All-in Signing Service

Reference Guide

Version: 2.8

Contents

1 Introduction ... 4
1.1 Terms and abbreviations ... 5
1.2 Referenced documents ... 7
2 Overview and main scenarios ... 8
2.1 Overview of Trusted Timestamps and Static CMS Signatures ... 9
2.2 Overview of On Demand CMS Signatures ... 9
2.2.1 Step-Up authentication ... 10
3 Preconditions and assumptions ... 11
3.1 Internet access ... 11
3.2 Certificate based client authentication ... 11
3.3 Request authorisation ... 12
3.3.1 DN format (On Demand signatures) ... 12
4 All-in Signing Service Introduction ... 13
4.1 Communication Modes ... 13
4.2 Type of Signatures ... 13
4.3 Adding Trusted Timestamps ... 13
4.4 Adding Revocation Information (long-term signature) ... 13
4.5 Declaration of Will (Step-Up Authentication) ... 14
4.6 Batch Processing ... 14
4.7 Detached Signature and Verification ... 14
5 All-in Signing Service Interface ... 15
5.1 Overview ... 15
5.1.1 Interface Description ... 15
5.1.2 HTTP/1.1 Header ... 15
5.1.3 Swisscom Basic Profile ... 15
5.1.4 Document Hash ... 15
5.1.5 Signing Options ... 16
5.1.6 On Demand Certificate Policy and Certification Practice Statement (CP/CPS) ... 23
5.2 Trusted Timestamp ... 24
5.2.1 Trusted Timestamp SignRequest ... 24
5.2.2 Trusted Timestamp SignResponse ... 24
5.3 CMS Signatures ... 27
5.3.1 CMS SignRequest for Static Signatures ... 27
5.3.2 CMS SignRequest for On Demand Signatures ... 29
5.3.3 CMS SignResponse ... 30
5.4 Asynchronous Mode ... 33
5.4.1 SignRequest ... 33
5.4.2 SignResponse ... 33
5.4.3 PendingRequest ... 34
5.4.4 PendingResponse ... 35
5.5 CMS On Demand Signatures with Step-Up Authentication ... 36
5.5.1 SignRequest ... 36
5.5.2 SignResponse ... 38
5.5.3 PendingRequest ... 39
5.5.4 PendingResponse ... 39
5.5.5 SignResponse (SUCCESS) ... 39
5.6 Static Plain Signatures (PKCS#1) ... 41
5.7 Fault Response Message ... 42
5.7.1 Wrong Digest Size (example) ... 42

5.7.2 Step-Up Authentication: Mobile ID User Account Problem (example) ... 42
5.7.3 Step-Up Authentication: User Cancel (example) ... 44
5.7.4 Step-Up Authentication: SerialNumber Mismatch (example) ... 45
5.8 Best Practices ... 46
5.8.1 Example of Codes and Scripts ... 46
5.8.2 On Demand Step-Up Pre-Signing Process ... 46
5.8.3 Signing Requests ... 47
5.8.4 Response handlings ... 47
5.8.5 Adobe PDF ... 47
6 Appendix ... 49
6.1 Create self-signed certificate with openssl ... 49
6.1.1 Generate Key and CSR ... 49
6.1.2 Self-sign it and create your certificate ... 49
6.1.3 Convert into PKCS#12 (if needed) ... 49
6.2 Create self-signed certificate with Java keytool ... 49
6.2.1 Generate KeyStore & export the self-signed certificate ... 49
6.2.2 Root CA and Intermediate CA certificate import ... 49
6.2.3 Verification ... 49
6.3 Swisscom signed certificate ... 49
6.4 Result Major and ResultMinor list ... 50
6.4.1 ResultMajor Status Codes ... 50
6.4.2 ResultMinor Status Codes ... 50
6.5 Swisscom CA Hierarchy ... 51
6.5.1 CMS Signature ... 51
6.5.2 Timestamp Signature ... 51
7 Migration Guide ... 52
7.1 New Profile ... 52
7.2 On-Demand Signatures with Declaration of Will ... 52
7.2.1 Additional Profiles in Signing Request ... 52
7.2.2 Step-Up Information in Signing Request ... 53
7.2.3 Signing Response (pending) ... 54
7.2.4 Pending Request ... 56
7.2.5 Pending Response (pending/success) ... 56
7.2.6 Error and Status Codes ... 57
7.2.7 User Cancel ... 58

1 Introduction

The purpose of this document is to give advice and support to integrators, developers and customers who have to implement the Swisscom All-in Signing Service, referred to as AIS, based on the OASIS Digital Signature Service (DSS) specification [OASIS DSS].

This manual assumes that you are familiar with general Web Services (SOAP, WSDL/WADL, XML, JSON and Application Server) as well as with the digital signing topic itself.

This guide refers to the newest version of the AIS Interface, which supports alternative signature authorisation mechanisms in addition to Mobile ID. The legacy version 1.x is still fully supported and documented in [AIS1x].

1.1 Terms and abbreviations

Abbreviation	**Definition**
ℹ	Please note.
⚠	Be careful, important.
AIS	Swisscom All-in Signing Service
AP	Application Provider
AS	An Application Server (AS) is a server which provides software applications with services.
Authentication	An authentication process verifies who a person is.
Authorisation	An authorisation process verifies the access rights of a given person/system and grants access.
CA	Certificate Authority
CMP	Certificate Management Protocol, an Internet protocol for obtaining X.509 digital certificates in a public key infrastructure.
CMS	The Cryptographic Message Syntax (CMS) is a standard for cryptographically protected messages. It can be used to digitally sign, digest, authenticate or encrypt any form of digital data. CMS is based on the syntax of PKCS#7.
CP/CPS	Certificate Policy (CP) and Certification Practice Statement (CPS)
DSS	Digital Signature Service: Declared XML Namespace(s): urn:oasis:names:tc:dss:1.0:core:schema
ERP	Enterprise Resource Planning is a business management software.
Hash value	A mapping of an original document into a smaller one such as a fingerprint made of integer numbers.
HSM	Hardware security module
JSON	JavaScript Object Notation is a text-based open standard designed for human readable data interchange. Although derived from the JavaScript scripting language it is language independent. The JSON format is often used for serializing and transmitting structured data over a network connection, primarily between a server and a web application, as an alternative to XML.
LTV	Digitally signed documents may be used or archived for many years – even many decades. At any time in the future, when the CA will have no obligations to make revocation information available, it must still be possible to verify that the signature was valid at the time it was created – a concept known as Long-Term Validation (LTV).
OASIS	Organization for the Advancement of Structured Information Standards (OASIS), consortium for the development, convergence, and adoption of e-business and web service standards, www.oasis-open.org
OCSP	Online Certificate Status Protocol is an Internet protocol used for obtaining the revocation status of a digital certificate.
RESTful	Representational State Transfer is a style of software architecture for distributed systems such as the World Wide Web. It is based on the existing design of HTTP/1.0. REST-style architectures consist of clients and servers. Clients initiate requests to servers; servers process requests and return appropriate responses.

