Requirements Report

D33.1

Document Identification	
Date	2013-12-16
Status	Final
Version	1.1

Related SP/WP	SP3/WP33	**Document Reference**	D33.1
Related Deliverable(s)	D21.4, D21.5, D22.x, D32.3	**Dissemination Level**	CO
Lead Participant	TUG	**Lead Author**	Peter Lipp (TUG) Christof Rath (TUG)
Contributors	Peter Lipp (TUG) Christof Rath (TUG) Moritz Horsch (TUD) Kristi Uukkivi (SK) Jaan Murumets (SK) Christopher Ruff (USTUTT) Jens Kubieziel (AG) André Gutwirth (AG)	**Reviewers**	Omar Almousa (DTU) Detlef Houdeau (IFAG)

Abstract: This document collects the requirements for the electronic signature service of the *FutureID* client.

This document is issued within the frame and for the purpose of the *FutureID* project. This project has received funding from the European Unions Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 318424.

1 Executive Summary

For the *FutureID* client we want to provide the functionality to sign digital documents and to verify such digital signatures. For this purpose we will develop a framework that is capable to integrate current and, in the best case, also future electronic signature standards. The framework will be interfaced via OASIS-DSS. The different signature formats will be implemented as plug-ins to get a modular design that shall be flexible enough for future developments.

Within the project we will focus on the various formats of advanced electronic signatures as specified by ETSI. These are XAdES (ETSI TS 101 903), CAdES (ETSI TS 101 733) and PAdES (ETSI TS 103 172) in their different levels BER, LT and LTA. This document will collect the requirements regarding the framework as such, the different signature formats and protocols we want to support, and the requirements that are derived from other *FutureID* work packages.

Some potentially relevant standards, like CEN CWA 15171, are currently in a phase of being updated during the European Commissions' mandate on electronic signatures (M/460) by CEN or ETSI, and new, potentially relevant standards are in the process of development. Following such standards may be considered during the project. The use of other standards is currently not foreseen.

2 Document Information

2.1 Contributors

Name	Affiliation
Peter Lipp	TUG
Christof Rath	TUG
Moritz Horsch	TUD
Kristi Uukkivi	SK
Jaan Murumets	SK
Christopher Ruff	USTUTT
Jens Kubieziel	AG
André Gutwirth	AG

2.2 History

0.1	2013-04-24	Peter Lipp (TUG)	Initial version
0.2	2013-05-16	Moritz Horsch (TUD)	TUD contribution
0.3	2013-05-24	Kristi Uukivi (SK)	SK contribution
0.4	2013-06-11	Christopher Ruff (USTUTT)	USTUTT contribution
0.5	2013-07-08	Jens Kubieziel (AG)	AG contribution
0.6	2013-07-11	Christof Rath (TUG)	Unify formatting and bibliography
0.7	2013-07-16	Peter Lipp (TUG)	Improving requirements (he thought)
0.8a	2013-09-30	Christof Rath (TUG)	Incorporate review comments from DTU
0.8b	2013-10-22	Christof Rath (TUG)	Incorporate review comments from IFAG
0.9	2013-10-23	André Gutwirth (AG)	AG contribution (revised)
1.0	2013-10-31	Christof Rath (TUG)	Finalization
1.1	2013-12-16	Christof Rath (TUG)	Changes according to the new Reference Architecture terminology

2.3 Table of Contents

2.4 List of Acronyms

ASiC	Associated Signature Container ETSI TS 102 918
ASN.1	Abstract Syntax Notation One
BER	Basic Encoding Rules
BS	Broker Service
CA	Certification Authority
CAdES	CMS Advanced Electronic Signatures
CEN	European Committee for Standardization
CMS	Cryptographic Message Syntax
CRL	Certificate Revocation List
DA	Driving Application
DSS	Digital Signature Services
DOW	Description of Work FutureID-DOW
ECDSA	Elliptic Curve Digital Signature Algorithm
eID	Electronic Identity
ETSI	European Telecommunications Standards Institute
IdM	Identity Management
IETF	Internet Engineering Task Force
LT	Long Term conformance level
LTA	Long Term with Archive time-stamps conformance level
OASIS	Organization for the Advancement of Structured Information Standards
OCSP	Online Certificate Status Protocol
PAdES	PDF Advanced Electronic Signatures
PDF	Portable Document Format
PGP	Pretty Good Privacy
PIN	Personal Identification Number
PKI	Public-Key Infrastructure
RFC	Requests for Comments
RSA	Algorithm for public-key cryptography named after Rivest, Shamir and Adleman, who first publicly described the algorithm.
SIM	Subscriber Identity Module
SP	Sub Project
SPKI	Simple PKI
SVA	Signature Verification Application
TS	Trust Service
WP	Work Package
WPKI	Wireless PKI
XAdES	XML Advanced Electronic Signatures
XML	eXtensible Markup Language
XMLDSig	eXtensible Markup Language Digital SIGnature

2.5 Glossary of Terms

application
An Application is a service consumed by the user and integrated into the FutureID infrastructure via the Application Integration Services. The AS may be a government service for citizens (ie. health service, public administration service) as well as business services which can be used by both large and small enterprises.

attribute
An attribute is a physical or abstract property belonging to an entity. An attribute typically consists of a name value pair.

attribute value
An attribute value is a particular instance of the class of information indicated by an attribute type component which indicates the class of information given by that attribute.

authentication
Authentication is the process of corroborating a claimed set of attributes or facts with a specified, or understood, level of confidence.

authentication protocol
An authentication protocol is a protocol used to authenticate data or entities. When relating to entities, it usually consists of the presenting of an identifier (e.g. a 'user-ID') and verifying the proof of something (e.g., the knowledge of a secret).

availability
The availability can be described as the property of being accessible and useable upon demand by an authorized entity.

In the context of service level agreements, availability generally refers to the degree to which a system may suffer degradation or interruption in its service to the customer as a consequence of failures of one or more of its parts.

binding
A binding is an explicit established association, bonding, or tie.

broker service (BS)
The Broker Service is a helpful component for Service Providers to connect to the FutureID infrastructure and use the various authentication tokens (national eID cards, electronic health cards, electronic signature cards etc.) connected to the FutureID Client.

The Broker Service will evaluate the policy information provided by the Service Provider to determine an appropriate Identity Provider, Authentication Service or another Broker Service to which the request is forwarded (Dispatcher Mode).

If necessary or convenient, the Broker Service may aggregate different identity sources and produce an appropriate short term credential as requested by the Service Provider. In a similar manner the Broker Service may issue a long term minimum disclosure credential to the user (Claims Transformer Mode).

certificate

A certificate is an affidavit whereby a certification body attests to the truth of certain stated facts. In identity management, the term is often used to refer to a public-key digital certificate. X.509 Digital Certificates are a prominent example thereof.

certification authority (CA)

A Certification Authority is an entity that certifies public keys. Specifically, it guarantees the relationship between the identified entity and the public verification key. This association is achieved in a digital certificate that binds the public key to a partial identity of an entity.

