship or responsibility for a document).

- electronic seals generated by legal persons,
- electronic signatures intended for entity authentication purposes only,
- electronic signatures created with the intention to sign the associated data (signed data object(s)):
    - as a draft,
    - as an acknowledgement of receipt,
    - as an intermediate approval as part of a decision process,
    - to indicate authorship or responsibility for a document (signed data),
    - to indicate having reviewed a document (signed data),
    - to certify that a document is an authentic copy,
    - to indicate witnessing of someone else signature on the same document (signed data)

- having read, approving and being bound accordingly to the content of the data object that is signed
- etc.

and being, as a signatory, bound by the content of the data object that is signed.

7.2.3 BSP (h): Level of assurance of timing evidences

For each signature identified in the concerned workflow (see BSP(a)) implementers should describe and specify the requirement on the level of assurance on the required timing evidences. This component is closely related to the components BSP(a), (j) and (k).

Implementers should distinguish between claimed assertions with regards to time information, and trusted time evidence, such as time-stamps provided by trust service providers issuing time-stamp tokens or trusted time-marks.

When trusted time evidence are required, implementers should consider the requirements and level of assurance associated respectively to the time-stamp tokens and the providers, and on which type of information the time-stamp tokens are generated (e.g. time information only, signed data object(s), signature(s), signature(s) and validation data, etc.).

7.2.4 BSP (i): Formalities of signing

One of the most important characteristics of a signature is the manner of its creation. Often referred to as the "ceremony of signing", it is the way the attention of the signer is drawn to the significance of the commitment she is undertaking by performing this act of signing.

Implementers should identify requirements on any type of evidence of the will or intention to sign that would have an influence on the manner the electronic signature is created. Implementers should also specify how the act of signing is presented to the signer in order to draw signer’s attention to the significance of the commitment he is undertaking under the electronic signing process.

Such requirements are likely to imply the signer interface to be designed in a way to guarantee, to the extent possible, a valid legal signature environment. Below follow some ideas:

1) Provide users with a “What You See Is What You Sign” environment.

2) Provide users with proper advice and information on the application’s signature process;

3) Provide users with proper advice and information on the legal consequences.

4) Design the user interface in a way to guarantee, to the extent possible, a valid legal signature environment, including:

   - Provision to the user of clear information about the application’s signature process and legal consequences;

   - Implementation allowing and demonstrating clear expression of a will to sign and the user’s intention to be bound by the signature;

   - Implementation allowing and demonstrating an informed consent;

   - Consistence between the use of the appropriate signature creation and verification data, signature creation device, the data to be signed and the expected scope and purpose of the signature (or the act of signing);

This BSP may impact the selection of appropriate protection profiles and conformity assessment schemes against which the signature creation application will be designed and assessed.

7.2.5 BSP (j): Longevity and resilience to change

It is not unusual that certain business processes and/or their regulatory or legal framework require that signatures have a certain longevity, being it possible in certain occasions that the implied elapsed time since their generation until their potential re-validation is of a certain number of years.

Time passing has two different effects on the electronic signatures: firstly the validation material used for generating and validating them (certificates) may expire or even not be available anymore; secondly, the cryptographic algorithms (also including digest algorithms) may become weak as cryptology techniques and computer capabilities improve.

Longevity and resilience to change (understood as the resistance of electronic signatures to the uncovering of weaknesses of their algorithms) are in consequence strongly related to each other.

Implementers should identify those signatures whose re-validation is required some time after their generation, as well as the time period during which their re-validation has to be made possible. These factors will help implementers in making right decisions when planning the means to be put in place for ensuring the required longevity of the signatures.

7.2.6 BSP (k): Archival

Archival is related with the longevity of the signatures. Regarding this issue, implementers should identify requirements on the archival of the signed data objects, their signatures and the material used for their validation, including requirements on whether archiving them together or not.

Implementers should respect the prerequisites of electronic archiving from the early stages of the design of new developments as well as when integrating electronic signature solutions in current products. This aims to ensure proper implementation of electronic archiving where it is legally recognized and facilitate compliance with future regulations applicable on electronic archival.

7.3 Business scoping parameters mainly related to the actors involved in generating the signature

7.3.1 BSP (l): Identity (and roles/attributes) of the signer

In most cases, a signature is worthless if it cannot be attributed to the purported signer. Implementers should identify and specify:

1) who are the anticipated signers,

2) the associated signer identification rules,

3) if any, the rules applicable to the roles and/or attributes of the signers, as well as

4) if any, the requirements on an associated proof of authority.

They should, in consequence, identify and describe what are the necessary elements to ensure that a signature is that of a specified individual (whether a physical or legal person, a business or transactional functional entity, a machine, an application or server, etc.), i.e. what is the required identification element (identity attributes) for each type of signer. For instance where a contract names an individual as a party to be bound by its terms, what is required as signer identification elements; names, date of birth, unique identification number, etc.

In some business scenarios, the role or attributes of a signer are at least as important as his identity. Under these circumstances, the term “signer role” does not refer to the “signing” role played by the signer in the electronic signature supported business process (e.g. primary signature, countersignature) but relates to roles such as “official representative of a legal person” or “sales director”, which may be claimed or certified, but which implies some attribute(s) associated with the signer. Implementers should describe the set of attributes, authorities and responsibilities which are associated with each signatory, his access rights, or authority to sign, to act on behalf of the organization he purports to represent, etc.

Implementers should state the type of proof of authority to sign that is acceptable. This may include, among others:

1) proof that an employee or representative is authorized to enter into transactions over a specified value,

2) proof that delegation to sign has been authorized.

7.3.2 BSP (m): Level of assurance required for the authentication of the signer

Implementers should identify what is the level of assurance required for the authentication for the signer in each signature to be generated within the business process, i.e. what are the expectations in terms of trust on the signatory identification (e.g. quality level of certificate). For instance, certificates may be required to be qualified certificates and/or issued by an accredited, supervised, certified, or audited certification authority, or be issued according to a specific Certificate Policy, etc.

This, very likely, will not impact the specific contents of the signature itself but the signing application; nevertheless, a failure in reaching the level required by the legal/normative framework would lead to the potential rejection of the signatures in case of auditing or dispute.

7.3.3 BSP (n): Signature Creation devices

Implementers should also identify any existing requirement on the signature creation devices that will be used for generating the signatures within the business process, in order to ensure their fulfilment. Again, a failure to satisfy these requirements would lead to the potential rejection of the signatures in case of auditing or dispute.

7.4 Other Business scoping parameters

The present clause addresses business scoping parameters that are not mainly related either to the business process, the legal/regulatory framework, and the signatory

7.4.1 BSP (o): Other information to be included within the signatures

Implementers should indicate any other applicable signature attributes, such as :

- Geographic location where the signature was created: This may be an example of such a specific signature attribute as the location or jurisdiction in which the signature was made, might have legal consequences in the event of a dispute, in determining where the dispute should be heard/subject to the laws of which jurisdiction it should be.
- Claimed signing time: Another example of applicable signature attributes may be the signer's claim on the time at which he generated the signature. This is only to be considered as a claim and should not be considered as trusted unless the corresponding time is provided as a the result of a trusted time service provided by a Trusted Time-stamping Service Provider.
- Content time-stamp.
- Content related information (e.g. its type).
- Signer’s role(s) and/or specific qualifications attributes.

7.4.2 BSP (p): Cryptographic suites

Implementers should describe and specify requirements on the robustness of cryptographic suites used to generate or upgrade each electronic signature in the concerned business process. Implementers should carefully read the TR 119 300: “Business Driven Guidance for Cryptographic Suites” [i.22], the guidance document that specifically addresses area 4 (Cryptographic Suites) of the Rationalised Framework for Electronic Signature Standards, and where they will find guidance on how to select the cryptographic suites that properly fulfil the aforementioned requirements.

7.4.3 BSP (q): Technological environment

From the business process specification, implementers should also pay attention to the technological environment where the data objects to be signed and the signatures will be managed, as this may have an impact on a number of technological decisions to be made, among which the signature formats to be used.

In particular it is suggested to identify whether it is required (or even could be required in a future) allowing that the generation and/or validation of certain signatures applied to certain document to be done, not only in classical environments, but also within mobile environments. In case this requirement exists, implementers should clearly identify which type(s) of document(s) and which signatures within them need to also be managed within mobile environments. This is extremely relevant, as the mobility aspect may require making use of specific services for supporting these tasks, and in consequence, to use specific sets of standards.

8 Selecting the most appropriate standards and options technical mechanisms

The European Rationalised Framework of Standards for Electronic Signatures includes standards defining three electronic signature formats:

1) CAdES (defined in the EN 319 122 [i.2] multi-part document),

2) XAdES (defined in the EN 319 132 [i.3] multi-part document),

3) and PAdES (EN 319 142 [i.4]} multi-part document).

It also includes one standard defining a container able to embed several data objects and detached electronic signatures that selectively sign some of them: the ASiC container (defined in the EN 319 162 [i.6] multi-part document).

NOTE: Readers should take into account that when making references to specific parts of XAdES, PAdES, CAdES and ASiC specifications, the present document uses the clauses numbering of the EN 319 1X2 under production (and not distributed for public commenting yet), which differs, in most of the cases, from the numbering implemented in the corresponding ETSI TSs. Nevertheless, whenever this occurs, the text within the present document makes it easy to identify what is the relevant part of the aforementioned specifications the text is referencing, and in consequence, it is not difficult to identify the referenced material even in the aforementioned ETSI TSs.