Abbreviation	Definition
RFC	A Request for Comments (RFC) is a publication of the Internet Engineering Task Force (IETF) and the Internet Society, the principal technical development and standards-setting bodies for the Internet.
SOAP	Simple Object Access Protocol (SOAP) is a protocol specification for exchanging structured information in the implementation of Web Services relying on Extensible Markup Language (XML)
TSA	Time Stamp Authority
WS	A Web Service (WS) is a method of communication between two electronic devices over the Web (Internet). The W3C defines a "Web service" as "a software system designed to support interoperable machine-to-machine interaction over a network". It has an interface described in a machine-processable format (specifically Web Services Description Language, known by the acronym WSDL).
WSDL	The Web Services Description Language (WSDL) is an XML-based language that is used for describing the functionality offered by a Web service. A WSDL description of a web service provides a machine-readable description of how the service can be called, what parameters it expects, and what data structures it returns.
X.509	X.509 is a standard for a public key infrastructure. X.509 specifies, amongst other things, standard formats for public key certificates, certificate revocation lists, attribute certificates, and a certification path validation algorithm.
XML	Extensible Markup Language (XML) is a markup language that defines a set of rules for encoding documents in a format that is both human-readable and machine-readable.

1.2 Referenced documents

[AIS1x]	All-In Signing Service Reference Guide Version 1.x. with the legacy step-up syntax.
[CP/CPS]	Certification Practice Statement and Certificate Policy http://www.swissdigicert.ch/sdcs/portal/page?node=download_docs
[MIDSOAP]	Mobile ID Client Reference Guide http://www.swisscom.ch/mid see Technical details > Technical documents
[OASISDSS]	OASIS DSS Standard http://docs.oasis-open.org/dss/v1.0/oasis-dss-core-spec-v1.0-os.pdf
[PDFSIG]	Digital signatures in Adobe® Portable Document Format (PDF) http://www.adobe.com/devnet-docs/acrobatetk/tools/DigSig/
[RFC6960]	X.509 Internet Public Key Infrastructure, Online Certificate Status Protocol – OCSP http://www.ietf.org/rfc/rfc6960.txt
[RFC2986]	Certification Request Syntax Specification http://www.ietf.org/rfc/rfc2986.txt
[RFC3161]	X.509 Public Key Infrastructure, Time-Stamp Protocol (TSP) http://www.ietf.org/rfc/rfc3161.txt
[RFC3369]	Cryptographic Message Syntax (CMS) http://www.ietf.org/rfc/rfc3369.txt Note: this RFC has been obsoleted by RFC 3852
[RFC5126]	CMS Advanced Electronic Signatures (CAdES) http://www.ietf.org/rfc/rfc5126.txt
[RFC5652]	Cryptographic Message Syntax (CMS) - obsoletes RFC3369 and RFC3852 http://www.ietf.org/rfc/rfc5652.txt
[RFC3447]	Public-Key Cryptography Standards (PKCS) #1 https://www.ietf.org/rfc/rfc3447.txt

2 Overview and main scenarios

All-in Signing Service (AIS) allows customers' documents and files to be electronically signed: AIS itself cannot view the files and documents to be signed as only the hash values are transferred to the service. The signature types provided by AIS and applied to the hash values are **Trusted Timestamps** and **Cryptographic Message Syntax (CMS) Signatures**.

Trusted Timestamps

Trusted Timestamps applied to the hash values as signatures by AIS are qualified timestamps provided by a trusted third party Time Stamp Authority (TSA), according to the [RFC3161] standard. Timestamp signatures are used to prove the existence of certain data at a certain point in time without a person or an organization behind them. This kind of signature is well suited for system transactions and log files.

Additional clarifications can be found in Wikipedia[1].

CMS Signatures

The CMS (PKCS#7) is a standard for cryptographically protected messages. The AIS is using this signature type when it comes to digitally sign the hash values with a customer certificate (X509). This certificate used during this signature process can be either **Static** or **On Demand**. Timestamps can be additionally applied to CMS Signatures to define a trusted point in time or adhere to local regulations. Swiss qualified signatures must include a qualified timestamp.

Static certificates are standard ones proposed and issued by any official Certificate Authority (CA) for the customer and are securely hosted at the AIS on its Hardware Security Module (HSM). After the certificate's registration process, the corresponding customer can address and use it in a secure and exclusive manner. Static certificates are well suited for any organization planning to sign a large number of documents in its name in an automated manner, for example invoices, account listings, archives of documents.

For example, static certificates can be used to fulfil the requirements for the electronic archiving of business records in accordance with GeBüV.

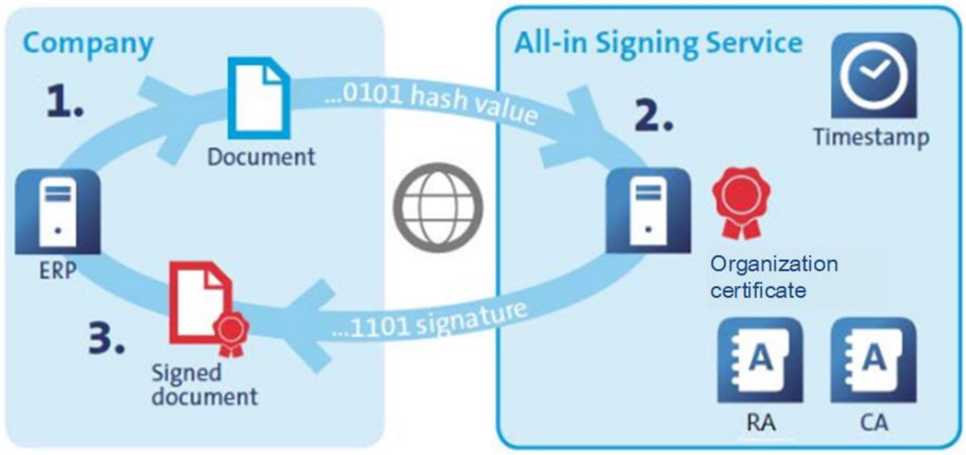
On Demand certificates are context-based issued certificates that will contain the end user information collected at the customer's service side itself. The collected information can be set as attributes in the Distinguished Name (DN) of the short-lived certificate. Before issuing the certificate and using it only for one request, a declaration of will by the signer is enforced. On Demand certificates are well suited for signing documents interactively/online such as contracts, medical assessments, construction permits, tax declarations…

Static Plain (PKCS#1)

RSASSA-PKCS1 [RFC3447] signatures are also provided and can be used for special purposes like for example the signing of EDIFACT or XML documents.