It is generally a trusted party or trusted third party that accepts the responsibility of managing the certificate process by issuing, distributing and verifying certificates.

claim

A claim is an statement made by one entity about itself or another entity that a relying party considers to be in doubt until it passes Claims Approval.

confidentiality

Confidentiality means keeping the content of information secret from all entities except those that are authorized to access it.

credential

A credential is:

i. An identifiable object that can be used to authenticate the claimant is what it claims to be and authorize the claimant's access rights.

ii. Data that is transferred to establish the claimed identity of an entity.

iii. The private part of a paired identity assertion (user-id is usually the public part). The thing(s) that an entity relies upon in an assertion at any particular time, usually to authenticate a claimed identity. Credentials can change over time and may be revoked.

Examples include: a signature, a password, a drivers licence number (not the card itself), an ATM card number (not the card itself), data stored on a smart-card (not the card itself), a digital certificate, a biometric template.

cryptographic protocol

A cryptographic protocol is a distributed algorithm defined by a sequence of steps precisely specifying the actions required of two or more entities to achieve a specific security objective.

data

Data is a carrier of information. This information is encoded in a specific physical representation, usually a sequence of symbols that have meaning and can be processed or produced by a computer.

database
A database is a data repository, in which a collection of information organized in such a way that a computer program can quickly select desired pieces of data. Traditional databases are organized by fields, records, and files.

digital signature
A digital signature is data appended to, or a cryptographic transformation of, a data unit that allows a recipient of the data unit to prove the authenticity and integrity of the data unit.

See also electronic signature.

electronic signature
An electronic signature is data in electronic form which is attached to or logically associated to other electronic data and serves as a method of authentication.

From a legal perspective, an electronic signature is not necessarily considered equivalent to a handwritten signature. When it meets a number of conditions, it can be put on par with a handwritten one.

entity
An entity is an item of interest, inside or outside a system, such as an automated process, a subsystem, a device, a person or group of persons that incorporates a specific set of attributes.

federated identity
A federated identity is a partial identity which is the result of federation, and which usually implies that the entity to which this identity corresponds is recognized by several service providers or applications that are part of the federation.

identity
The identity of an entity is the dynamic collection of all of the entity's attributes. As such, the identity of an entity this is more a fluid and evolving ("philosophical") concept, rather than a practical one.

An entity has only one identity, but it is neither possible nor desirable for an information system to gather all the attributes of a specific entity. Information systems focus on a specific subset of relevant attributes, commonly referred to as 'partial identities'.

identity management (IdM)
The definition, designation and administration of identity attributes and the management of access to and the usage of applications, services and resources.

integrity
Integrity implies that the items of interest (facts, data, attributes etc.) have not been subject to manipulation by unauthorized entities.

intervenability
Intervenability is a data protection goal that ensures that data subjects, operators and supervisory authorities can intervene in all privacy-relevant data processing where necessary.

name
A name is the identifier of an entity (e.g., subscriber, network element) that may be resolved/translated into an address.

person
A person can be a natural person (a human being) or a legal person.

personal data
Personal data is any information relating to an identified or identifiable natural person (the data subject).

policy
A policy is one or more definite goals, courses or methods of action to guide and determine present and future decisions.

Policies are implemented or executed within a particular context (such as policies defined within an organization or business unit) or across contexts (e.g., in the case of an identity federation).

Common examples of such policies are security policies, privacy policies, access control policies, registration policies etc.

profile
A profile of an entity or a group of entities is an organized set of attributes that characterizes the specific properties of that entity or entities within a given context for a specific purpose.

protocol
A protocol may be described as a set of rules (i.e., formats and procedures) to implement and control some type of association (e.g., communication) between systems (e.g. an internet protocol).

In more general terms, a protocol can be qualified a series of ordered steps involving computing and communication that are performed by two or more system entities to achieve a joint objective.

revocation
Revocation is the act (by someone having the authority) of annulling something previously done.

role A role is a set of one or more authorisations related to a specific application or service.

sector
A sociological, economic, or political subdivision of society.

service
A service is a digital entity comprising software, hardware and / or communications channels that interacts with subjects.

service provider (SP)
This is an entity that provides services to other entities.

token
A token is any hardware or software component that contains credentials related to attributes. They may take any form, ranging from a digital data set to smart cards or mobile phones. They can be used for both data/entity authentication and authorization purposes.

transparency
Transparency is a data protection/privacy goal that ensures that all privacy-relevant data processing including the legal, technical and organizational setting can be understood by all involved parties including the Users.

trust
Trust is a measure of reliance on the character, ability, strength, or truth of someone or something.

trust infrastructure
The technical infrastructure used by system entities in order to trust each other. Trust infrastructures may be built on Public Key Infrastructure, Kerberos, etc.

trust service (TS)
The Trust Service provides a comprehensive repository for trusted certificates and services, service meta data for trusted providers and other trust-related information. It is also used to validate digital signatures.

unlinkability
Unlinkability is a privacy goal that ensures that all data processing is operated in such a way that the privacy-relevant data cannot be linked across privacy domains or used for different purposes than initially intended.

validation
Validation means confirming that information given is correct, often by seeking independent corroboration or assurance.

verification
The process or an instance of establishing the truth or validity of something.

verifier
Entity that corroborates a claim with a specified or understood level of confidence.

2.6 List of References

CEN CWA 15171 *General guidelines for electronic signature verification*, May 2004, ftp://ftp.cen.eu/CEN/Sectors/TCandWorkshops/Workshops/eSIGN_CWAs/cwa14171-00-2004-May.pdf

EC 1999/93 *Directive 1999/93/EC of the European Parliament and of the Council of 13 December 1999 on a Community framework for electronic signatures*, Dec 1999, http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=CELEX:31999L0093:EN:NOT

ETSI TS 101 733 *CMS Advanced Electronic Signatures (CAdES)*, v1.8.3, Jan 2011, http://www.cryptopro.ru/sites/default/files/products/tsp/ts_101733v010803p.pdf

ETSI TS 101 903 *XML Advanced Electronic Signatures (XAdES)*, v1.4.2, Dec 2010, http://uri.etsi.org/01903/v1.4.1/ts_101903v010401p.pdf

ETSI TR 102 038 *XML format for signature policies*, v1.1.1, Apr 2002, http://docbox.etsi.org/EC_Files/EC_Files/tr_102038v010101p.pdf

ETSI TR 102 272 *ASN.1 format for signature policies*, v1.1.1, Dec 2003, http://www.etsi.org/deliver/etsi_tr/102200_102299/102272/01.01.01_60/tr_102272v010101p.pdf

ETSI TS 102 778-3 *PDF Advanced Electronic Signature Profiles; Part 3: PAdES Enhanced - PAdES-BES and PAdES-EPES Profiles*, v1.2.1, Jul 2010, http://www.etsi.org/deliver/etsi_ts/102700_102799/10277803/01.02.01_60/ts_10277803v010201p.pdf