8.1 Format of signatures: CAdES, XAdES or PAdES

The suitable format of signature strongly depends on the business process itself. Under certain circumstances it clearly makes one option much better suited than the others. Under other circumstances, though, the advantages of a choice among other choices are not so clear and even arguable.

This clause lists some considerations that implementers may use when they need to decide the format(s) of electronic signatures to be implemented in their business processes.

However, it is worth to address first PAdES signatures as they represent a special case, as they actually are built on different formats. PAdES signatures conformant to PAdES specification part 2, build on CMS signatures. PAdES signatures conformant to PAdES parts 3 and 4 build on CAdES signatures. Finally, PAdES signatures conformant to PAdES specification part 5, build on XAdES signatures. PAdES part 5 defines two profiles groups: one for XAdES signatures on XML documents embedded within PDF containers, and another one for XAdES signatures on XFA forms.

Henceforward the acronym PAdES will be used in sentences that apply to signatures conformant to any PAdES specification part. PAdES-NoXML acronym will be used in sentences that apply only to signatures conformant to PAdES parts 2, 3 and 4 indistinctly. PAdES-n, with n being 2, 3 or 4 will be used in sentences that apply only to signatures conformant to the indicated part of PAdES specifications. PAdES-5-XML will be used in sentences that apply only to the PAdES part 5 profiles for XAdES signatures on XML documents embedded in PDF containers. PAdES-5-XFA will be used in sentences that apply only to PAdES part 5 profiles for XAdES signatures on XFA forms. PAdES-5 will be used in sentences that apply to a PAdES signature conformant to any of the profiles specified in PAdES part 5.

8.1.1 Format of the document

This is one of the first elements that implementers have to take into account. In principle, the closer the formats of signatures and documents are, the better.

Under this perspective, for XML documents, XAdES signatures would be the natural option.

Also in principle PAdES-NoXML signatures would be the natural option for embedding electronic signatures within PDF documents. PAdES-5-XFA would be the natural option for signing XFA forms, and PAdES-5-XML would be the natural option for signing XML documents that are embedded within a PDF container.

CAdES is also in principle the natural option for signing data objects whose structure has been defined in ASN.1, and that have been encoded in DER or BER.

For other binary formats, both XAdES and CAdES would initially work properly. Nevertheless, depending of the specific business process, one format could present advantages that would make that format more advisable. Implementers should, in consequence, analyse at least the aspects that are mentioned in subsequent clauses.

Despite what it has been said before, there are a number of additional considerations that modulate the former assertions and even, under certain circumstances, could fully justify selecting a signature format not considered initially as “the natural option”.

These considerations are discussed in subsequent clauses 8.1.2 and 8.1.3.

8.1.2 Relative placement of signatures and signed data objects

This clause provides information on how the different formats may manage different combinations with regards to the relative placement of signatures and signed data objects.

In essence, one may distinguish 3 pure relative placements of signatures with regards to where the signed data objects may appear: enveloped, enveloping and detached signatures. It is not unusual that a certain business process actually requires some form of combination of these placements (for instance, the business process may require that one of the signatures of a signed data object is enveloped by the object, while it also requires that another signature is actually detached or even enveloping the signed data object). Under these circumstances, implementers should carefully analyse the features provided by each format and also consider the potential benefits that a packaging mechanism like the one provided by ASiC could bring to the solution.

8.1.2.1 Enveloped signatures

PAdES-NoXML signatures are, by their own document-centric nature, enveloped signatures, i.e., they are embedded within the PDF document they sign. Also PAdES-5 signatures may be embedded within the object they sign.

CAdES signatures may be embedded within objects whose structure is defined in ASN.1 as long as this structure defines fields for embedding them. However, neither CMS nor CAdES specifications specify what exactly they actually sign under these circumstances. This means that very likely the scope of the signatures has to be specified separately, when specifying the syntax and semantics of the signed data object itself. In terms of implementation, this means that an application claiming conformance against CAdES would require additional software for scoping what the CAdES signature is actually signing if it is embedded within an ASN.1-defined object.

XAdES signatures may be embedded within XML documents. Unlike CAdES, XAdES inherits the XML Signature mechanisms for explicitly referencing any signed data object, and in consequence, a standardized way of retrieving such data objects (the `ds:Reference` element). This referencing mechanism allows to explicitly referring to (and actually sign) the whole XML document or only parts of it. The important consequence is that any XAdES application based on another one claiming conformance against XML Signature W3C Recommendation does not require any additional software for scoping what the signature is actually signing.

8.1.2.2 Enveloping signatures

PAdES-NoXML signatures are not allowed to envelop the data object they sign document they sign, by their own document-centric nature. However, PAdES-5-XML may envelope the data object they sign.

CAdES signatures, as they are built on CMS signatures, may envelop the signed data object, by encapsulating it within the `encapContentInfo`'s `eContent` field. CAdES applications built on applications claiming conformance to CMS do not require additional software for scoping what the signature is actually signing.

XAdES signatures may also envelop the signed data object. When this is a binary object, it is previously base64 encoded, which increases its size, and encapsulated within a `ds:Object` element. XAdES applications built on applications claiming conformance against the XML Signature W3C Recommendation do not require additional software for scoping what the signature is actually signing.

8.1.2.3 Detached signatures

PAdES-NoXML signatures are not allowed to exist detached from the PDF document they sign, by their own document-centric nature. However, PAdES-5 may be detached from the data objects they sign.

CAdES signatures may be detached from the signed data object, by leaving the `encapContentInfo`'s `eContent` field empty. However, neither CMS nor CAdES incorporate mechanisms that make it explicit any hint on how to retrieve the detached signed data object.

XAdES signatures also may be detached from the signed data object. Unlike CAdES, XAdES inherits the XML Signature mechanisms for explicitly referencing any signed data object, included the detached ones, and in consequence, a standardized way of retrieving such data objects. This has the important implication that any XAdES application built on an application claiming conformance to XML Signature W3C Recommendation is able to retrieve the detached signed data object in a standardized way.

8.1.3 Number of signatures and signed data objects

One of the elements to be also taken into account when specifying the signature format to be implemented is the cardinality of the relationship between signed data objects and its (their) signature(s). Different situations may appear, depending on the business case, which are explored in sub-clauses below.

8.1.3.1 One document is signed by only one signature

The three formats deal well with this situation.

8.1.3.2 One document is signed by more than one signature

When one document requires to be signed by more than one signature, implementers should take into account a number of considerations that are presented below.

Any PAdES-NoXML signature signs any other PAdES-NoXML signature already present within the document when it is created: they are always serial signatures; no PAdES-NoXML parallel signatures are allowed. More than one PAdES-5 signature may be used for signing the same data object. In addition to that, as they are XAdES signatures, any combination of parallel and serial signatures is allowed.

As CAdES signatures build on CMS signatures, they also incorporate within its specification native means for managing parallel signatures on one data object. CMS and CAdES signatures may also incorporate countersignatures as an unsigned attribute, which allows a sequence of countersignatures on one of the parallel signatures. However, arbitrary combinations of parallel and serial signatures are not easily implementable, as CMS and CAdES lack mechanisms for explicitly referencing signed data objects, and in consequence, applications should be configured for properly managing each specific combination.

XAdES signatures inherit from XML Signatures their native mechanisms for explicitly referencing and processing the data objects they sign (including other XML or XAdES signatures). Additionally XAdES incorporates an unsigned property that encapsulates a countersignature (be it a XML Signature or a XAdES signature). This makes any XAdES application built on an application fully compliant with XML Signature W3C Recommendation inherently able to manage any number of signatures signing one XML document (completely or partially), with any combination of serial and parallel signatures, and without any restriction on the relative placement of signatures and the signed data object. However, unlike CAdES, no standard mechanism is defined within XML Signatures W3C Recommendations or XAdES specifications for placing together a set of parallel XAdES signatures. This requires additional specifications. At present there are several examples on how this may be achieved; below follows some of them:

1) Embed several XAdES signatures within a XML document, each one being a parallel signature of the document itself or certain parts of the document.

2) Define containers that specify elements where parallel XAdES signatures on the same data object are placed (like ASiC does, for instance).

Several XAdES signatures may also sign one binary data object. However, in this case, XAdES signatures may only sign the complete data object.

8.1.3.3 One signature is required to sign more than one data object

PAdES-NoXML signatures only sign a PDF container by their own document-centric nature. Anything that is within the PDF container is signed, but nothing else. PAdES-5 signature, being XAdES signatures, may sign more than one data object within the XML content of the PDF container. Additionally, PAdES-5-XML may also sign data objects that are outside the PDF container.

CAdES signatures are not able by their own, to sign more than one data object. This requires doing some previous work on the signed data objects or use CAdES within appropriate containers. Below follow some examples on how to achieve this:

1) Sign a multi-part MIME object.

2) Define an ASN.1 structure for the document to be signed allowing several occurrences of CAdES signature fields each one being a parallel signature of the document itself.

3) Define containers that specify elements where CAdES signatures on the same data object are placed (like ASiC does, for instance).

XAdES signatures incorporate native mechanisms for signing more than one data object. Additionally, the usage of signed ds:Manifest also allows that if the validation of the collective digital signature succeeds and some check of certain signed data objects fails, applications may still decide that the rest of the data objects are correctly signed and proceed with their processing. In other words, this mechanism allows that failures in some individual checks of the signed data objects do not invalidate the whole collective signature.

8.2 A container for packaging together signed data objects and signatures on the objects?

Certain business process could require facilitating the management of certain data objects and their detached signatures by packaging them together. Implementers should, under these circumstances, seriously consider the suitability of using ASiC containers.

An ASiC container may, in its more complex form, include several data objects and several signatures, detached from the aforementioned data objects, selectively signing some of them. Objects of any format are allowed. Also CAdES or XAdES signatures are allowed and even co-existing within the same ASiC container.