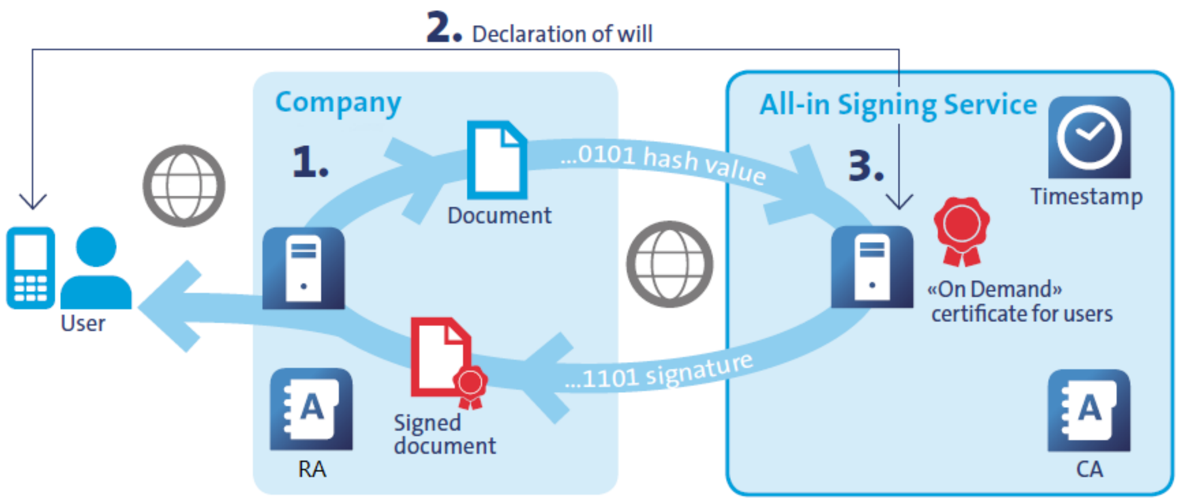
[1] Trusted timestamping: http://en.wikipedia.org/wiki/Trusted_timestamping

2.1 Overview of Trusted Timestamps and Static CMS Signatures



1. The customer (company) application server (Enterprise Resource Planning, ERP) calculates the hash value of the document to be timestamped/signed and sends it within a signature request to AIS.
2. AIS receives and validates the signature request. It timestamps or signs the hash value with the corresponding customer static certificate.
3. AIS sends the signature back to the customer. The signature is integrated in the document by the application server and stored.

2.2 Overview of On Demand CMS Signatures



The context is the following: an employee would like to sign documents interactively/online from the corporate intranet.

1. The customer (company) application server (Enterprise Resource Planning, ERP) calculates the hash value of the document to be signed and sends it within a signature request to AIS.

2. AIS receives and validates the signature request. Optionally, it first performs a declaration of will (step-up authentication: refer to next chapter). AIS requests a unique certificate issued by the CA with a short validity period and which will be used only once.
3. It signs the hash value with this certificate.
4. AIS sends the signature back to the customer. The signature itself is integrated by the application server in the related document and stored.

2.2.1 Step-Up authentication

Step-authentication is an optional client-based setting which is provided when onboarding to AIS. If step-up authentication is defined for a client, a step-up authentication will be enforced where the end user must approve the signature request. Currently, there are two available methods:

1. Mobile ID: the end user shall approve the signature and prove sole control through Mobile ID authentication. (Remark: in case the Mobile-ID check returns an error – for example if the Mobile-ID of the user is not activated – the step-up process fallbacks to Password & SMS-Challenge. An enforcement of only MobileID without this fallback is still possible while using the old request payload for the definition of the Step-Up method (refer to chapter 7.2.2)

2. Password & SMS-Challenge: the client application must redirect the user to a unique consent URL delivered by AIS in the signature response. On the site available under this URL, the user shall approve the signature and prove sole control through password and/or SMS-Challenge authentication.

If step-Up authentication is not enforced by Swisscom, the client is responsible to get the appropriate end user consent for the signature creation. The selected method must be documented and approved by Swisscom.

3 Preconditions and assumptions

Before using the All-in Signing Service some prerequisite steps are required.

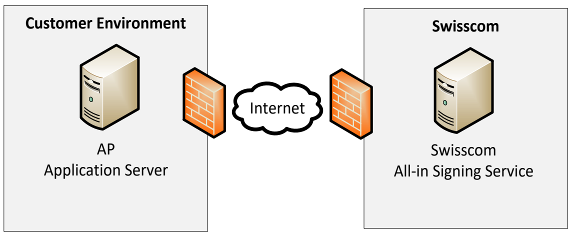
In this reference guide we assume that:

1. The customer has an agreement with Swisscom and is already provisioned on AIS.
2. The customer has received from Swisscom its AIS Claimed Identities and the relevant CA certificates.

3.1 Internet access

The AIS interface is accessible through Internet. The max amount of concurrent sessions is limited to 1500.