ETSI TS 102 778-4 *PDF Advanced Electronic Signature Profiles; Part 4: PAdES Long Term - PAdES-LTV Profile*, v1.1.2, Dec 2009, http://www.etsi.org/deliver/etsi_ts/102700_102799/10277804/01.01.02_60/ts_10277804v010102p.pdf

ETSI TS 102 853 *Signature verification procedures and policies*, v1.1.1, Jul 2012, http://www.etsi.org/deliver/etsi_ts/102800_102899/102853/01.01.01_60/ts_102853v010101p.pdf

ETSI TS 102 918 *Associated Signature Containers (ASiC)*, v1.2.1, Feb 2012, http://www.etsi.org/deliver/etsi_ts/102900_102999/102918/01.02.01_60/ts_102918v010201p.pdf

ETSI TS 103 171 *XAdES Baseline Profile*, v2.1.1, Mar 2012, http://www.etsi.org/deliver/etsi_ts/103100_103199/103171/02.01.01_60/ts_103171v020101p.pdf

ETSI TS 103 172 *PAdES Baseline Profile*, v2.2.2, Apr 2013, http://www.etsi.org/deliver/etsi_ts/103100_103199/103172/02.02.02_60/ts_103172v020202p.pdf

ETSI TS 103 173 *CAdES Baseline Profile*, v1.1.1, Sep 2011, http://www.etsi.org/deliver/etsi_ts/103100_103199/103173/01.01.01_60/ts_103173v010101p.pdf

ETSI TS 103 174 *ASiC Baseline Profile*, v2.2.1, Jun 2013, http://www.etsi.org/deliver/etsi_ts/103100_103199/103174/02.02.01_60/ts_103174v020201p.pdf

FutureID-DOW *FutureID* Grant agreement, *Annex I - "Description of Work"*, 2012, https://dms-prext.fraunhofer.de/livelink/livelink.exe/overview/2625944

OASIS-DSS *OASIS Digital Signature Services (DSS)*, 2007, https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=dss

OCK 2010 Orthacker, C., et al., *Qualified Mobile Server Signature*, 2010 https://online.tugraz.at/tug_online/voe_main2.getvolltext?pCurrPk=52961

RFC 2119 Bradner, S., *Key words for use in RFCs to Indicate Requirement Levels*, 1997, https://tools.ietf.org/html/rfc2119

RFC 3161 Adams, C., et al., *Internet X.509 Public Key Infrastructure Time-Stamp Protocol (TSP)*, Aug 2001, https://tools.ietf.org/html/rfc3161

RFC 5280 Cooper, D., et al., *Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile*, May 2008, https://tools.ietf.org/html/rfc5280

RFC 6960 Santesson, S., et al., *Online Certificate Status Protocol - OCSP*, Jun 2013, https://tools.ietf.org/html/rfc6960

3 Project Description

The *FutureID* project seeks to build a comprehensive, flexible, privacy aware and ubiquitously usable identity management infrastructure for Europe, which integrates existing eID technology and trust infrastructures, emerging federated identity management services and modern credential technologies to provide a user-centric system for the trustworthy and accountable management of identity claims. The *FutureID* infrastructure will provide great benefits to all stakeholders involved in the eID value chain. Users will benefit from the availability of a ubiquitously usable open source eID client that is capable of running on arbitrary desktop computers, tablets and modern smart phones. *FutureID* will allow applications and service providers to easily integrate their existing services with the *FutureID* infrastructure, providing them with the benefits from the strong authentication offered by eIDs without requiring them to make substantial investments. This will enable service providers to offer this technology to users as an alternative to username/password based systems, providing them with a choice for a more trustworthy, usable and innovative technology. For existing and emerging trust service providers and card issuers *FutureID* will provide an integrative framework, which eases using their authentication and signature related products across Europe and beyond. To demonstrate the applicability of the developed technologies and the feasibility of the overall approach *FutureID* will develop two pilot applications and is open for additional applications, which want to use the innovative *FutureID* technology. *FutureID* is a three-year duration project funded by the European Commission Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 318424.

4 Introduction

4.1 Scope

The role of WP33, eSign Services, is the development of a framework that is capable to process digital signature related tasks, like signature creation and verification. The framework will be interfaced via the OASIS-DSS protocol. However, the design shall be flexible enough to be interfaced via different protocols and bindings. The various formats of advanced digital signatures will be connected to the framework as plugins allowing to select implementation of different vendors (FutureID-DOW).

4.2 Outline

The remainder of this document is structured as follows: In the next section we will summarize the requirements of other *FutureID* work packages regarding the eSign Service. This is followed by the requirements for the supported signature formats and access protocols. Thereafter we will discuss requirements regarding signature and verification policies. Requirements that have not been addressed so far are collected in the section Other Requirements, which is followed by the requirements regarding the interfaces.

4.3 Key Words

The key words MUST, MUST NOT, REQUIRED, SHALL, SHALL NOT, SHOULD, SHOULD NOT, RECOMMENDED, MAY, and OPTIONAL in this document are to be interpreted as described in RFC 2119.

5 Requirements from other *FutureID* Work Packages

This section provides requirements for the eSign Services that have been identified and derived from other deliverables of the *FutureID* project. This includes in particular the requirement analyses from WP22, the work packages of the *FutureID* client (WP3x) and should coordinate the requirements of the eSign Services with other *FutureID* requirement efforts.

5.1 WP21 - Vision, Approach and Inventory

This section defines requirements that are derived from the reference architecture as well as the business and use cases of *FutureID*. WP21 provides the vision, approach, and inventory of *FutureID*. It comprises the terminology, technology inventory, vision, reference architecture, and business and use case analyses of *FutureID*.

5.1.1 D21.4 - Reference Architecture

The deliverable D21.4 defines the reference architecture comprising the system components and interfaces for the *FutureID* infrastructure. The following requirements have been identified from deliverable D21.4:

No.	*Req. 5.1*
Description	The eSign Services MUST comply with the reference architecture.

5.1.2 D21.5 - Business and Use Case Analysis

The deliverable D21.5 provides the most promising and compelling use cases that are relevant for *FutureID*. The eSigning and Validation use case, as described in D21.5, is relevant for the eSign Services. The use case describes the granting of a bank loan to show the practicality and economics of using various eIDs and their associated digital signatures in typical transactions. The technical description of the use case mentioned various signature formats, signature schemes, signature tokens, and validation procedures that are already covered by the requirements for the eSign Services (cf. Section 6).

No.	*Req. 5.2*
Description	The eSign Services MUST support the business and use cases described in D21.5.

5.2 WP22 - Requirements Analysis

This section summarizes the requirements defined in the deliverables of WP22 which are relevant for the eSign Services. WP22 provides an overall requirement analysis for *FutureID*, which comprises technical, security, privacy, usability, socio economic, legal, and accessibility issues.

Please note that because of the different deadlines of the deliverables D22.x and D33.1 only limited information was available.

5.2.1 D22.1 - Technical Requirements

The technical requirements in the deliverable D22.1 consider mandatory and optional functionality of *FutureID*. Currently, no requirements have been identified from deliverable D22.1.