As it has been already mentioned, ASiC containers allow packaging together parallel XAdES signatures. As for CAdES, ASiC containers puts in place a mechanism that allows that one CAdES signature indirectly signs more than one detached data object. This means that ASiC containers provide mechanisms that allow overcoming limitations inherent to each format.

8.3 Core specification or profile?

So far only the so called “Baseline Profile” has been specified for XAdES, CAdES, PAdES electronic signature formats, and ASiC container.

Baseline profiles are meant to minimize the number of options in the usage of AdES signatures and ASiC containers and maximize interoperability. As such, its usage is compulsory in the context of the EU Services Directive, but may also be used in other business and government use cases, if the provided functionality is sufficient for satisfying their requirements. These profiles do not envisage the incorporation of references to the validation material in XAdES, CAdES and ASiC containers.

Implementers should in consequence, firstly check whether the business context, and the regulatory/legal framework explicitly require the usage of the Baseline Profile. If this is not the case, implementers should check whether the requirements imposed by the business process, and the legal/regulatory framework (including electronic signatures life-cycle management related issues) could be satisfied with the functionality provided by the Baseline Profiles. If so implementers should seriously consider the usage of such profiles. Otherwise, implementers should proceed to use the core specifications, deciding what specific contents should be incorporated to the signatures/containers as indicated in the present document.

8.4 Selecting the proper level of the signature

Where the legal/regulatory framework requires that electronic signatures have a certain legal level(s), implementers should put in place the corresponding technical mechanisms for ensuring that such a level(s) is (are) reached.

Implementers should take into consideration that for ensuring a certain legal level(s) for the signature(s), they have to ensure that the following elements fulfil the requirements corresponding to such a level(s):

1) The Signing Device,

2) The Certificate Provision,

3) The Independent Assurance on (2),

4) The Signature Cryptographic Suite,

5) The desired longevity of the signatures,

6) The Signature Application, and

7) The Independent Assurance on (6)

8.5 Mapping formalities of signing to the electronic domain

Implementers should ensure that the provided signing environment gives satisfaction to the right subset of ideas listed within clause 7.2.4 as applicable to the specific legal/regulatory framework and business process.

8.6 Satisfying timing and sequencing requirements

8.6.1 Satisfying sequencing requirements

As mentioned before, certain business processes may impose constraints in the order to be followed for generating signatures on specific data objects.

Although these constraints always apply to counter-signatures (it is obvious that a counter-signature will be generated after the counter-signed signature), they may also be imposed to parallel signatures. In this later case any specific requirement on their sequencing may lead to the addition of a generation time indication (see next clause) or even to the specification of their relative placement.

8.6.1.1 Including counter-signatures

AdES forms allow to counter-sign a specific AdES signatures. In all the cases, the counter-signatures may also be AdES signatures.

Implementers are referred to clause 6.2.7 of EN 319 132 [i.3] when implementing XAdES signatures. This format allows managing counter-signatures in two ways:

1) Embedded within the counter-signed signature. Implementers are referred to clause 6.2.7.2 of EN 319 132 [i.3]. It specifies `xades:CounterSignature` unsigned property, a container for a `ds:Signature` element which may be a regular XML signature or a XAdES signature counter-signing the embedding signature.

2) Not embedded within the counter-signed signature. This is achieved by setting the `Type` attribute of the counter-signature's `ds:Reference` element referencing the counter-signed signature, to a pre-defined value. This allows to effectively detaching both signatures while making it explicit that one is a counter-signature or the other. Implementers are referred to clause 6.2.7.1 of EN 319 132 [i.3].

Implementers are referred to clause 6.2.7 of EN 319 122 part 2, when implementing CAdES signatures, which specifies the `counter-signature` unsigned attribute, a container for a regular CMS or a CAdES signature counter-signing the embedding signature.

When PAdES signatures are used, implementers should take into account the following considerations:

1) Counter-signatures for PAdES-NoXML signatures are other PAdES-NoXML signatures added afterwards. They actually sign all the previously existing data within the PDF container, including signed data objects and any signature. Usage of the `counter-signature` attribute is not allowed.

2) PAdES-5 signatures allow the usage of the `xades:CounterSignature` unsigned property (clauses 4.2.6.1 and 5.2.5.1 of EN 319 142 [i.4] part 5).

8.6.2 Satisfying timing requirements

All the AdES electronic signatures provide containers including information of different nature about the time when the signature and/or the signed data objects have been generated. Implementers may:

1) Include within an electronic signature time-stamp token(s) on the data objects to be signed, before the signature is actually generated, in case it is required to prove that certain data object(s) to be signed had been generated before a certain given time instant

2) Include within an electronic signature an indication of the claimed signature generation time. This is understood as a claim made by the signer and as such is generally treated by the relying parties, i.e., it does not deserve, generally speaking, the same confidence as a trusted time indication like a time-stamp token generated by a Time-stamp service provider (unless the signer is an entity entitled for being trusted when claiming that time –a certain Registered Electronic Mail Management Domain could be an example).

3) Include within an electronic signature a time-stamp token on the signature generated. This proves that the signature was generated before the time indicated within the time-stamp token.

Sub-clauses below provide additional details of these mechanisms.

8.6.2.1 Time-stamping the data objects to be signed before signature generation

All the AdES electronic signatures provides mechanisms for including time-stamp tokens on the data objects to be signed before the actual signature is generated.

Implementers are referred to clauses 6.2.8.1 and 6.2.8.2 of EN 319 132 [i.3], when implementing XAdES signatures. The first clause specifies `xades:AllDataObjectsTimeStamp` signed property, a container for a time-stamp token that collectively time-stamps all the data objects referenced in the `ds:SignedInfo` element within the XAdES signature. Clause 6.2.8.2 specifies `xades:IndividualDataObjectsTimeStamp`, a container for a time-stamp token on some of the data objects referenced within the `ds:SignedInfo` element.

Implementers are referred to clause 6.2.8 of EN 319 122 [i.2] part 2, when implementing CAdES signatures, which specifies the `content-time-stamp` signed attribute, a container for a time-stamp token on the signed data object.

When PAdES signatures are used, implementers should take into account the following considerations:

4) PAdES-4 specifies the DocumentTime-Stamp dictionary, a special type of PDF signature dictionary that contains a time-stamp on the PDF document. Implementers are referred to clause A.2 of EN 319 142 [i.4] part 4.

5) PAdES-5 signatures make use of the optional `xades:AllDataObjectsTimeStamp` (clauses 4.2.5.7 and 5.2.4.7 of EN 319 142 [i.4] part 5) and `xades:IndividualDataObjectsTimeStamp` signed properties (clauses 4.2.5.8 and 5.2.4.8 of EN 319 142 [i.4] part 5).

8.6.2.2 Including claimed signing time

All the AdES electronic signatures provide mechanisms for incorporating as signed information, an indication of the claimed signing time. Implementers should also have in mind that this time, is not, in general, a trusted time.

Implementers are referred to clause 6.2.1 of EN 319 132 [i.3] part 2, when implementing XAdES signatures, which specifies the `xades:SigningTime` signed property.

Implementers are referred to clause 6.2.1 of EN 319 122 [i.2] part 2, when implementing CAdES signatures, which specifies the `signing-time` signed attribute.

When PAdES are used, implementers should take into account the following considerations:

1) Within PAdES-3 and PAdES-4 signatures, the claimed signing time, if required, will be indicated by the value of M entry of the signature dictionary (clause 4.5.3 of EN 319 142 [i.4] part 3).

2) Within PAdES-5-XML signatures, the claimed signing time, if required, will be indicated within `xades:SigningTime` signed property (clause 4.2.5.1 of EN 319 142 [i.4] part 5).

3) Within PAdES-5-XFA signatures, the claimed signing time, if required, will be indicated by the content of the `CreateDate` element defined within the XMP `ns.adobe.com/xap/1.0/` namespace (clause 5.2.4.1 of EN 319 142 [i.4] part 5).

8.6.2.3 Including time-stamp token on the signature

Implementers are referred to clause 6.3 of EN 319 132 [i.3] part 2, when implementing XAdES signatures, which specifies the `xades:SignatureTimeStamp` unsigned property.

Implementers are referred to clause 6.3.1 of EN 319 122 [i.2] part 2, when implementing CAdES signatures, which specifies the `signature-time-stamp` unsigned attribute.

When PAdES are used, implementers should take into account the following considerations:

1) PAdES-2 signatures may incorporate a time-stamp token as specified in ISO 32000-1 clause 12.8.3.3.1 (clauses 4.3 and 5.4 of EN 319 142 [i.4] part 2).

2) PAdES-3 signatures make use of the optional `signature-time-stamp` unsigned attribute (clause 4.5.2 of EN 319 142 [i.4] part 3)

3) PAdES-5 signatures make use of the optional `xades:SignatureTimeStamp` unsigned property (clauses 4.2.5.9 and 5.2.4.9 of EN 319 142 [i.4] part 5).

8.7 Including indication of commitments assumed by the signer

All the AdES electronic signatures provide mechanisms for indicating the commitment made by the signer.

Implementers are referred to clause 6.2.3 of EN 319 132 [i.3] part 2, when implementing XAdES signatures. The signed property `xades:CommitmentTypeIndication` uses URI values as the way for indicating the commitment made by the signer. The aforementioned clause lists a set of pre-defined URIs, each one corresponding to a specific commitment, whose semantics is precisely defined. Implementers should also take into account that as one XAdES signature may collectively sign different data objects, each commitment identifies the data object(s) it refers to.