If not otherwise specified use the following default access configuration information:



Base-URL

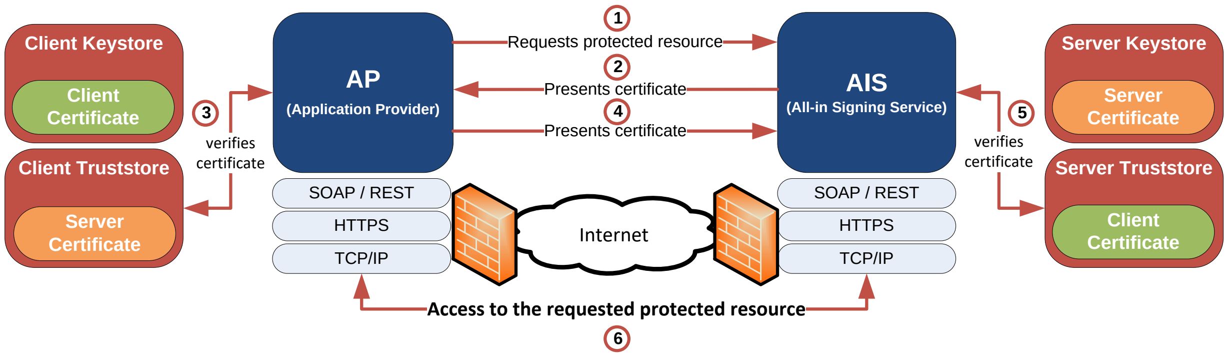
https://ais.swisscom.com

Service Endpoint(s):

SOAP:	<**Base-URL**>/AIS-Server/ws
RESTful:	<**Base-URL**>/AIS-Server/rs/v1.0/sign
	<**Base-URL**>/AIS-Server/rs/v1.0/pending

3.2 Certificate based client authentication

The AIS requires a certificate-based authentication[2] in order to identify the client, referred as the Application Provider (AP), and grant access:



1. The client AP requests access to a protected resource on AIS.
2. AIS presents its server certificate to the client AP.
3. The client AP optionally verifies the AIS server certificate.
4. If successful, the client AP sends its client certificate to AIS.
5. AIS verifies the AP client certificate.
6. If successful, AIS grants access to the protected resource requested by the client AP.

[2] http://en.wikipedia.org/wiki/Secure_Sockets_Layer#Client-authenticated_TLS_handshake

ⓘ Authentication at the AIS side does not consider any validation of a client certificate chain or restrictions of the root CA. **The client shall send only its end entity certificate**. The authentication is denied in case the client sends a bag with the full certificate chain.

ⓘ Enhanced Key Usage value of client certificates must contain Client Authentication (1.3.6.1.5.5.7.3.2). Chapter 6 contains examples on how to create self-signed certificates.

3.3 Request authorisation

AIS provides for each AP one or many Claimed Identities (see 5.1.5.8) in order to authorize the signing request. Each Claimed Identity must be used for the proper signature type mentioned in chapter 2.

3.3.1 DN format (On Demand signatures)

For On Demand signatures, as part of the contract document, each customer defines the DN format he will use in his AIS requests. The Distinguished-Name provided in the signing request is matched against the defined DN format: only requests with matching DNs are allowed; invalid DNs are rejected with a specific error message. It is therefore important that APs understand their agreed DN-format and make sure that valid DNs are provided in the signing requests.

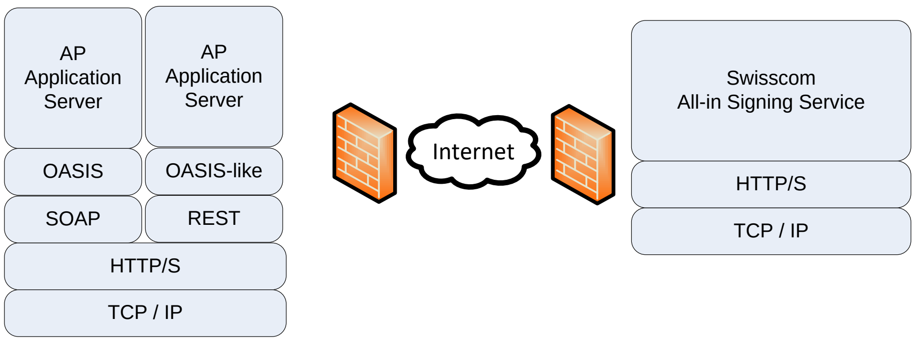
Here an example of such a DN-filter:

```
CN=<First name and last name of the customer>, pseudonym=<mobile number of the customer>,
O=My Company AG,C=ch
With the given DN-filter, the following processing results may apply
•   CN=John Doe,pseudonym=+41791234567,O=My Company AG, C=ch -> OK
•   CN=John Doe, O=My Company AG,C=ch -> fail (pseudonym missing)
•   pseudonym=+41791234567,CN=John Doe,O=My Company AG, C=ch -> fail (wrong sequence)
```

4 All-in Signing Service Introduction

The All-in Signing Service exposes its services over HTTPS in SOAP and RESTful in an [OASIS DSS] standard compliant form.

The following picture shows the available alternatives, from a protocol-stack point of view:



4.1 Communication Modes

The standard defines synchronous and asynchronous (client-server) modes and AIS supports both of them. When placing asynchronous requests, the result must be fetched within 15 minutes otherwise it will be dropped.

4.2 Type of Signatures

As defined in chapter 2, for both types of supported signatures, the signature part in the response is Base64 encoded and represents either a [RFC3161] compliant Trusted Timestamp or a [RFC3369] / [RFC5652] compliant CMS Signature. See chapter 5.1.5.1 for further details.

[OASIS DSS] is using urn:ietf:rfc:3369 for the request definition. AIS is using this to be compliant to the standard itself, but the provided answer is [RFC5652] compliant.

4.3 Adding Trusted Timestamps

For CMS signatures, an additional [RFC3161] Trusted Timestamp may be requested and applied. By asking this, the response will additionally contain a Trusted Timestamp. This will define a trusted point in time for the signature. See chapter 5.1.5.4 for further details.

4.4 Adding Revocation Information (long-term signature)

AIS supports the concept of long-term signature validation (LTV) which allows you to check the validity of a signature long time after the document was signed, when the CA will have no longer any obligations to make revocation information available. To achieve long-term validation, all the required revocation information for signature validation must be embedded in the signed document.

Without revocation information, a signature can be validated for only a limited time. This limitation occurs because certificates related to the signature eventually expire or are revoked. Once a certificate expires, the issuing authority is no longer responsible for providing revocation status on that certificate. Without conforming revocation information, the signature cannot be validated.

Revocation information may be requested for any type of Signature. By asking this, the response will additionally contain certificate status information (signed CRLs or OCSP responses, refer to chapter 6.5) for the signing certificate chain. See chapter 5.1.5.5 for further details.

4.5 Declaration of Will (Step-Up Authentication)

A declaration of will (subsequently referred to as *step-up authentication*) is an additional step in the signing process, which enforces the authorization of the requested signature through a method or device, which must remain under the sole control of the signature requestor. This step is mandatory for qualified signatures and optional for advanced ones.

AIS newly also allows the generation of qualified signatures without Step-up authentication. In such a customer configuration, the AP himself is responsible of guaranteeing that the physical person the signing request has been submitted for has given his allowance according to the local regulations.