5.2.2 D22.2 - Security Requirements

The security requirements focus on the availability, confidentiality, and integrity of the *FutureID* infrastructure. The deliverable D22.2 mentioned the threat of malicious software running on the client platform. It assumes and expects from the *FutureID* infrastructure that the client is protected against malware. Therefore, the user is not threatened by malicious software, which runs adverse actions in parallel or eavesdrops on PIN entry. Further, D22.2 expects that service providers and tokens use appropriated cryptographic protocols to protect security-sensible information and to prevent unauthorized access. The following requirements have been derived from deliverable D22.2:

No.	*Req. 5.3*
Description	The eSign Services SHOULD use secure and standardized protocols (D22.2, OE.Client and OE.Provider).

No.	*Req. 5.4*
Description	The eID token SHOULD comply with relevant standards, cryptography requirements, and authentication protocols (D22.2, OE.Token).

No.	*Req. 5.5*
Description	The eID token MUST be tamper proof (D22.2, OE.Token).

No.	*Req. 5.6*
Description	The communication between the *FutureID* client, the service provider, and the broker service MUST only take place after a successful mutual authentication (D22.2, O.Authentication).

No.	*Req. 5.7*
Description	The eSign Services SHOULD ensure that no tracing of the user is possible (D22.2, O.Tracing).

No.	*Req. 5.8*
Description	The eSign Services MUST enforce appropriate access rules such that only authorized persons or instances are allowed to access the services (D22.2, O.Access).

No.	*Req. 5.9*
Description	The eSign Services, if used for authentication, MUST be resistant against replay attacks (D22.2, O.Replay).

5.2.3 D22.3 - Privacy Requirements

The privacy requirements consider privacy principles and privacy protection goals like unlinkability, transparency, and intervenability. The following requirements have been identified from deliverable D22.3:

No.	*Req. 5.10*
Description	The eSign Services SHOULD minimise the processing of personal data as far as possible.

No.	*Req. 5.11*
Description	The eSign Services SHOULD limit the possible linkage of personal data.

No.	*Req. 5.12*
Description	The eSign Services SHOULD prevent unauthorised access to personal data.

5.2.4 D22.4 - Usability Requirements

The usability requirements focus on the ease of use and learnability of *FutureID* services. Currently, no requirements have been identified from deliverable D22.4.

5.2.5 D22.5 - Socio Economic Requirements

The socio economic requirements focus on a successful application and deployment of eSign Services. Currently, no requirements have been identified from deliverable D22.5.

5.2.6 D22.6 - Legal Requirements

The legal requirements consider the current legal frameworks applicable to the *FutureID* infrastructure. In particular this address the legal topics of data protection, e-commerce and e-signatures in the various national and cross border legislations.

No.	*Req. 5.13* – electronic signing
Description	If *FutureID* infrastructure provides the function to sign electronically it MUST adhere to the applicable national restrictions/requirements for electronic signatures.

No.	*Req. 5.14* – restricted electronic certificates
Description	If the *FutureID* infrastructure uses electronic certificates with restrictions, it MUST use the electronic certificates within the borders of their eventual restrictions.

No.	*Req. 5.15* – electronic certificates T&C
Description	If the *FutureID* infrastructure uses electronic certificates which are restricted by the terms&conditions of the certificate service provider, it MUST use the electronic certificates within these restrictions.

No.	*Req. 5.16* – electronic certificates Member States
Description	*FutureID* infrastructure MUST use electronic certificates within the borders of eventual restrictions set by Member States.

5.2.7 D22.7 Accessibility and Inclusion Requirements

The accessibility and inclusion requirements make sure that the *FutureID* eSign Services are usable by everyone, regardless of ability. The following requirement has been identified from deliverable D22.7:

No.	*Req. 5.17*
Description	The eSign Services SHOULD support assistive technologies to make it accessible to people with different disabilities.

5.3 WP32 - eID Services

WP32 provides the eID Services for the *FutureID* client. It comprises a survey of existing eID and credential systems, requirement analysis, interface and module specification, and implementation tasks. Further, it considers the creation of CardInfo files for arbitrary credentials and legal issues of the eID Services.

5.3.1 D32.3 - Interface and Module Specification

The deliverable D32.3 specifies the eID Services and in particular the Add-on Framework of the *FutureID* client. The add-on framework allows adding additional functionality to the client by simply plug in a new component like a signature or a credentials management module. The following requirement has been identified from deliverable D32.3:

No.	*Req. 5.18*
Description	The eSign Services MUST comply with the add-on framework of the *FutureID* client.

No.	*Req. 5.19*
Description	The eSign-Services MUST use the eID-Service of the FutureID-Client to determine the currently available, potentially operational[1] and operational signature creation devices and use them for the creation of signatures upon request. If there is no local signature creation device available, the user may either provide one, (generate a key pair and) download a certificate or delegate the signing process to a remote signature service. The set of supported signature services and certification services MUST be configurable and extensible.

[1] A potentially operational signature creation device is characterized by the fact that it is capable to download a certificate from a appropriate certification service.

6 Signature Format Requirements

The eSign services module is supposed to support different signature format standards:

- CMS and CAdES
- XMLDSig and XAdES
- PAdES
- different forms: AdES-T, AdES-C, AdES-XL and AdES-A

6.1 General requirements

Here, we collect requirements that are independent from the underlying signature format.

No.	*Req. 6.1*
Description	The eSign services module MUST support the RSA signature scheme with 2048 bit or larger keys and the ECDSA signature scheme with 256 bit or larger keys.

No.	*Req. 6.2*
Description	The eSign services module MUST support SHA-256 or a stronger hash function. The hash functions SHA256, SHA384 and SHA512 are, thus, supported.

No.	*Req. 6.3*
Description	The eSign services module MUST include the signer's certificate and all certificates not available to verifiers that can be used during path building. NOTE: It might be necessary to provide unser interfaces to select those chain certificates.

No.	*Req. 6.4*
Description	The eSign services module MUST be able to retrieve the revocation data of all the certificates in the signature in the form of an OCSP response (RFC 6960) or CRL (RFC 5280).

No.	*Req. 6.5*
Description	The generator MUST support IETF-style (RFC 3161) time-stamps.

In the following we list the requirements per format.

6.2 Requirements for XAdES signatures

XAdES (XML Advanced Electronic Signatures) is a set of extensions to XML-DSig recommendation making it suitable for advanced electronic signature. While XML-DSig is a general framework for digitally signing documents, XAdES specifies precise profiles of XML-DSig for use

with advanced electronic signature in the meaning of European Union Directive 1999/93/EC (EC 1999/93). One important benefit from XAdES is that electronically signed documents can remain valid for long periods, even if underlying cryptographic algorithms are broken.

No.	*Req. 6.6*
Description	The signature MUST conform to XAdES standard (ETSI TS 101 903, v1.4.2) and its Baseline Profile (ETSI TS 103 171, v2.1.1).

No.	*Req. 6.7*
Description	The eSign service MUST support XAdES-BES, XAdES-LT and XAdES-LTA levels.