Implementers are referred to clause 6.2.3 of of EN 319 122 [i.2] part 2, when implementing CAdES signatures. The signed attribute `commitment-type-indication` uses OID values as the way for indicating the commitment made by the signer. The aforementioned clause lists a set of pre-defined OIDs, each one corresponding to a specific commitment, whose semantics is precisely defined. This list identifies the same commitments as the list of URIs in EN 319 132 [i.3] part 2.

If ASiC containers are used implementers should include commitment indications in each CAdES and XAdES signature where their presence is required, using the aforementioned elements.

When PAdES signatures are used, implementers should take into account the following considerations:

1) Within PAdES-2, the commitments made by the signer, are identified by an array of strings, each one identifying one commitment, within the optional signed entry `Reason`, within the signature field seed dictionary. Implementers are referred to EN 319 142 [i.4] Part 2, clause 4.2 and ISO 3200-1 12.7.4.5 for further details.

2) Within PAdES-3 and PAdES-4 signatures, the commitments made by the signer are signalled in two different ways. Implementers are referred to EN 319 142 [i.4] Part 3, clause 4.5.8 for further details:

   - The optional signed entry `Reason` within the signature field seed dictionary if these signatures do not contain the optional `signature-policy-identifier` signed attribute.

   - The optional signed attribute `commitment-type-indication` if these signatures contain the optional `signature-policy-identifier` signed attribute. The reason for using this attribute in this case is that the explicit signature policy document establishes specific constraints for each commitment made by the signer, which makes imperative that, if a certain commitment is made by the signer, this one is signalled using the aforementioned attribute.

3) Within PAdES-5-XML signatures, the commitments made by the signer is indicated using the `xades:CommitmentTypeIndication` signed property (clause 4.2.5.6 of EN 319 142 [i.4] Part 5).

4) Within PAdES-5-XFA signatures, the commitments made by the signer are signalled in two different ways (clause 5.2.4.6 of EN 319 142 [i.4] Part 5):

   - The optional `description` child of `ds:SignatureProperties` element, if these signatures do not contain the optional `signature-policy-identifier` signed attribute. The description element is defined within the Dublin Core http://purl.org/dc/elements/1.1/ namespace.

   - The optional `xades:CommitmentTypeIndication` signed property if these signatures contain the optional `xades:SignaturePolicyIdentifier` signed property.

8.8 Including indication of signer roles and/or attributes

All the AdES electronic signatures provide mechanisms for indicating the role played by the signer, which entitles her with certain attributes.

This indication may be a mere claim stated by the signer, which the relying party may trust or not as his own discretion, or may be a “certified” statement, i.e., a signed assertion (e.g. attribute certificate, signed SAML assertion) provided by a third party that is trusted by both the signer and the relying parties.

Implementers should assess, for each data object to be signed and for each signature, whether the inclusion of an indication of the signing role of the signer is required or not. Implementers should take into account the legal/regulatory framework of the business process while doing this assessment. For those signatures requiring an indication of the role played by the signer, implementers should assess whether a claimed indication would be enough or a certified indication is required.

Implementers are referred to clause 6.2.6 of EN 319 132 [i.3] part 2, when implementing XAdES signatures. The `xades:SignerRole` signed property may include a set of claimed and/or certified indications of roles. Certified indications of roles may be attribute certificates or SAML assertions signed by third parties that are trusted for issuing such tokens.

Implementers are referred to clause 6.2.6 of EN 319 122 [i.2] part 2, when implementing CAdES signatures. The `signer-attribute` signed attribute may include a set of claimed and/or certified indications of roles. Certified indications of roles may be attribute certificates or SAML assertions signed by third parties that are trusted for issuing such tokens.

When PAdES signatures are used, implementers should take into account the following considerations:

1) It is recommended not to include attribute certificates within PAdES-2 signatures (clause 5.1 of EN 319 142 [i.4] part 2).

2) Within PAdES-3 and PAdES-4 signatures, the signer roles/attributes, if required, are indicated within the `signer-attribute` signed attribute (clause 4.5.10 of EN 319 142 [i.4] part 3).

3) Within PAdES-5 signatures, the signer roles, if required, are indicated within the `xades:SignerRole` signed property. (clauses 4.2.5.4 and 5.2.4.4 of EN 319 142 [i.4] part 5)

8.9 Including additional signed information

Sub-clauses below provide guidance on how to include additional information that is also signed by the signer. Any piece of signed information (including signer commitment and signer role) further qualifies the signed data object(s), the signer or the electronic signature itself.

8.9.1 Including explicit indication of the signature policy

Implementers should include this signed information within a certain signature if such an explicit signature policy has been identified as being the one that has to govern the generation and validation of that signature.

All the AdES electronic signatures provide mechanisms for incorporating explicit information of the signature policy that actually governs their generation and validation.

Within XAdES and CAdES signatures, this information consists in a unique identifier of the signature policy and a digest value computed on the whole or certain part of the unique binary representation of the signature policy document. Optionally this information may include pointers to sites where such a binary representation may be reached.

Implementers are referred to clause 6.2.9 of EN 319 132 [i.3] part 2, when implementing XAdES signatures, which specifies the `xades:SignaturePolicyIdentifier` signed property.

Implementers are referred to clause 6.2.9 of EN 319 122 part 2, when implementing CAdES signatures, which specifies the `signature-policy-identifier` signed attribute.

When PAdES are used, implementers should take into account the following considerations:

1) Within PAdES-3 and PAdES-4 signatures, the signature policy identifier, if required, will appear within the `signature-policy-identifier` signed attribute (clause 4.5.1 of EN 319 142 [i.4] part 3).

2) Within PAdES-5 signatures, the signature policy identifier, if required, will appear within the `xades:SignaturePolicyIdentifier` signed property (clauses 4.2.5.2 and 5.2.4.2 of EN 319 142 [i.4] part 5).

8.9.2 Including indication of the of signed data object format

CAdES, XAdES and PAdES-XML electronic signatures provide mechanisms for incorporating an indication of the format of the signed data object as signed information.

Implementers are referred to clause 6.2.4 of EN 319 132 [i.3] part 2, when implementing XAdES signatures, which specifies the `xades:DataObjectFormat` signed property. This property may contain among other information, the mime type and the encoding of each signed data object.

Implementers are referred to clause 6.2.4 of EN 319 122 [i.2] part 2, when implementing CAdES signatures. This clause specifies two signed attributes, namely: `content-hints`, which is to be used for multi-layered CAdES signatures, and `mime-type`, which may also be used in not multi-layered CAdES signatures. Both attributes allow to indicate the mime type of the signed data object. Should a CAdES signature collectively sign a multipart mime structure, each of these parts may individually indicate its own mime type.

When PAdES are used, implementers should take into account the following considerations:

1) Signed attributes `content-hints` and `mime-type` are not allowed within PAdES-3 and PAdES-4 signatures: what they sign is a PDF container (clause 4.5.7 of EN 319 142 [i.4] part 3).

2) However, `xades:DataObjectFormat` signed property is allowed within PAdES-XML signatures as they may actually sign different types of objects (clauses 4.2.5.5 and 5.2.4.5 of EN 319 142 [i.4] part 5).

8.9.3 Including indication of the of the signature production place

All the AdES electronic signatures provide mechanisms for incorporating an indication of the location where the signature has been purportedly generated as signed information.

Implementers are referred to clause 6.2.5 of EN 319 132 [i.3] part 2, when implementing XAdES signatures, which specifies the optional `xades:SignatureProductionPlace` signed property.

Implementers are referred to clause 6.2.5 of EN 319 122 [i.2] part 2, when implementing CAdES signatures, which specifies the `signer-location` signed attribute.

When PAdES signatures are used, implementers should take into account the following considerations:

1) PAdES-3 and PAdES-4 signatures make use of the optional `Location` entry within the signature dictionary (clause 4.5.9 of EN 319 142 [i.4] part 3).

2) PAdES-5 signatures make use of the optional `xades:SignatureProductionPlace` signed property (clauses 4.2.5.3 and 5.2.4.3 of EN 319 142 [i.4] part 5).

8.10 Supporting signatures lifecycle

The clauses above have provided details on how the signer may embed within the signature signed attributes/properties that further qualify the signature, the signer, or the signed data objects.

It is, however, not unusual that business processes require that additional data are added to the signatures after they have been generated for supporting their lifecycles. Part of these data is validation data, i.e., data that has to be used for validating the signature. Part of this data may also be data for increasing signatures' longevity.

The signer may add part of this information; other may be added by the relying parties or even by third parties specifically entitled for doing that.

Sub-clauses below provide details on the different types of data that may be added to an electronic signature throughout its lifecycle.

8.10.1 Including references to validation data

Certain business processes might advice the signer to incorporate in the signature references of the validation data. These references incorporate means for individually identifying the validation material and also its digest value computed with a certain hash algorithm. This would facilitate these parties the identification and retrieval of such data when validating the signature, without needing to include them within the signature.

XAdES and CAdES specify containers for references to validation data. PAdES signatures do not manage such type of references.

8.10.1.1 Including references to certificates

Both CAdES and XAdES signatures define containers for references to CA certificates and to Attribute Authorities certificates (the later ones are required when the signer signs attribute certificates or signed SAML assertions).

Each reference contains an identifier of the referenced certificate and a digest value computed on it using a specific digest algorithm. Relying parties may use this value for checking that the certificate retrieved is actually the referenced one.