AIS supports Mobile ID as default consent method and a web-based solution as alternative.

See chapter 5.5 for technical details regarding the signature request and response for this use case.

4.6 Batch Processing

AIS allows signing multiple hash values with a single request. Remarks:

- There is no limitation of the number of hash values in a single request.
- If any error occurs during the processing, the entire batch request will fail.
- For On Demand CMS Signatures, only one certificate is issued and used to sign all hashes.
- The step-up authentication takes place only once and is valid for the release of the entire batch.
- Each hash value can have its own digest algorithm.

A batch must always contain at least two documents. Do not use the batch functionality to send a batch containing one single element.

4.7 Detached Signature and Verification

Since only the document hash is provided to AIS, the returned signature itself is detached from the document per se. If the signature is not embedded into the document, then the signature is called a detached signature. For the verification process, both the detached signature and the document are required.

The verification proves the authentication, the integrity and non-repudiation of the signed hash value, respectively the document. Be aware that the verification process may require Internet connection to the CA online services (to an OCSP Responder or CRL source), unless you have requested revocation information to perform long-term signature validation (see chapter 4.4).

There are document formats that support the integration of such detached signatures, e.g. Portable Document Format (PDF). The CMS Signature provided by AIS can be used as is for integrated signatures. For PDFs refer to [PDFSIG] chapter 8.7.2 and to sample code referred in chapter 5.8.1. The customer must perform the integration of the signature into the document by using libraries or partner solutions (referred on AIS website [3]).

[3] http://swisscom.ch/signing-service

5 All-in Signing Service Interface

5.1 Overview

5.1.1 Interface Description

A client who wants to connect to the All-in Signing Service can read the related **Web Service Description Language (WSDL)** file for SOAP or the **Web Application Description Language (WADL)** file for RESTful to determine what functions are available. Any special data types used are embedded in those files in the form of XML Schemas. It describes how the service can be called, what parameters it expects, and what data structures it returns.

WSDL: https://ais.swisscom.com/AIS-Server/ws/?wsdl

WADL: http://git.io/tPMdpQ

5.1.2 HTTP/1.1 Header

You can POST signature requests via HTTP/1.1 using the SOAP or RESTful interface. The RESTful interface supports two media types: XML and JavaScript Object Notation (JSON).

The header fields should be set as follows:

HEADER FIELD	SOAP	RESTFUL/XML	RESTFUL/JSON
Content-Type	text/xml;charset=UTF-8	application/xml;charset=UTF-8	application/json;charset=UTF-8
Accept	-	application/xml	application/json

5.1.3 Swisscom Basic Profile

The Swisscom Basic Profile describes AIS extensions based on the [OASIS DSS] interface. A version number ensures backward compatibility in case of future improvements.

The Profile URI must be specified as an attribute in every <SignRequest> element.

Example:

```
SOAP/XML:  <SignRequest Profile="http://ais.swisscom.ch/1.1">
JSON:      "SignRequest": {"@Profile": "http://ais.swisscom.ch/1.1"}
```

5.1.4 Document Hash

5.1.4.1 Digest Method

AIS supports the following digest algorithm URIs.

- SHA-256: http://www.w3.org/2001/04/xmlenc#sha256
- SHA-384: http://www.w3.org/2001/04/xmldsig-more#sha384
- SHA-512: http://www.w3.org/2001/04/xmlenc#sha512

Example:

```
SOAP/XML:  <dsig:DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha512"/>
JSON:      "dsig.DigestMethod": {"@Algorithm": "http://www.w3.org/2001/04/xmlenc#sha512"}
```

5.1.4.2 Digest Value

The document hash (digest) value shall be in a **Base64** encoded **binary form**.

Here is an example how to get the Base64 encoded binary SHA-256 digest of a document using OpenSSL:

```
$ openssl dgst -binary -sha256 <document> | openssl enc -base64
-A 1WON4H3Hrinf7LYRNmhV6Uf7apdUvuYEsmhxAklxumA=
```

5.1.5 Signing Options

With the element <OptionalInput>, it is possible to include additional parameters to each signing request.

- Signature Type, see 5.1.5.1
- Signature Standard, see 5.1.5.3
- Additional Profiles, see 5.1.5.3
- Add Timestamp, see 5.1.5.4
- Add Revocation Information (long-term validation), see 5.1.5.5
- On Demand Distinguished Name, see 5.1.5.6
- On Demand Step Up Authorisation, see 5.1.5.7
- Claimed Identity, see 5.1.5.8

5.1.5.1 Signature Type

The element <SignatureType> defines the type of signature to be applied to the hash. There are currently three types of signature provided by AIS. Exactly one type of the following URN must be given in the request:

- **urn:ietf:rfc:3161**
  Creation of a Trusted Timestamp according to [RFC3161]. The response contains a Base64 encoded timestamp token. Optionally it may contain additional **revocation information** (chapter 5.1.5.5).

- **urn:ietf:rfc:3369**
  Creation of a CMS Signature according to [RFC5652]. The response contains a Base64 encoded signature. Optionally it may contain additional embedded **revocation information** (chapter 5.1.5.5) and/or an additional **timestamp** (chapter 5.1.5.4).

- **urn:ietf:3447 (http://ais.swisscom.ch/1.1/signaturetype/plain)**
  Creation of a static plain signature in PKCS#1 format (according to [RFC3447]).

In all cases, the response contains the signer's certificate as well as the full certificate chain up to the root certificate.

Example:

```
SOAP/XML:  <SignatureType>urn:ietf:rfc:3161</SignatureType>
JSON:      "SignatureType": "urn:ietf:rfc:3161"
```

5.1.5.2 Signature Standard

The optional element <SignatureStandard> defines the standard of the signature and can take the values "PAdES" or "CAdES". If not specified, the signature standard defaults to CAdES. The signing time is only included in the signature if the signature standard is CAdES.

For CMS signatures, if the type of the element <AddRevocationInfo> is not specified, the type of the revocation information will be selected based on the specified signature standard. For timestamps, the signature standard does not apply, so the revocation information type must be set explicitly.