No.	*Req. 6.8*
Description	The eSign service MUST support XAdES signature with ASiC-E container type.

No.	*Req. 6.9*
Description	As a result of *Req. 6.2*, the supported `ds:DigestMethod` element's Algorithm attribute values are at least: • http://www.w3.org/2001/04/xmlenc#sha256 • http://www.w3.org/2001/04/xmldsig-more#sha384 • http://www.w3.org/2001/04/xmlenc#sha512

No.	*Req. 6.10*
Description	As a result of *Req. 6.1* and *Req. 6.2*, at least the following values are possible in the `ds:SignatureMethod` element: • http://www.w3.org/2001/04/xmldsig-more#rsa-sha256 • http://www.w3.org/2001/04/xmldsig-more#rsa-sha384 • http://www.w3.org/2001/04/xmldsig-more#rsa-sha512 • http://www.w3.org/2001/04/xmldsig-more#ecdsa-sha256 • http://www.w3.org/2001/04/xmldsig-more#ecdsa-sha384 • http://www.w3.org/2001/04/xmldsig-more#ecdsa-sha512

No.	*Req. 6.11*
Description	Required certificates (see *Req. 6.3*) MUST be stored in `ds:KeyInfo` element.

No.	*Req. 6.12*
Description	The generator MUST support the following canonicalization methods: • http://www.w3.org/2006/12/xml-c14n11 • http://www.w3.org/2001/10/xml-exc-c14n# • http://www.w3.org/TR/2001/REC-xml-c14n-20010315 The default method is http://www.w3.org/2001/10/xml-exc-c14n#.

No.	*Req. 6.13*
Description	The generator MUST be able to bind the source data and signatures together in a single data unit, e.g. the ASiC-E container.

No.	*Req. 6.14*
Description	Multiple signatures MUST be supported in the form of parallel signatures.

LT-Level signatures

LT-Level profiles the incorporation of all the material required for validating the signature in the signature. This level is understood to tackle the long term availability of the validation material.

No.	*Req. 6.15*
Description	Proof of existence of the signature at a certain point in time MUST be provided in the signature by including a time-stamp token in `xades:SignatureTimeStamp` element. Only IETF-style (RFC 3161) time-stamps MUST be used, i.e. `xades:EncapsulatedTimeStamp` is supported only.

No.	*Req. 6.16*
Description	The generator MUST retrieve revocation data (see *Req. 6.6*). The revocation data MUST be included in `xades:RevocationValues` element.

LTA-Level signatures

LTA-Level profiles the incorporation of time-stamp tokens that allow validation of the signature long time after its generation. This level is understood to tackle the long term availability and integrity of the validation material.

No.	*Req. 6.17*
Description	Only `xadesv141:ArchiveTimeStamp` elements MUST be supported for LTA-level signatures. Only IETF-style (RFC 3161) time-stamps, that is `xades:EncapsulatedTimeStamp`, MUST be supported.

Requirements for ASiC container

A number of application environments use ZIP based container formats to package sets of files together with meta-information. ASiC technical specification is designed to operate with a range of such ZIP based application environments. Rather than enforcing a single packaging structure ASiC describes how these package formats can be used to associate advanced electronic signatures with any data objects.

No.	*Req. 6.18*
Description	The container MUST conform to ASiC standard (ETSI TS 102 918, v1.2.1) and its Baseline Profile (ETSI TS 103 174, v2.2.1).

No.	*Req. 6.19*
Description	The generator MUST support the ASiC-E container type.

No.	*Req. 6.20*
Description	META-INF/manifest.xml file MUST be included in the container. The file MUST not be signed.

A sample file for XAdES-LTA and ASiC-E can be found in the Appendix.

6.3 Requirements for PAdES signatures

PAdES (PDF Advanced Electronic Signatures) is a set of restrictions and extensions to PDF and ISO 32000-1 making it suitable for advanced electronic signature. This is published by ETSI as TS 102 778.

No.	*Req. 6.21*
Description	The signature MUST conform to PAdES standards ETSI TS 102 778-3 (v1.2.1), ETSI TS 102 778-4 (v1.1.2), and PAdES Baseline Profile (ETSI TS 103 172, v2.2.2) according to the signature's level.

No.	*Req. 6.22*
Description	The eSign service MUST support PAdES signatures based upon CAdES signatures (specified in ETSI TS 101 733, v1.8.3).

No.	*Req. 6.23*
Description	The eSign service MUST support PAdES-BES, PAdES-LT and PAdES-LTA levels (according to PAdES Baseline Profile ETSI TS 103 172, v2.2.2).

No.	*Req. 6.24*
Description	Minimum supported PDF version MUST be 1.7.

No.	*Req. 6.25*
Description	PDF/A standard MUST be supported.

No.	*Req. 6.26*
Description	Multiple signatures MUST be supported in the form of serial signatures.

No.	*Req. 6.27*
Description	Required certificates (see *Req. 6.3*) MUST be stored in the element `SignedData.certificates`.

LT-Level signatures

LT-Level profiles the incorporation of all the material required for validating the signature in the signature. This level is understood to tackle the long term availability of the validation material.

No.	*Req. 6.28*
Description	Proof of existence of the signature at a certain point in time MUST be provided in the signature by including a time-stamp token in `signature-time-stamp` attribute. Only IETF-style (RFC 3161) time-stamps MUST be used.

No.	*Req. 6.29*
Description	The generator MUST retrieve revocation data (see *Req. 6.4*). The revocation data MUST be included in Document Security Store (DSS).

LTA-Level signatures

LTA-Level profiles the incorporation of time-stamp tokens that allow validation of the signature long time after its generation. This level is understood to tackle the long term availability and integrity of the validation material.

No.	*Req. 6.30*
Description	The generator MUST add `document-time-stamp` to the signature.

6.4 Requirements for CAdES signatures

CAdES (CMS Advanced Electronic Signatures) is a set of extensions to Cryptographic Message Syntax(CMS) signed data making it suitable for advanced electronic signature. While CMS is a general framework for digitally signing documents such as E-Mail(S/MIME) or PDF, CAdES specifies precise profiles of CMS signed data for use with advanced electronic signature in the meaning of European Union Directive 1999/93/EC.

No.	*Req. 6.31*
Description	The signature MUST conform to CAdES standard (ETSI TS 101 733, v1.8.3) and its Baseline Profile (ETSI TS 103 173, v1.1.1).

No.	*Req. 6.32*
Description	The eSign service MUST support CAdES-BES, CAdES-LT and CAdES-LTA levels.

No.	*Req. 6.33*
Description	The eSign service MUST support CAdES signature with ASiC-E container type.

No.	*Req. 6.34*
Description	Required certificates (see *Req. 6.3*) MUST be stored in the element `SignedData.certificates`.

No.	*Req. 6.35*
Description	The generator MUST be able to bind the source data and signatures together in a single data unit, i.e. the ASiC-E container.

No.	*Req. 6.36*
Description	Multiple signatures MUST be supported in the form of parallel signatures.