Implementers are referred to clause A1.1 of EN 319 132 [i.3] part 2, when implementing XAdES signatures. This clause specifies the optional `xades:CompleteCertificateReferences` unsigned property, the container for references to CA's certificates required for validating the signature. Implementers are referred to clause A1.3.1 of EN 319 132 [i.3] part 2 when the signature contains attribute certificates or signed SAML assertions. This clause specifies the optional `xades:AttributeCertificateRefs` unsigned property, the container for references to Attribute Authorities' certificates.

Implementers are referred to clause A.1.1.1 of EN 319 122 [i.2] part 2, when implementing CAdES signatures. This clause specifies the optional `complete-certificate-references` unsigned attribute, the container for references to CA's certificates required for validating the signature. Implementers are referred to clause A.1.3 of EN 319 122 [i.2] part 2 when the signature contains attribute certificates or signed SAML assertions. This clause specifies the optional attribute-certificate-references unsigned property, the container for references to Attribute Authorities' certificates.

8.10.1.2 Including references to certificate status data

CAdES and XAdES define containers for references to certificate status data. Both define references to OCSP responses and CRLs. They also define a placeholder for references to other types of certificate status data. Each reference incorporates an identifier of the object and its digest value.

Implementers are referred to clause A.1.2 of EN 319 132 [i.3] part 2, when implementing XAdES signatures. This clause specifies the optional xades:CompleteRevocationReferences unsigned property, the container for references to certificate status data corresponding to CA's certificates required for validating the signature. Also, implementers are referred to clause A1.3.2 of EN 319 132 [i.3] part 2 when the signature contains attribute certificates or signed SAML assertions. This clause specifies the optional xades:AttributeRevocationRefs unsigned property, a container able to contain references to the full set of certificate status data that have been used in the validation of the attribute certificate(s) or signed SAM assertions present in the signature.

Implementers are referred to clause A1.2.1 of EN 319 122 [i.2] part 2, when implementing CAdES signatures. This clause specifies the optional complete-revocation-references unsigned attribute, the container for references to CA's certificates required for validating the signature. Implementers are referred to clause A.1.4 of EN 319 122 [i.2] part 2 when the signature contains attribute certificates or signed SAML assertions. This clause specifies the optional attribute-revocation-references unsigned property, the container for references to certificate status corresponding to Attribute Authorities' certificates and the attribute certificates.

8.10.2 Time-stamping references to validation data

Certain business processes may require relying parties to prove the time when they firstly validated a certain signature and, simultaneously, due to the fact that a good part of the validation data required by a relevant number of signatures is the same, also may require not including this validation material within the signatures.

Under these circumstances, implementers may opt for including references to validation data and time-stamp tokens on them. Using this combination a relying party may prove that at the time instant present within the time-stamp token it had gained access to the referenced material.

XAdES and CAdES define two types of containers for time-stamp tokens on references to validation data.

Implementers are referred to clause A.1.4 of EN 319 132 [i.3] part 2, when implementing XAdES signatures. This clause specifies two unsigned properties. The first one is xades:SigAndRefsTimeStamp, a container for a time-stamp token computed on the ds:SignatureValue, any present xades:SignatureTimeStamp, and any container of references to validation data. The second one is xades:RefsOnlyTimeStamp, a container for a time-stamp token computed on any container of references to validation data only.

Implementers are referred to clause A.1.5 of EN 319 122 [i.2] part 2, when implementing CAdES signatures. This clause specifies two unsigned properties. The first one is time-stamped-certs-crls-references, a container for a time-stamp token computed on any container of references to validation data only. The second one is CAdES-C-time-stamp, a container for a time-stamp token computed on the OCTETSTRING of the SignatureValue field within SignerInfo, any present signature-time-stamp, and any container of references to validation data.

Although there is no mandatory constraint on the scenarios where to use one or the other, a good practice is to use the xades:SigAndRefsTimeStamp or CAdES-C-time-stamp when references to OCSP responses are used, while xades:RefsOnlyTimeStamp or time-stamped-certs-crls-references are better for references to CRLs.

8.10.3 Ensuring longevity and resilience to change of the signatures

Certain business processes require large longevity and high change resilience to signatures. Under these circumstances, implementers may opt by building archival forms of electronic signatures.

At a minimum, archival forms are signatures including a time-stamp token on the signature, all the validation data required for its validation and one or more archive time-stamp tokens (that time-stamp anything in the signature present at the time of generating the archive time-stamp tokens). However, more complete forms may also incorporate references on the validation data and time-stamp tokens on them.

Archival forms require at least two specific components:

1) Containers for validation data values.

2) Containers for archival time-stamp tokens.

Additionally, certain formats require containers for ancillary information.

All the AdES signatures may build up archival forms of signatures. Sub-clauses below provides guidance on the mechanisms used within each format.

8.10.3.1 CAdES signatures

CAdES signatures have evolved with time since its first version was published as ETSI Technical Specification. This has resulted in changes in the containers of validation data, the containers of the archive-time-stamp tokens, and the containers of ancillary information.

8.10.3.1.1 Containers for validation data

CAdES signatures compliant with EN 319 122 [i.2] part 2 embed the certificates and certificate status values required for validating the signature and any present attribute certificate or signed SAML assertion, within `SignedData.certificates` and `SignedData.crls` fields.

Business processes might require implementations to be able to validate legacy CAdES signatures that use different containers (currently superseded by EN 319 122 [i.2] part 2). In such cases, implementers should take into account that these signatures could contain the following containers:

1) Unsigned attributes `certificate-values` and `revocation-values` (clauses A.1.1.2 and A.1.2.2 of EN 319 122 [i.2] part 2 respectively). These were containers for validation data required for validating the signature and any present attribute certificate or signed SAML assertion or any time-stamp token not containing all needed information before the first archive time-stamp token (or `long-term-validation` attribute) was added to the signature.

2) Fields `extraCertificates` and `extraRevocation` embedded within the `long-term-validation` unsigned attribute. These were containers for extra validation data after the first `long-term-validation` attribute was added (see clause A.2.3 of EN 319 122 [i.2] part 2).

8.10.3.1.2 Containers for archival time-stamp tokens

EN 319 122 [i.2] part 2 that new CAdES signatures embed the `archive-time-stamp-v3` unsigned attribute as container for the archive time-stamp token (see clause 6.5.2).

As before business processes might require implementations to be able to validate legacy CAdES signatures that use different containers (currently superseded by EN 319 122 [i.2] part 2). In such cases, implementers should take into account that these signatures could contain the following time-stamp tokens containers:

1) `timeStamp` field within the `long-term-validation` unsigned attribute.

2) Archive time-stamp unsigned attribute whose OID is: { `iso(1) member-body(2) us(840) rsadsi(113549) pkcs(1) pkcs-9(9) smime(16) id-aa(2)  48.`

3) Archive time-stamp unsigned attribute whose OID is: `object identifier { iso(1) member-body(2) us(840) rsadsi(113549) pkcs(1) pkcs-9(9) smime(16) id-aa(2) 27.`

8.10.3.1.3 Containers for ancillary information

EN 319 122 [i.2] part 2 requires embedding the `ats-hash-index` unsigned attribute within `archive-time-stamp-v3`'s signature. That attribute contains sequences (`SEQUENCE OF` ASN.1 structures) of digest values of all the certificates, certificate status data and unsigned attributes within the electronic signature that the archive time-stamp has to cover.

It serves two purposes: first it unambiguously identifies what parts of the validation material and unsigned attributes present in the signature are actually covered by the time-stamp token; secondly, it solves the problem associated to the fact that the unsigned attributes and the `SignedData.certificates` and `SignedData.crls` fields are contained within `SET OF` ASN.1 structures. These structures do not define an inner order among their components, which has historically caused problems to interoperability. The soluition is achieved by concatenating the contents of the aforementioned `ats-hash-index` to the archive time-stamp's message imprint computation input, instead of individually concatenating the different pieces of validation data and unsigned attributes.

Readers are referred to clause 6.5.1 of EN 319 122 [i.2] part 2 for further details.

8.10.3.2 XAdES signatures

XAdES signatures have also evolved with time since its first version was published as ETSI Technical Specification. This has resulted in changes in the containers of validation data, the containers of the archive-time-stamp tokens, and the containers of ancillary information.

8.10.3.2.1 Containers for validation data

EN 319 132 [i.3] part 2 identifies the following containers for certificates and certificate status data:

1) `ds:KeyInfo` element, and unsigned properties `xades:CertificateValues`, `xades:RevocationValues`, `AttrAuthoritiesCertValues`, and `AttributeRevocationValues`. These are containers for validation data required for validating the signature and any present attribute certificate or signed SAML (see clause 6.4 of EN 319 132 [i.3] part 2).

2) Fields `xadesv141:TimeStampValidationData`. This is a container for validation data corresponding to one or more time-stamp tokens present within the signature (see clause 6.6 of EN 319 132 [i.3] part 2).

8.10.3.2.2 Containers for archival time-stamp tokens

EN 319 132 [i.3] part 2 requires that new XAdES signatures embed the `xadesv141:ArchiveTimeStamp` unsigned attribute as container for the archive time-stamp token.

Business processes might require implementations to be able to validate legacy XAdES signatures that use different containers (currently superseded by EN 319 132 [i.3] part 2). In such cases, implementers should take into account that these signatures could contain the following time-stamp tokens containers:

1) `xades:ArchiveTimeStamp` unsigned property (see clause A.2.1 in EN 319 132 [i.3] part 2).

8.10.3.3 PAdES signatures

9.10.3.3.1 Containers for validation data

EN 329 142 part 4 specifies two PDF dictionaries as containers for validation data:

1) Document Security Store (DSS) dictionary. This dictionary is designed as a single container for validation data of some or all signatures in the document (see clause A.1 of EN 329 142 part 4).