5.1.5.3 Additional Profiles

The element <AdditionalProfile> enables the following profile enhancements.

mandatory:	the profile must be set in the request
optional:	the profile is supported and may be set
n/a:	the profile is not supported and must not be set

ADDITIONAL PROFILE	**URN:IETF:RFC:3161** **TRUSTED TIMESTAMP**	**URN:IETF:RFC:3169** **CMS SIGNATURE STATIC**	**URN:IETF:RFC:3369** **CMS SIGNATURE ON DEMAND**	**URN:IETF:RFC: 3447** **STATIC PLAIN PKCS#1**
urn:oasis:names:tc:dss:1.0:profiles:timestamping The signature response will only contain a timestamp of the document hash	**Mandatory**	N/A	N/A	N/A
urn:oasis:names:tc:dss:1.0:profiles:asynchronousprocessing The signature may be processed in asynchronous mode. The response may[4] be a **Pending**[5] result that contains the **ResponseID**. The client must be able to poll the result with a **PendingRequest** (see 5.4).	Optional	Optional	**Mandatory** with step-up auth., Optional otherwise	Optional
http://ais.swisscom.ch/1.1/profiles/redirect Required for step-up authentication. AIS receives an asynchronous call request. After receiving a "pending" response including a Consent URL, the client needs to redirect the user to this URL and to start polling the result with a PendingRequest.	N/A	N/A	Optional	N/A
http://ais.swisscom.ch/1.0/profiles/batchprocessing Required if a single signature request contains multiple document digests. Since you cannot rely on the order of the <DocumentHash> elements in the response, it's mandatory for a request to define a unique ID[6] attribute for each <DocumentHash>.	Optional	Optional	Optional	Optional
http://ais.swisscom.ch/1.0/profiles/ondemandcertificate Used to indicate the use of an On Demand certificate instead of a Static certificate. The signature request shall contain a subject **Distinguished Name** (see 5.1.5.6) and optionally a **Step-Up Authentication** (see 5.1.5.7)	N/A	N/A	**Mandatory**	N/A
http://ais.swisscom.ch/1.1/profiles/plainsignature Used to indicate the use of a static plain PKCS#1 signature.	N/A	N/A	N/A	**Mandatory**

[4] The term ‚may' is used because there is always the possibility of synchronous processing or an error response
[5] urn:oasis:names:tc:dss:1.0:profiles:asynchronousprocessing:resultmajor:Pending
[6] DocumentHash ID is of type xsd:ID - refer to http://books.xmlschemata.org/relaxng/ch19-77151.html

Example:

```
SOAP/XML:   <AdditionalProfile>
               urn:oasis:names:tc:dss:1.0:profiles:timestamping
            </AdditionalProfile>
JSON:       "AdditionalProfile": "urn:oasis:names:tc:dss:1.0:profiles:timestamping"
```

5.1.5.4 Add Timestamp

The element <AddTimestamp Type="urn:ietf:rfc:3161"/> may be added to the signature request to include a timestamp information according to [RFC3161] to the signature response.

ⓘ This option only applies to the **urn:ietf:rfc:3369** CMS Signature Type.

Example:

```
SOAP/XML:   <AddTimestamp Type="urn:ietf:rfc:3161"/>
JSON:       "AddTimestamp": {"@Type": "urn:ietf:rfc:3161"}
```

5.1.5.5 Add Revocation Information (long-term validation)

The element <AddRevocationInformation type="…"/> may be added to the signature request to include revocation information (RI) in the signature response. The attribute '**type**' supports the following values[7].

TYPE	DESCRIPTION	
Empty	For CMS signatures, if the attribute 'type' is not provided, the revocation information will match the defined SignatureStandard (CAdES or PAdES). (In case the SignatureStandard is not set, the default is CAdES). For timestamps, the signature standard does not apply so the revocation information type must be set explicitly.	
CAdES	RI will be **embedded** as an **unsigned** attribute with OID 1.2.840.113549.1.9.16.2.24	
PAdES	CMS Signatures:	RI will be **embedded** in the signature as a **signed** attribute with OID 1.2.840.113583.1.1.8
	Trusted Timestamps:	RI will be provided in the response as Base64 encoded OCSP responses or CRLs within the <OptionalOutputs>-Element
BOTH	Both RI types (CAdES,PAdES) will be provided	

AIS collects revocation information for every certificate in the signing certificate chain. It first tries to fetch an OCSP Response. If no OCSP Responder is available, it will try to fetch the CRL instead. An overview about the Swisscom CA Hierarchy and related OCSP or CRL can be found in chapter 6.5.

Example:

```
SOAP/XML:  <sc:AddRevocationInformation Type="BOTH"/>
JSON:      "sc.AddRevocationInformation": {"@Type": "BOTH"}
```

[7] The values are case insensitive

5.1.5.6 On Demand Distinguished Name

The *subject name* of a certificate is a distinguished name (DN) that contains identifying information about the entity to which the certificate is issued. For On Demand certificate signatures, you must provide a DN (of type string) in your request message.

The following table shows the AIS supported attributes.

You may use either the name or the abbreviation in the DN:

NAME	ABBREVIATION	DESCRIPTION	REQUIRED?[8]
CommonName	CN	Common Name	**Mandatory**
CountryName	C	Country Code (exactly two characters)	**Mandatory**
EmailAddress	-	E-Mail Address	Optional
GivenName	-	Given Name	Mandatory if no pseudonym is used
Surname	SN	Surname	Mandatory if no pseudonym is used
LocalityName	L	Locality	Optional
OrganizationalUnitName	OU	Organizational Unit (multiple fields are allowed)	Optional
OrganizationName	O	Organization	Optional
SerialNumber	-	Sequence of characters uniquely identifying the signatory.	Optional
StateOrProvinceName	ST	State or Province	Optional
Pseudonym		A name or key that uniquely identifies the user.	Optional

Example:

```
SOAP/XML:  <sc:DistinguishedName>cn=Jo
Muster,GivenName=Johannes,sn=Muster,o=ACME,c=CH</sc:DistinguishedName>
JSON:      "sc.CertificateRequest": {"sc.DistinguishedName": " Jo
Muster,GivenName=Johannes,sn=Muster,o=ACME,c=CH "}
```

This option only applies to the **urn:ietf:rfc:3369** CMS Signature Type.

In addition to the mandatory attributes defined above, the DN will be validated against the configured DN Pattern. All attributes defined in the pattern will be enforced by AIS. See 3.3.1 for more information regarding the DN filter.

[8] "Required" in this case denotes if the attribute is technically enforced by AIS or not. The regulatory requirements for building the DN as described in [CP] and [CPS] are mandatory and must be taken into consideration.