LT-Level signatures

LT-Level profiles the incorporation of all the material required for validating the signature in the signature. This level is understood to tackle the long term availability of the validation material.

No.	*Req. 6.37*
Description	Proof of existence of the signature at a certain point in time MUST be provided in the signature by including a time-stamp token in `signature-time-stamp` attribute. Only IETF-style (RFC 3161) time-stamps MUST be used.

No.	*Req. 6.38*
Description	The generator MUST retrieve revocation data (see *Req. 6.4*). The revocation data MUST be included in Document Security Store (DSS).

LTA-Level signatures

LTA-Level profiles the incorporation of time-stamp tokens that allow validation of the signature long time after its generation. This level is understood to tackle the long term availability and integrity of the validation material.

No.	*Req. 6.39*
Description	The generator MUST include `archive-time-stamp` attribute in the signature. Only IETF-style (RFC 3161) time-stamps MUST be supported.

Requirements for ASiC container format

A number of application environments use ZIP based container formats to package sets of files together with meta-information. ASiC technical specification is designed to operate with a range of such ZIP based application environments. Rather than enforcing a single packaging structure ASiC describes how these package formats can be used to associate advanced electronic signatures with any data objects.

No.	*Req. 6.40*
Description	The container MUST conform to ASiC standard (ETSI TS 102 918, v1.2.1) and its Baseline Profile (ETSI TS 103 174, v2.2.1).

No.	*Req. 6.41*
Description	The generator MUST support the ASiC-E container type.

7 OASIS-DSS Protocol Requirements

The Digital Signature Services (DSS) standard, specified by the OASIS consortium[2] describes two request/response protocols for processing digital signatures:

- a signing protocol
- a verifying protocol

The specification consists of Core Protocols, Elements and Bindings. Interfaces implementing these protocols can then be used to:

- sign pieces of data with an electronic signature
- verify signatures on a given piece of data
- verify that a signature was created within its key validity period

While the specification can be used and adapted for various use cases, there already exist a number of profiles geared towards more specific use cases such as:

- XML Timestamping Profile
- Signature Gateway Profile
- German Signature Law Profile
- Entity Seal Profile
- Electronic PostMark (EPM) Profile
- Abstract Code-Signing Profile
- J2ME Code-Signing Profile
- Asynchronous Processing Abstract Profile
- Advanced Electronic Signature Profiles

XML and CMS, as well as a range of other signature formats are supported by DSS.

7.1 Benefits

When implementing DSS as web services, the above mentioned functionality can be used remotely. A client may send data or documents to a remote, specialized server to receive back a signature on the documents or to check whether a provided signature verifies the given data or document. This has the benefit of having the keys and other aspects needed by the signing service being held and managed centrally on a server.

No.	*Req. 7.1*
Description	The eSign service MUST provide an interface for the OASIS-DSS protocol.

[2] https://www.oasis-open.org

No.	*Req. 7.2*
Description	The eSign service SHOULD be able to support different interface protocols.

8 Signature Policy Requirements

8.1 Introduction

Signature policies govern technical details on how to create and validate electronic signatures. While also non-technical aspects are involved, we are only concerned with policy elements that influence the way a piece of software that creates or validates an electronic signature behaves. A policy can be implicitly or explicitly stated. ETSI TS 102 853 says:

> The validation process is controlled by a set of validation constraints in use. These constraints may be defined:
>
> - using formal policy specifications, e.g., in one of the standard policy formats ASN.1 (ETSI TR 102 272) or XML (ETSI TR 102 038); or
> - explicitly in system specific control data: e.g., in conventional configuration-files like property or in-files or stored in a registry or database; or
> - implicitly by the implementation itself.
>
> Additionally constraints may be provided by the DA to the SVA via parameters implied by the application or the user.

The same holds for generation of a signature. The effective policy in use is then a combination of any elements influencing the behaviour of the generation/validation software.

Annex A of ETSI TS 102 853 lists validation constraints that need to be supported by a validation module. Some of the constraints are taken from other standard documents, especially from RFC 5280.

The ETSI specifications currently allow the specification of the following policy elements:

- Common rules: rules that are common to all commitment types
- Commitment rules: specific rules for selected types of commitments
- Signer rules: rules applicable to the signer
- Verifiyer rules: rules applicable to the verifier
- Certifiate requirements: identifies a set of self signed certificates for the trust points used to start (or end) certificate path processing
- Revocation requirements: specifies minimum requirements for revocation information for checking certificate revocation
- Signing certificate trust conditions: identifies trust conditions for certificate path processing used to validate the signing certificate.
- Time-Stamp trust conditions: identifies trust conditions for certificate path processing used to authenticate the timstamping authority and constraints on the name of the timestamping authority.

- Attribute trust conditions: can require a signature to contain a specific attribute to make it acceptable
- Algorithm constraints: specifies acceptable algorithm for specific usages (signatures, CA certificates, end entity certificates, Attribute and Time Stamping certificates

Currently, European Standardisation Groups are working on new version of standards related with digital signature under a mandate from the European Commission[3]. Among those, new documents on signature policies will be published. First drafts will be available during 2013 and *FutureID*-partners are involved in the standardisation process.

No.	*Req. 8.1*
Description	The eSign service MUST support the validation constraints from RFC 5280.

8.2 Combining and Linking Policies

In *FutureID*, an average user will rarely be able to take highly complex decisions on how to configure her client software to create and validate signatures apropriately. She will in many cases reuse other trusted people or organisations suggestions. In other cases employees are required to use their company policy. Nevertheless the local client will need to have a local configuration. Thus, the local configuration must be able to incorporate policies from trusted sources.

In addition, some settings, like acceptable algorithms or key sizes or lists of trust anchors, may be delegated to competent authorities, like national security agencies or the European Commission, if found trustworthy to do so.

No.	*Req. 8.2*
Description	The eSign Service MUST use a local configuration file to configure the policy.

No.	*Req. 8.3*
Description	The eSign Service MUST be able to import settings from (potentially multiple) external policies. In case of conflicts of constraints, stronger constraint prevail or an error message MUST be thrown.

No.	*Req. 8.4*
Description	Imported policy files SHOULD be signed by the issuer, allowing validation of the authenticity of the policy while importing them. The eSign Service MUST validate the signatures of imported policies.

No.	*Req. 8.5*
Description	The eSign Service MUST use the *FutureID* Trustservice to decide, which trust anchors are considered trustworthy

[3] http://www.e-signatures-standards.eu

8.3 Formats

Currently, there are two specifications available that define formats for policies, ETSI TR 102 038 (XML Format for signature policies) and ETSI TR 102 272 (ASN.1 format for signature policies). As already mentioned, these standards will be superseded by new version in the near future. For *FutureID*, supporting both formats is not required.

No.	*Req. 8.6*
Description	*FutureID* Software MUST support ETSI TR 102 038 and it's successor, as published by ETSI.