2) Validation Related Information (VRI) dictionary. This dictionary acts as a container for validation data related to one specific signature in the document (see clause A.1 of EN 329 142 part 4).

8.10.3.3.2 Containers for archival time-stamp tokens

EN 329 142 part 4 clause A.2 specifies the Document Time-stamp dictionary as a special type of signature dictionary, which contains a time-stamp token time-stamping the PDF document's byte range indicated in its ByteRange entry. As such, it may actually time-stamp the totality of the contents of the PDF document, including any present signature.

8.11 Managing detached signatures and signed data objects: ASiC containers

EN 319 162 [i.6] specifies containers that hold one or more detached signatures (XAdES or CAdES) and the data objects signed by these signatures. These containers allow to manage detached signatures and their signed data objects in an standardized way.

Whenever the analysis done in previous phase shows that the business e-processes require to generate and manage detached signatures, and advices that, in order to facilitate such a management, it is worth to embed both the signatures and their signed objects within a container, implementers are referred to implement EN 319 162 [i.6].

ASiC containers standardize mechanisms for referencing data objects signed by detached CAdES signatures.

If there is only one document that may be signed by several detached signatures, implementers should use the ASiC Simple (ASiC-S) form. Implementers are referred to clause 5 of EN 319 162 [i.6] part 2.

If, on the contrary, there are more than one data objects signed by detached signatures, then implementers should consider using the ASiC Extended (ASiCE) form. Implementers are referred to clause 6 of EN 319 162 [i.6] part 2.

If the embedded signatures are CAdES signatures, the ASiC container incorporates one additional XML file (known an ASiCManifest file) per each CAdES signature embedded within the container. Each ASiCManifest file references (using URIs) all the documents signed by the corresponding CAdES signature.

If the embedded signatures are XAdES signatures, ASiC relies on the native mechanisms of XML Signatures (i.e. the usage of `ds:Reference` elements) for referencing all the documents signed by them.

8.12 Selecting proper Signature Creation Devices

It is out of the scope of the present document to provide guidance on devices for electronic signature creation.

Instead, implementers are strongly recommended to read ETSI TR 119 200: "Business Driven Guidance for Signature Creation and Other Related Devices" [i.21]. This is another guidance document of the guidance documents series, which specifically addresses area 2 ("Signature Creation and Other Related Devices") of the Rationalised Framework [i.1].

Implementers will find in that document material that will guide them in the usage of the different types of documents within that area (Policy & Security Requirements, Technical Specifications, and Conformity Assessment) for selecting the signature creation device most suitable for the targeted business processes.

8.13 Selecting proper cryptographic suites

It is out of the scope of the present document to provide guidance on cryptographic suites.

Instead, implementers are strongly recommended to read ETSI TR 119 300: "Business Driven Guidance for Cryptographic Suites" [i.22]. This is another guidance document of the guidance documents series, which specifically addresses area 3 ("Cryptographic Suites") of the Rationalised Framework [i.1].

At the time of writing the present document, this area contains only two documents, namely: the aforementioned ETSI TR 119 300, and ETSI TS 119 312: "Cryptographic Suites for Secure Electronic Signatures" [i.23].

ETSI TS 119 312 [i.23] defines a number of different cryptographic suites for secure electronic signatures. Implementers will find in ETSI TR 119 300 material that will guide in the selection of cryptographic suites for the requirements identified within the targeted business processes.

8.14 Signature generation, upgrade and validation applications

When dealing with the technicalities of implementing (or selecting) applications for generating, upgrading and/or validating advanced electronic signatures, implementers should carefully read the following documents present within area 1 of the Rationalised Framework [i.1]:

1) EN 419 111 [i.9]: “Protection Profiles for Signature Creation & Validation Applications” [i.9].

2) EN 319 102 [i.7]: “Procedures for Signature Creation and Validation” [i.7].

Sub-clauses below provide details on both documents.

8.14.1 Selecting the suitable Protection Profile

EN 419 111 [i.9] is a multi-part document, which in its introduction defines the security requirements for Signature Creation and Signature Validation Applications. Implementers will find there the details of the terminology used in the rest of the document, as well as the functions and environment of the SCA/SVA.

Implementers of a Signature Creation Application should carefully read EN 419 111 [i.9] part 2: “Core Protection Profile for a Signature Creation Application” and part 3, which defines extensions to core Protection Profiles for a variety of situations. Part 2, as its name indicates, defines the core protection profile for a SCA, whose Target of Evaluation is software running on an operating system and a Signature Creation Platform hardware.

Implementers of a Signature Validation Application should carefully read EN 419 111 [i.9] part 4: “Core Protection Profile for a Signature Validation Application” and part 5, which defines extensions to core Protection Profiles for a variety of situations. Part 4, as its name indicates, defines the core protection profile for a SVA, whose Target of Evaluation is software running on an operating system and a Signature Validation Platform hardware

Implementers, after reading these documents should select the Protection Profile(s) that their tools should be compliant with for properly fulfilling the requirements imposed by the targeted business processes.

8.14.2 Implementing the signature generation and upgrade processes

With regards to the process of generating and upgrading an electronic signature, ETSI EN 319 102 [i.7]: “Procedures for signature creation and validation” [i.7] specifies procedures for creating and upgrading (Advanced) electronic signatures in a format-agnostic way. It introduces general principles, objects and functions relevant when creating and upgrading signatures. It also defines general forms of advanced electronic signatures that increase their longevity. It is based on the use of public key cryptography to produce such signatures, which are supported by public key certificates.

Implementers will find within this document a functional model for a SCA that include the signature creation functions, the information objects, and those interfaces that are relevant to its security. Implementers should ensure that their implementations actually provide the functionality specified as mandatory within this document. However, the distribution of such functionality may be done among a set of components that is different from the set identified within ETSI EN 319 102 [i.7]. Below follows a summary of this functionality:

1) Functions that support the different types of interactions between the signer and the SCA. Implementers should implement them in a way that allows building environments able to fulfil the requirements related with the formalities of signing.

    - Signer Interaction Component. Function that controls the signature creation process and that is used for all the interactions between the signer and the SCA, except of the interaction for authentication.

    - Signer Document Composer. Function that is used for creation, input or selection of the data object(s) to be signed. Text editors are an example.

    - Signature Attributes Viewer. Function that allows the signer to view and select the attributes (properties) that will be signed together with the data object(s).

    - Signer’s Document Presentation Component. Function that presents the data object(s) to the signer, and also allows the signer to select them.

    - Data To Be Signed Formatter. Function that allows formatting of the data objects to be signed.

  - Signer's Authentication Component. Function that allows the signer to input authentication data to the SCA. This function should be implemented in a way that fully satisfies the requirements (in terms of inputs required) imposed by the authentication mean(s) selected in the previous tasks of the process.

  - Signed Data Object Composer. Function that associates the computed digital signature with the signed data object(s), suitably formatted and outputs the result of signing in some standard format.

2) Data Hashing Component. This function is the responsible for producing the DTBS Representation (which might be non- hashed, partially hashed or completely hashed as required by the SCDev). As the business model may require different combination/sequencing of data object(s) to be signed and signed properties (attributes) the implementers should ensure that this function is designed in a way that allows to properly treat all these cases.

3) Functions that support the work that SCA and SCDev have to perform in close co-operation. Implementers should take all these issues into account in the view of the different requirements imposed by the business context.

   - SCDev/SCA Communicator. This function manages all the interactions between the SCA and the SCDev that are required for the generation of the signature, including the establishment of the physical communication, the retrieval of the SCDev Token information, the retrieval of certificates, the selection of the signature creation data, the actual performance of signer authentication, and the selection of the SCDev functionality in the case that the SCDev functions are part of a larger application that has more functions than just the signature creation function.

   - SCDev/SCA Authenticator. A conditional function in charge of establishing a trusted path between SCDev and SCA, for those situations where this trusted path can not be established by organizational means.

   - Work sharing between SCA and SCDev. This function controls the way in which the SCA and the SCDev share the work of computing the sequence of octets that are eventually digitally signed. As mentioned with the Data Hashing Component, implementers should ensure that this co-operation is implemented in a way that ensures a proper treatment of all the different combinations/sequencing of data object(s) and signed attributes (properties) identified in the business model.

4) Signature Logging Component. Function that records details of the signatures created. Implementers should take into consideration any specific logging requirement within the business context when implementing such function.

After the SCA functional model, ETSI EN 319 102 [i.7] provides details of data flow envisioned for the process of the generation of an Electronic Signature on the data object(s) to be signed and a set of signed attributes (properties), highlighting its relationship to the SCA functions aforementioned.

Finally, the part of ETSI EN 319 102 [i.7] devoted to the SCA provides details of the lifecycle of an electronic signature, addressing the initial creation of the signature, the post-signature creation validation, and the different forms to which an electronic signature may be upgraded by incorporation of unsigned attributes (properties) enveloping validation data (certificate references and/or values, certificate status data references and/or values) and/or time-stamp tokens proving the existence of certain components of the signatures, until their most complete form: the archival form that increase their longevity. Implementers should specify the lifecycle of each of the electronic signatures that have to be generated and managed within the targeted electronic business.

8.14.3 Implementing the signature validation process

With regards to the process of validating an electronic signature, ETSI EN 319 102 [i.7]: "Procedures for signature creation and validation" [i.7] specifies procedures for establishing whether an (Advanced) electronic signature is technically valid and is the capital reference for implementing a Signature Validation Application (SVA).