5.1.5.7 On Demand with Step-Up Authentication

Including an element <StepUpAuthorisation> in the signature request enforces step-up authentication for On Demand signatures. Although this element is optional, most customer certificate profiles require it. The parameter can only be omitted by special arrangement with Swisscom if it is replaced by a verified customer method.

The first method for step-up authorisation is Mobile ID. In case this is not supported by the user's mobile subscription (for example, for a non-Swiss number) an alternative method consisting of password and/or OTP verification will be invoked.

To invoke a Step-Up Authorisation, following data must be included within the <Phone> element.

ELEMENT	DESCRIPTION	REQUIRED?
<MSISDN>[9]	The Mobile phone number of the user. Example: `<sc:MSISDN>+491791234567</sc:MSISDN>`	**Mandatory**
<Message>[10]	The text displayed to the user. Example: `<sc:Message>Sign Contract.pdf? (#2Uk3Lv)</sc:Message>`	**Mandatory**
<Language>[10]	The Language of the <Message> content. Example: `<sc:Language>EN</sc:Language>`	**Mandatory**
<SerialNumber>	This number is a unique serial number (ID) associated to the user's phone number by the backend step-up service. If provided in the request, AIS will only compute and return a signature if it equals the one returned by the step-up service (see 5.8.2). Example: `<sc:SerialNumber>SAS01E0D9GAI7OO1</sc:SerialNumber>`	Optional

Example:

```
SOAP/XML:  <sc:StepUpAuthorisation>
              <sc:Phone>
                 <sc:MSISDN>491791234567</sc:MSISDN>
                 <sc:Message> Sign Contract.pdf? (#2Uk3Lv)</sc:Message>
                 <sc:Language>EN</sc:Language>
                  <sc:SerialNumber>SAS01E0D9GAI7OO1</sc:SerialNumber>
              </sc:Phone>
           </sc:StepUpAuthorisation>
JSON:      "sc.StepUpAuthorisation": {
              "sc.Phone": {
                 "sc.MSISDN": "491791234567",
                 "sc.Message": "Sign Contract.pdf? (#2Uk3Lv)",
                 "sc.Language": "EN",
                "sc.SerialNumber": "SAS01E0D9GAI7OO1"
              }
           }
```

⚠ Step-up authentication is only supported in asynchronous mode!

5.1.5.8 Claimed Identity

The customer (your company) must have an agreement with Swisscom in order to receive his personal <customer name> and <key entity> values.

The <ClaimedIdentity> element must have a value constructed as **<customer name>:<key entity>**

<key entity> is only required for static certificate and On Demand certificate signature requests. Refer to your customer specific certificate profile configuration for further details.

[9] The value must be compliant with the format defined in [MIDSOAP] chapter 5.1

All parts of the string are case sensitive.

Example:

```
SOAP/XML:  <ClaimedIdentity><Name>MyCustomer:MyKey</Name></ClaimedIdentity>
JSON:      "ClaimedIdentity": {"Name": "MyCustomer:MyKey"}
```

5.1.6 On Demand Certificate Policy and Certification Practice Statement (CP/CPS[10])

In the certificate creation process, the RSA key pair is generated on the HSM. The keys are always 2048 bit.

The AIS software creates a certificate signing request (CSR) according to [RFC2986]. The CSR contains the following extensions:

- Distinguished Name (DN) for the subject (see 5.1.5.6)
- Subject Alternative Name (SAN)
- Issuer Alternative Name (IAN)

All other fields and extensions are set by the CA.

Subject Alternative Name (SAN)
In case of success for On Demand signatures including step-up authentication, a DirectoryName entry in the SAN contains the related details:

- **Pseudonym** (OID 2.5.4.65)
  Contains the unique serial number linked to the user's mobile phone number in the Step-Up Authorisation System.
- **Description** (OID 2.5.4.13)
  May contain the Data To Be Signed (DTBS) string from the origin SignRequest
- **SerialNumber** (OID 2.5.4.5)
  Contains the transaction number which uniquely identifies the session established between AIS and the Step-up Authorisation System
- **Name** (OID 2.5.4.41)
  May contain the end user mobile phone number (MSISDN)

Issuer Alternative Name (IAN)
The IAN is used to enhance the certificate with the SignRequest details:

- **Description** (OID 2.5.4.13)
  Contains either the customer's name or the customer's unique ID.
- **SerialNumber** (OID 2.5.4.5)
  Contains the unique Request ID of the SignRequest

[10] Refer to [CP/CPS]

5.2 Trusted Timestamp

5.2.1 Trusted Timestamp SignRequest

Grey = SOAP specific; **Blue** = Optional batch processing; ***Italic*** = Optional but highly recommended

SOAP/XML Timestamp SignRequest

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
               xmlns:ais="http://service.ais.swisscom.com/">
   <soap:Body>
      <ais:sign>
         <SignRequest RequestID="YOUR_UNIQUE_ID"
                      Profile="http://ais.swisscom.ch/1.1"
                      xmlns="urn:oasis:names:tc:dss:1.0:core:schema"
                      xmlns:dsig="http://www.w3.org/2000/09/xmldsig#"
                      xmlns:sc="http://ais.swisscom.ch/1.0/schema">
            <OptionalInputs>
               <ClaimedIdentity>
                  <Name>YOUR_CUSTOMER_NAME</Name>
               </ClaimedIdentity>
               <SignatureType>urn:ietf:rfc:3161</SignatureType>
               <sc:AddRevocationInformation Type="BOTH"/>
               <AdditionalProfile>urn:oasis:names:tc:dss:1.0:profiles:timestamping</AdditionalProfile>
               <AdditionalProfile>http://ais.swisscom.ch/1.0/profiles/batchprocessing</AdditionalProfile>
            </OptionalInputs>
            <InputDocuments>
               <DocumentHash ID="YOUR_DOCID_#1">
                  <dsig:DigestMethod Algorithm="DIGEST_ALGO_#1"/>
                  <dsig:DigestValue>HASH_VALUE_#1</dsig:DigestValue>
               </DocumentHash>
               <DocumentHash ID="YOUR_DOCID_#2">
                  <dsig:DigestMethod Algorithm="DIGEST_ALGO_#2"/>
                  <dsig:DigestValue>HASH_VALUE_#2</dsig:DigestValue>
               </DocumentHash>
            </InputDocuments>
         </SignRequest>
      </ais:sign>
   </soap:Body>
</soap:Envelope>
```

JSON Timestamp SignRequest