8.4 Trust Models

Trust in signatures are derived from trust in certificate accompanying the signatures. There are two trust models that both have significant deployment:

- Centralised, hierarchical trust models (PKI)
- Web of Trust

8.4.1 Centralised, hierarchical trust models (PKI)

Here trust is derived from trust anchors, which typically are (self signed) certificates from trustworthy Certification Authorities. *FutureID* needs to support those, especially certificates from qualified Certification Authorities, which are widely used in Europe in the eGovernment and other sectors. The European Commission publishes a trust status list, which contains pointers to similar lists from different member states. Their lists contain information, including certificates, for all supervised and accredited Certification Authorities and other trust services. Such information will, in *FutureID*, be processed by the *FutureID* Trust Services (WP43) and made available to the eSign Service.

8.4.2 Web of Trust

The alternative is a decentralized web of trust, e.g., used for PGP[4], where keys are exchanged between individuals (or made available using key servers). Manual verification of the authenticity of the key is strongly recommended. One can import sets of keys from other people one trusts, effectively making them a small certification service certifying a (small) set of keys. Users of this trust model often distrust centralized trust services in general and prefer to make such decision themselves.

[4] http://www.pgpi.org

8.4.3 *FutureID*

Both trust models have their merits. They can easily be combined, by following the SPKI-model: The user is the "center of trust", he can decide to trust another user directly, a certain CA or a set of CAs (EU trust list) indirectly.

No.	*Req. 8.7*
Description	The *FutureID* infrastructure SHOULD support both trust models.

9 Signature Validation Requirements

Signature Validation is defined in ETSI TS 102 853 as

> the process of checking that a signature is valid including overall checks of the signature against local or shared signature policy requirements as well as certificate validation and signature verification.

where verification is defined as the process of checking the cryptographic value of a signature using signature verification data.

Inputs to signature validation are

- the signature (including the data to be signed)
- a signature validation policy (see correspondig section about policies) governing the details of the signature validation

The validation process outputs a status indication

- VALID, if
  - the signature is cryptographically valid
  - the signer's certificate consequently has been found trustworthy
  - the signature has been positively validated against the validation constraints and hence is considered conformant to these constraints.
- INVALID, if the signature has been found invalid according to the signature policy
- INDETERMINATE, if the available information was insufficient to ascertain the validity of the signature

No.	*Req. 9.1*
Description	The eSign service MUST support the basic building blocks of ETSI TS 102 853 including status code and validation report structures.

No.	*Req. 9.2*
Description	The eSign service MAY support time stamp validation and validation for AdES-T and long term validation from ETSI TS 102 853.

No.	*Req. 9.3*
Description	The validation reports MUST be user friendly and understandable (as much as possible) especially in case of failed validation. This denotes the fact that these validation reports can be very extensive and the actual reason for a failed verification might likely be hidden in the depths the validation report.

No.	*Req. 9.4*
Description	*FutureID* Software MUST conform to certificate validation as specified in RFC 5280

10 Other requirements

This part deals with other aspects, which are not covered elsewhere. In the following "other requirements" are non-functional requirements explained. Often non-functional requirements are only experienced and tested and measured, if the whole system is running. Similarly, non-functional requirements affect and depend on each other.

For example, a software requires a higher level of software security, hence additional security measures are required. The additional security measures extend the response time.

Non-functional requirements are typically more difficult to implement and guarantee.

No.	*Req. 10.1* – Portability and Interfaces
Description	The programming interface for eSign Service should be accessed via different operating systems. Therefore it is quite important that all components work independently from the underlying operating system. The framework should therefore provide an abstract interface to functions which are specific to an operating system. The eSign services provide a framework for other applications. So the framework should cover all requirements from the applications which interact with the eSign services. For interfaces both simplicity and clarity are important issues. The eSign services framework should have a simple and clear definition. This should make the implementation as easy as possible. To ensure good quality requirements, specification, software and others should be tested. The deliverable D37.1 already defined tasks related to testing. The interfaces should also be easy to test. Tests should happen in an automatic fashion.

No.	*Req. 10.2* – Documentation
Description	The FutureID client must be documented so that it can be installed, distributed and used by untrained users. For that reason the minimum requirement shall include a built-in-help-function and a user manual for the functions of the software. The documentation must be provided in the language of the user.

No.	*Req. 10.3* – Extensibility
Description	The FutureID client architecture should provide a mechanism to add future features and functions. Therefore the client-software must be able carry-forward customizations at next major version upgrades.

No.	*Req. 10.4* – Robustness
Description	The FutureID-client should fulfill all aspects of robustness. That means to implement the software in a way that misbehavior of the user is completely tolerated by the system (fault tolerance). Main focus is the process stability, clear error messages must be presented. The software should be as most as possible as compatible to the client operation system.

No.	*Req. 10.5* – Legal Conformity
Description	All legal and licensing issues or patent-infringement avoidabilities must be avoided. The country specific legal necessity must be implemented and - most important – automatically updated when changed by law. As an example we must consider a scenario where a Spanish user signs a German document with a German signature card according to the "Gesetz über Rahmenbedingungen für elektronische Signaturen" versus signing a Spanish document with the same signature card.

No.	*Req. 10.6* – Performance
Description	The client-software has to provide a suitable performance and effectiveness for private uses. Suitable parameters have to be defined in order to make the performance measurable, or verifiable. This includes the definition of a response time. It is planned to develop this client via performance engineering.

No.	*Req. 10.7* – Supportability and Maintainability
Description	The secure usage of the FutureID-client is to be eased by support services. Furthermore, the FutureID-client has to be maintainable. This means e.g. that error reports, log result documentation or similar can be used.

No.	*Req. 10.8* – Usability by Target User Community
Description	The FutureID client has to be usable by untrained personal. Therefore a graphical interface has to be available. This interface has to support multilingualism and has minimum requirements. Icons and corresponding design elements should be used.

10.1 Mobile signatures

There are several countries which use mobile signatures. The most relevant for FutureID seem the Austrian server-side solution, the Estonian Mobile ID[5], the Finnish Mobile Ink[6] provided by SICAP and the Finnish Mobile ID by Valimo. Apart from the Austrian solution, all services are SIM-based solutions, that is, solutions where the actual cryptographic operation is performed on the SIM card of the users mobile phone.

The Estonian Mobile ID was planned by the Baltic WPKI Forum[7] between 2006 and 2008. WPKI means PKI (Public-Key-Infrastructure) over a wireless medium. The specification[8] makes use of a CA infrastructure. It seems that the system is not widely adopted. There were some implementation approaches in 2007, but according to publicly available information there are not many advances. Both Finnish solutions provide marketing material about the solutions.

[5] http://e-estonia.com/components/mobile-id
[6] http://www.sicap.com/solutions/mink/mink
[7] http://wpki.eu/
[8] http://wpki.eu/doku/lib/exe/fetch.php/wiki:baltic_wpki_standard_draft-0.3.pdf

But no public available technical information could be found. Hence there was no possibility to find something about possible requirements. Furthermore there are some mobile signature solutions which are abandoned and are not used anymore. That is why it is hard to develop good, consistent requirements for SIM-based solutions. So it seems safe not to trace requirements for this type of mobile signatures. In the previous sections the eSign services were designed to be modular and easy extendible. This will make it easy to react on new requirements.