More specifically it defines an algorithm to validate electronic signatures, with special consideration on signature validation of electronic signatures where certificates may have expired or been revoked or even the usage period of algorithms have been exceeded. The algorithm takes advantage of security measures that have been applied by the different entities that act on the signatures during their lifecycle (e.g. signer or previous verifiers that may have upgraded the initial signatures) and ensures that such signatures still can be validated. Although the process is presented as an algorithm, implementers are not supposed nor recommended to implement it as described. However, any implementation claiming conformance has to provide the same results as the algorithm would provide.

ETSI EN 319 102 [i.7] contextualizes the operation of a SVA as follows:

1) The SVA is called by the so-called Driving Application (DA), to which it has to return the results of the validation process, in the form of a validation report. ETSI EN 319 102 [i.7] specify a minimum set of pieces of information to be included within this report, including the status indication, which may be VALID, INVALID and INDETERMINATE (meaning that at the moment the validation was performed, the available information was insufficient to ascertain the signature to be VALID or INVALID, and that consequently, an ulterior validation could, under certain circumstances, return a different status indication).

2) The algorithm takes as inputs the electronic signature to be validated and a set of constraints coming from different sources whose fulfilment the SCA ascertains during the validation process. A constraint, according to that document, is any abstract formulation of rules, ranges and computation results whose fulfilment is assessed during the validation of the signature. These validation constraints may be defined in different ways:

   - Using formal policy specifications. An example of such situations is signature policy files containing the signature policy validation expressed in ASN.1 or XML syntaxes as specified in ETSI EN 319 172 [i.10]: “Signature Policies” [i.10].

   - Defined explicitly in system specific control data: e.g. in conventional configuration-files like property or in-files or stored in a registry or database.

   - Implicitly by the implementation itself.

   Additionally, the DA may provide constraints to the SVA via parameters implied by the application or the user. ETSI EN 319 102 [i.7] identifies input constraints on: X.509 certificate path validation, certification chain, on certificates revocation, on time-stamp trust, on X.509 certificates meta-data, and on cryptographic issues.

3) Finally, ETSI EN 319 102 [i.7] proposes the contents of the validation report (although without proposing any specific format). This report contains:

   - a result code, indicating the major result of the validation procedure (VALID, INVALID, INDETERMINATE),

   - a result sub-code, indicating the reasons for the major result, and

   - a set of associated validation report data, specific for each sub-code.

The algorithm specified by ETSI EN 319 102 [i.7]:

1) Identifies basic building blocks in charge of:

   - Identifying the signer’s certificate.

   - Initializing the validation context, i.e. initializing the validation constraints and parameters to be used during the validation process.

   - Validating X.509 certificate. The process defined for this block builds on the Certification Path Validation, as specified in IETF RFC 5280: "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile" [i.26].

   - Cryptographically verifying digital signature.

   - Validating the acceptance of the signature, i.e. performing any additional required validation on the attributes (properties) of the signature.

   As stated before, the validation process is presented as an algorithm that suitably makes use of the aforementioned building blocks.

2) Defines an algorithm for performing the so-called Basic Validation, i.e. the process required for performing a short-term signature validation, adequate for basic signatures (like the ones within CRLs, OCSP responses, etc.) as well as AdES-BES and AdES-EPES forms.

3) Defines an algorithm for performing the Validation of time-stamp tokens, which builds on the aforementioned Basic Validation algorithm by adding an additional step of data extraction, consisting in returning relevant data items from the time-stamp token itself (like the generation time, the message imprint, etc), which may be used in the process of validating more evolved forms of AdES signatures, where these time-stamp tokens are present.

4) Defines an algorithm for performing the validation of signatures with trusted time indication, i.e. AdES-T forms, which builds on the Basic Validation and the Validation of time-stamp tokens.

5) Defines an algorithm for performing the Validation of LTV forms, adequate for validating (X/C)AdES-C, (X/C)AdES-X, (X/C)AdES-XL, (X/C)AdES-A and PAdES-LTV. The algorithm is built on the concept of Proof Of Existence (POE) and a set of additional building blocks, listed below:

   - Proof Of Existence (POE) of an object, is an evidence that proves that this object (a certificate, a CRL, signature value, hash value, etc.) existed at a specific date/time, which may be a date/time in the past. Of special interest for this algorithm are the POEs of objects at a time in the past. There are several ways of generating such a type of POEs: time-stamping an object in certain time provides a POE of that object time afterwards; but also electronic notaries, archival services or other services may provide this type of POEs.

   - Past Certificate Validation process. This is a process that validates a certificate at a date/time that may be in the past. This may be needed in the verification of a long-lived signature, which may include expired certificates for instance.

   - POE extraction, a process that derives POEs from a given time-stamp token within the electronic signature.

   X.509 Certificate path validation constraints,Additional Chain Constraints, Additional Revocation Constraints, Additional Time-Stamp Trust Constraints, Constraints on X.509 Certificate meta-data, and Cryptographic Constraints

9 Signature creation and validation catalysing toolkit

Implementers should also be aware of the existence of a holistic toolkit that they may use for assessing the conformance of their implementations to referenced standards. This toolkit aims to further supporting and accelerating of the deployment of interoperable electronic signatures across Europe.

Sub-clauses below provide an overview of the elements that integrate the package.

9.1 Technical Specifications

The first element of the aforementioned toolkit is a set of ETSI Technical Specifications for testing conformance and interoperability of applications with regards to the implementation of standardised signature formats and of signature policies as listed below:

1) ETSI TS 119 104: “General requirements on Testing Conformance and Interoperability of Signature Creation and Validation” [i.12].

2) ETSI TS 119 124: “CAdES Testing Conformance and Interoperability” [i.13].

3) ETSI TS 119 134: “XAdES Testing Conformance and Interoperability” [i.14].

4) ETSI TS 119 144: “PAdES Testing Conformance and Interoperability” [i.15].

5) ETSI TS 119 154: “Testing Conformance and Interoperability of AdES in Mobile Environments” [i.16].

6) ETSI TS 119 164: “ASiC Testing Conformance and Interoperability” [i.17].

7) ETSI TS 119 174: “Testing Conformance and Interoperability of Signature Policies” [i.18].

ETSI TSs 119 124, 119 134, 119 144 and 119 164 address each of the AdES signature formats and the ASiC package. All of them have 4 parts. In all of them, implementers will find the following contents:

1) Parts 1 and 2 specify carefully defined test suites for testing interoperability. They include test cases aiming at ascertaining that different implementations generating and validating AdES signatures and ASiC containers are able to interoperate, i.e., that the signatures/containers validated by one implementation are properly validated by the others. The test suites defined within these documents address those aspects that have relevance for achieving interoperability. They also include different types of test cases:

   - Positive cross-verification test cases. These test cases require to an implementation to generate a valid AdES signature or ASiC container according to a detailed specification of its contents. Other implementations aiming at testing interoperability with the first one should try to validate this signature/container. A VALID result would mean that implementations successfully interoperate with regarding to the aspects tested.

   - Positive cross-verification, upgrade and arbitration test cases. These test cases require the participation of at least 3 different implementations and would work as follows: implementation A generates a valid AdES signature or ASiC container according to a detailed specification of its contents. Implementation B, acting as relying party, would validate this signature and upgrade it to a more evolved form, also according to the specifications of the test case. Finally, a third implementation C, acting as a purported arbitrator, would validate the upgraded signature. These test cases serve for testing how implementations behave in situations where signatures are upgraded and these upgraded signatures are in turn validated by entities that are neither the signer, nor the one that firstly validated the signature and after upgraded it.

   - Negative test cases. These test cases specify signatures for which the validation process cannot end with the VALID result, according to EN 319 102 [i.7]. They aim at ascertaining that implementations actually correctly deal with signatures or containers that cannot be considered as technically valid due to a number of reasons, and in consequence, do not generate false positive results.

   These test suites are built taking into account not only the specifications on the formats, but also on the signature validation process specified within EN 319 102 [i.7]. This, among other things, require the presence of different PKIs of different degree of complexity, ranging from a very simple one (where all the certificates, certificate status data, and time-stamps appertain to the same hierarchy of CAs), to complex combinations of PKIs that try to be close to real situations.

   For all the formats, parts 1 specify test suites for testing interoperability on the core specifications, while parts 2 of the document specify test suites for the corresponding baseline profiles.

2) Parts 3 and 4 define complete sets of test assertions that aim at ascertaining each and every of the requirements specified by the core specification and the baseline profile respectively. In consequence, if an AdES electronic signature or an ASiC container passes all the assertions specified within Part 3 it may be claimed that it is compliant with the corresponding core specification, and similarly, if it passes all the assertions specified within Part 4, it may be claimed that it is compliant with the corresponding baseline profile.

ETSI EN 119 174, in turn, also specifies test suites for testing interoperability and sets of tests assertions for testing conformance with ETSI EN 319 172 [i.10].

ETSI EN 319 103 [i.11] specifies general requirements for testing interoperability and conformance.

9.2 Conformance testing software tools

The second element of the catalysing toolkit is a set of software tools, freely available, that test conformance of AdES signatures, and ASiC containers against their corresponding core and baseline profiles specifications.

Each software tools actually does perform the whole set of test assertions specified in the corresponding part of ETSI ENs 319 124, 319 134, 319 144, and 319 164. The output of the tools do not only provide details on each assertion tested and its corresponding result, but also on the different components of the signature/container, focussing specially in certificates and time-stamp tokens. Additionally, they provide useful trace information on computations that experience has proved to be source of interoperability problems: they provide, for instance, the trace of the contributions that have to be made for building the input to the computation of the message imprints for the different time-stamp tokens types that appear within a signature. This has proved to be of great usefulness for implementers, as helps them to identify within their applications the sources of specific problems when dealing with such computations, and facilitates a unified reading and understanding of the corresponding specification.

These tools are freely available through the ETSI CTI Portal on Electronic Signatures (http://xades-portal.etsi.org).

9.3 Interoperability test events

The third element of the catalysing toolkit is the ETSI CTI Portal for Electronic Signatures. This is a portal that provides full support to the conduction of remote interoperability test events on signature creation and validation. Using the facilities provided by this portal, the participants in the event do not need to travel to a certain place and meet face to face for a certain number of days, devoting all the working hours to actually perform interoperability tests. Instead, they can organize their time in their own premises, working asynchronously, and meeting remotely at specific dates and times while the event is alive (the experience proves that a duration of 3 weeks is suitable for this kind of events). The portal contains all the information that the participants require for conducting their tests, namely:

1) The interoperability test suites. Participants find at the portal a complete and detailed specification of each test case.

2) Repository of signatures generated by each participant, suitably structured.

3) Repository of validation reports coming from each participant, suitably structured.

4) Global interoperability matrix, automatically updated each time that a participant uploads a new validation report at the portal.

5) Per participant interoperability matrixes, which reports to each participant the results obtained by the others after they have tried to validate each of her signatures.

6) Documentation explaining how to conduct while participating in the events, i.e., the steps to be performed by each participant, and how they have to interact with the portal for uploading signatures/containers/reports and downloading other participants' signatures/containers.

7) The conformance testing tools described above, allowing them to not only test interoperability with other implementations but also test conformance of their own tools against the corresponding specification.

The experience proves that implementers find at this kind of events a place where:

1) To ascertain the conformance of their own tools against the reference specification.

2) To ascertain the degree of interoperability of their tools with other tools in the market.

3) To identify conformance and/or interoperability problems within their own tools.

4) To discuss with other relevant players in the field about specific issues within the standards. This includes:

    - Identify bugs within the standards, discuss potential solutions and recommend one of them to the standardization body in charge of the specification.

    - Identify ambiguities within the standard that lead to different interpretations (and in consequence, to lack of interoperability), build consensus on a unique interpretation, and raise recommendations for fixing them to the standardization body in charge of the specification.

    - Discuss with other participants about what would be suitable in a potential evolution of the standard (e.g. addition of new functionality), and raise the corresponding request to the standardization body in charge of the specification.

10 Evaluation processes

While implementing a signature creation, upgrade and/or validation application, implementers should be aware that very likely the market is going to request that they pass an evaluation process that ensures that the application:

1) Generates signatures compliant with the selected formats, forms and levels.

2) Complies with the requirements defined within EN 319 102 [i.7] with regards to the procedures for generating, upgrading, and/or validating electronic signatures.

3) Is compliant with the selected Protection Profiles

4) The application itself and the environment where it is used are compliant against the Policy Requirements specified within EN 319 101 [i.8].

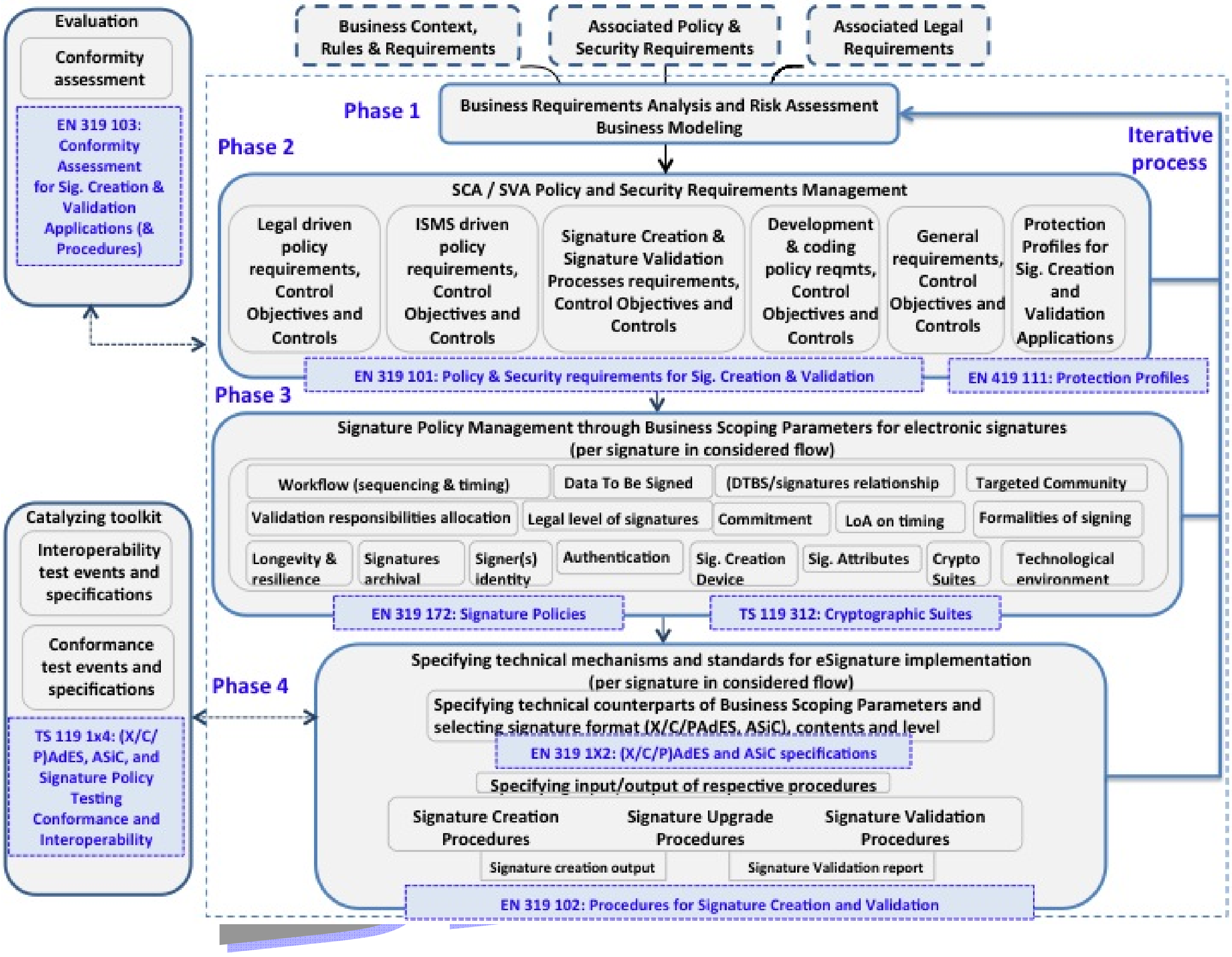
Implementers are suggested to read EN 319 103: “Conformity Assessment for Signature Creation and Validation Applications (& Procedures)” [i.11] for a deep understanding of the evaluation processes their applications may need to face.

11 Corollary: the process within the context of the Standardisation Framework.

As a corollary of this guide, this clause summarizes the existing relationships between each of the phases within the proposed process for implementing electronic signatures in electronic business and the existing documents within the area 1 (Signature Creation and Validation) of Standardisation Framework.

Figure 2 below graphically shows these relationships.

Figure 2: Relationship between process’ tasks and documents within the area 1 of the Rationalised Framework.


Evaluation
Conformity assessment
EN 319 103: Conformity Assessment for Sig. Creation & Validation Applications (& Procedures)
Business Context, Rules & Requirements
Associated Policy & Security Requirements
Associated Legal Requirements
Phase 1
Business Requirements Analysis and Risk Assessment Business Modeling
Iterative process
Phase 2
SCA / SVA Policy and Security Requirements Management
Legal driven policy requirements, Control Objectives and Controls
ISMS driven policy requirements, Control Objectives and Controls
Signature Creation & Signature Validation Processes requirements, Control Objectives and Controls
Development & coding policy reqmts, Control Objectives and Controls
General requirements, Control Objectives and Controls
Protection Profiles for Sig. Creation and Validation Applications
EN 319 101: Policy & Security requirements for Sig. Creation & Validation
EN 419 111: Protection Profiles
Phase 3
Signature Policy Management through Business Scoping Parameters for electronic signatures (per signature in considered flow)
Workflow (sequencing & timing)
Data To Be Signed
(DTBS/signatures relationship
Targeted Community
Validation responsibilities allocation
Legal level of signatures
Commitment
LoA on timing
Formalities of signing
Longevity & resilience
Signatures archival
Signer(s) identity
Authentication
Sig. Creation Device
Sig. Attributes
Crypto Suites
Technological environment
EN 319 172: Signature Policies
TS 119 312: Cryptographic Suites
Catalyzing toolkit
Interoperability test events and specifications
Conformance test events and specifications
TS 119 1x4: (X/C/P)AdES, ASiC, and Signature Policy Testing Conformance and Interoperability
Phase 4
Specifying technical mechanisms and standards for eSignature implementation (per signature in considered flow)
Specifying technical counterparts of Business Scoping Parameters and selecting signature format (X/C/PAdES, ASiC), contents and level
EN 319 1X2: (X/C/P)AdES and ASiC specifications
Specifying input/output of respective procedures
Signature Creation Procedures
Signature Upgrade Procedures
Signature Validation Procedures
Signature creation output
Signature Validation report
EN 319 102: Procedures for Signature Creation and Validation

History

Document history		
<Version>	<Date>	<Milestone>
0.0.0a	14/05/13	Initial version including the TOCs and identification of the main sources for several parts of the document.
0.0.0f	17/8/2013	First almost complete version of the document.
0.0.1	9/9/2013	Version distributed to ETSI ESI TC
0.0.2	30/9/2013	Stable draft for public comments