The relevant documents for the Austrian mobile signature are publicly available. The concept has been described in OCK 2010, the interface documentation and additional information can be found on the homepage of the Austrian citizen card[9].

No.	*Req. 10.9* – Support for server-side signatures
Description	The eSign Service SHOULD support server-side signature creation.

[9] http://www.buergerkarte.at/en/

Appendix

Sample file (XAdES-LTA + ASiC-E)

Container's structure

```
document.doc
mimetype
META-INF/manifest.xml
META-INF/signatures1.xml
```

Contents of "mimetype" file

```
application/vnd.etsi.asic-e+zip
```

Contents of "META-INF/manifest.xml" file

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE manifest:manifest PUBLIC "-//OpenOffice.org//DTD Manifest 1.0//EN" "Manifest.dtd">
  <manifest:manifest xmlns:manifest="urn:oasis:names:tc:opendocument:xmlns:manifest:1.0">
  <manifest:file-entry manifest:media-type="application/vnd.etsi.asic-e+zip" manifest:full-path="/" />
  <manifest:file-entry manifest:media-type="application/msword" manifest:full-path="document.doc" />
</manifest:manifest>
```

Contents of "META-INF/signatures1.xml" file

```
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<asic:XAdESSignatures xmlns:asic="http://uri.etsi.org/02918/v1.2.1#" xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
xmlns:xades="http://uri.etsi.org/01903/v1.3.2#" xmlns:xadesv141="http://uri.etsi.org/01903/v1.4.1#">
  <ds:Signature Id="S0">
    <ds:SignedInfo>
      <ds:CanonicalizationMethod Algorithm="http://www.w3.org/2006/12/xml-c14n11"/>
      <ds:SignatureMethod Algorithm="http://www.w3.org/2001/04/xmldsig-more#rsa-sha256"/>
      <ds:Reference Id="S0-RefId0" URI="document.doc">
        <ds:DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/>
        <ds:DigestValue>5UyKB9ht94y6CZNvLdO1C7Z3MXaYc2Qol3Dt3Qp4Ajg=</ds:DigestValue>
      </ds:Reference>
      <ds:Reference Id="S0-RefId1" Type="http://uri.etsi.org/01903#SignedProperties" URI="#S0-SignedProperties">
        <ds:DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/>
         <ds:DigestValue>YGDmd4GaWLgV4/hrEVV6/DvQ6uLhfnTSI0CQJX612KM=</ds:DigestValue>
      </ds:Reference>
    </ds:SignedInfo>
    <ds:SignatureValue Id="S0-SIG">
       YQs06u9ekMnZd2Jy+Won5VK0kIC9y5e2JPfraUItZOqwxd4rc4g3fiUnDkrf
       iHIdD2xOGyszCZA/JAicqDPiFkmXbjkgpYYF8gY3NB/xFwoKv/zaWu7HEi+T
       eq/OoSDlXVGi0H++27nI3xAl7P7Iz84xajilaquZQVl5iOtWD8k=
    </ds:SignatureValue>
    <ds:KeyInfo>
      <ds:X509Data>
        <ds:X509Certificate>
          MIIEnDCCA4SgAwIBAgIQfybdp3nKOMhPqk9YDxgaTTANBgkqhkiG9w0BAQU
          ...
          x3CqdYNWwQhU2bMirW4=
        </ds:X509Certificate>
      </ds:X509Data>
    </ds:KeyInfo>
    <ds:Object>
      <xades:QualifyingProperties Target="#S0">
        <xades:SignedProperties Id="S0-SignedProperties">
          <xades:SignedSignatureProperties>
            <xades:SigningTime>2013-05-09T15:49:32Z</xades:SigningTime>
            <xades:SigningCertificate>
              <xades:Cert>
                <xades:CertDigest>
```

```
                <ds:DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/>
                <ds:DigestValue>z/CsSIOu/w4lP63VzQEXRkxsT/oht2ggvA6rMxDQvoA=</ds:DigestValue>
              </xades:CertDigest>
              <xades:IssuerSerial>
                <ds:X509IssuerName>emailAddress=pki@sk.ee,CN=TEST of ESTEID-SK 2011,O=AS
                    Sertifitseerimiskeskus,C=EE</ds:X509IssuerName>
                 <ds:X509SerialNumber>169013758426626343561532977746185558605</ds:X509SerialNumber>
              </xades:IssuerSerial>
            </xades:Cert>
          </xades:SigningCertificate>
          <xades:SignatureProductionPlace>
            <xades:City>Tallinn</xades:City>
            <xades:StateOrProvince>Harju</xades:StateOrProvince>
            <xades:PostalCode>10122</xades:PostalCode>
            <xades:CountryName>Estonia</xades:CountryName>
          </xades:SignatureProductionPlace>
          <xades:SignerRole>
            <xades:ClaimedRoles>
              <xades:ClaimedRole>CEO</xades:ClaimedRole>
            </xades:ClaimedRoles>
          </xades:SignerRole>
        </xades:SignedSignatureProperties>
        <xades:SignedDataObjectProperties>
          <xades:DataObjectFormat ObjectReference="#S0-RefId0">
            <xades:MimeType>application/msword</xades:MimeType>
          </xades:DataObjectFormat>
        </xades:SignedDataObjectProperties>
      </xades:SignedProperties>
      <xades:UnsignedProperties>
        <xades:UnsignedSignatureProperties>
          <xades:SignatureTimeStamp Id="S0-T0">
            <xades:EncapsulatedTimeStamp>
              MIIMYzADAgEAMIIMWgYJKoZIhvcNAQcCoIIMSzCCDEcCAQMxCzAJBgUrDgMCGgUA
              ...
              ZSQAy4ewaA==
            </xades:EncapsulatedTimeStamp>
          </xades:SignatureTimeStamp>
          <xades:RevocationValues>
            <xades:OCSPValues>
              <xades:EncapsulatedOCSPValue Id="N0">
                MIIBtgoBAKCCAa8wggGrBgkrBgEFBQcwAQEEggGcMIIBmDCCAQGhcTBvMQswCQYD
                ...
                knf8XDhdklVD0w==
              </xades:EncapsulatedOCSPValue>
            </xades:OCSPValues>
          </xades:RevocationValues>
          <xadesv141:ArchiveTimeStamp Id="S0-A0">
            <xades:EncapsulatedTimeStamp>
              MIIMYzADAgEAMIIMWgYJKoZIhvcNAQcCoIIMSzCCDEcCAQMxCzAJBgUrDgMCGgUA
              ...
              ZSQAy4ewaA==
            </xades:EncapsulatedTimeStamp>
          </xadesv141:ArchiveTimeStamp>
        </xades:UnsignedSignatureProperties>
      </xades:UnsignedProperties>
    </xades:QualifyingProperties>
  </ds:Object>
 </ds:Signature>
</asic:XAdESSignatures>
```