```
{ "SignRequest": {
    "@RequestID": "YOUR_UNIQUE_ID",
    "@Profile": "http://ais.swisscom.ch/1.1",
    "OptionalInputs": {
      "ClaimedIdentity": {
        "Name": "YOUR_CUSTOMER_NAME"
      },
      "SignatureType": "urn:ietf:rfc:3161",
      "sc.AddRevocationInformation": {"@Type": "BOTH"},
      "AdditionalProfile": [
          "urn:oasis:names:tc:dss:1.0:profiles:timestamping",
          "http://ais.swisscom.ch/1.0/profiles/batchprocessing"
      ]
    },
    "InputDocuments": {
      "DocumentHash": [
       {
        "@ID": "YOUR_DOCID_#1",
        "dsig.DigestMethod": { "@Algorithm": "DIGEST_ALGO_#1" },
        "dsig.DigestValue": "HASH_VALUE_#1"
        },
       {
        "@ID": "YOUR_DOCID_#2",
        "dsig.DigestMethod": { "@Algorithm": "DIGEST_ALGO_#2" },
        "dsig.DigestValue": "HASH_VALUE_#2"
      }
      ]
    }
  }
}
```

5.2.2 Trusted Timestamp SignResponse

Grey = SOAP specific; **Blue** = Optional batch processing; ***Italic*** = Optional but highly recommended

SOAP/XML Timestamp SignResponse

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body>
      <ais:signResponse xmlns="urn:oasis:names:tc:dss:1.0:core:schema"
                        xmlns:dsig="http://www.w3.org/2000/09/xmldsig#"
                        xmlns:async="urn:oasis:names:tc:dss:1.0:profiles:asynchronousprocessing:1.0"
                        xmlns:ais="http://service.ais.swisscom.com/"
                        xmlns:sc="http://ais.swisscom.ch/1.0/schema"
                        xmlns:sas="http://sas.swisscom.ch/1.0/schema"
                        xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion">
         <SignResponse RequestID="YOUR_UNIQUE_ID"
                       Profile="http://ais.swisscom.ch/1.1"
                       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
            <Result>
               <ResultMajor>urn:oasis:names:tc:dss:1.0:resultmajor:Success</ResultMajor>
            </Result>
            <OptionalOutputs>
               <sc:RevocationInformation>
                  <sc:CRLs><sc:CRL>CRL_#1</sc:CRL></sc:CRLs>
                  <sc:OCSPs><sc:OCSP>OCSP_#1</sc:OCSP></sc:OCSPs>
               </sc:RevocationInformation>
            </OptionalOutputs>
            <SignatureObject>
               <Other>
                  <sc:SignatureObjects>
                     <sc:ExtendedSignatureObject WhichDocument="YOUR_DOCID_#1">
                        <Timestamp>
                           <RFC3161TimeStampToken>TIMESTAMP_TOKEN_#1</RFC3161TimeStampToken>
                        </Timestamp>
                     </sc:ExtendedSignatureObject>
                     <sc:ExtendedSignatureObject WhichDocument="YOUR_DOCID_#2">
                        <Timestamp>
                           <RFC3161TimeStampToken>TIMESTAMP_TOKEN_#2</RFC3161TimeStampToken>
                        </Timestamp>
                     </sc:ExtendedSignatureObject>
                  </sc:SignatureObjects>
               </Other>
            </SignatureObject>
         </SignResponse>
      </ais:signResponse>
   </soap:Body>
</soap:Envelope>
```

JSON Timestamp SignResponse

```
{ "SignResponse": {
    "@RequestID": "YOUR_UNIQUE_ID",
    "@Profile": "http://ais.swisscom.ch/1.1",
    "Result": {
      "ResultMajor": "urn:oasis:names:tc:dss:1.0:resultmajor:Success"
    },
    "OptionalOutputs": {
      "sc.RevocationInformation": {
        "sc.CRLs": { "sc.CRL": "CRL_#1" },
        "sc.OCSPs": { "sc.OCSP": "OCSP_#1" }
      }
    },
    "SignatureObject": {
      "Other": {
        "sc.SignatureObjects": {
          "sc.ExtendedSignatureObject": [
            {
              "@WhichDocument": "YOUR_DOCID_#1",
              "Timestamp": {
                "RFC3161TimeStampToken": "TIMESTAMP_TOKEN_#1"
              }
            },
            {
              "@WhichDocument": "YOUR_DOCID_#2",
              "Timestamp": {
                "RFC3161TimeStampToken": "TIMESTAMP_TOKEN_#2"
              }
            }
          ]
        }
      }
    }
  }
}
```

5.3 CMS Signatures

This chapter shows examples of signature requests and responses for static or On Demand signatures without additional step-up authentication.

5.3.1 CMS SignRequest for Static Signatures

Synchronous signature request for a static signature.

Grey = SOAP specific; **Blue** = Optional batch processing; ***Italic*** = Optional but highly recommended
SOAP/XML Static Certificate SignRequest

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
               xmlns:ais="http://service.ais.swisscom.com/">
   <soap:Body>
      <ais:sign>
         <SignRequest RequestID="YOUR_UNIQUE_ID"
                      Profile="http://ais.swisscom.ch/1.1"
                      xmlns="urn:oasis:names:tc:dss:1.0:core:schema"
                      xmlns:dsig="http://www.w3.org/2000/09/xmldsig#"
                      xmlns:sc="http://ais.swisscom.ch/1.0/schema">
            <OptionalInputs>
               <ClaimedIdentity>
                  <Name>YOUR_CUSTOMER_NAME:YOUR_KEY_ENTITY</Name>
               </ClaimedIdentity>
               <SignatureType>urn:ietf:rfc:3369</SignatureType>
               <AdditionalProfile>http://ais.swisscom.ch/1.0/profiles/batchprocessing</AdditionalProfile>
               <AddTimestamp Type="urn:ietf:rfc:3161"/>
               <sc:AddRevocationInformation Type="BOTH"/>
            </OptionalInputs>
            <InputDocuments>
               <DocumentHash ID="YOUR_DOCID_#1">
                  <dsig:DigestMethod Algorithm="DIGEST_ALGO"/>
                  <dsig:DigestValue>HASH_VALUE_#1</dsig:DigestValue>
               </DocumentHash>
               <DocumentHash ID="YOUR_DOCID_#2">
                  <dsig:DigestMethod Algorithm="DIGEST_ALGO"/>
                  <dsig:DigestValue>HASH_VALUE_#2</dsig:DigestValue>
               </DocumentHash>
            </InputDocuments>
         </SignRequest>
      </ais:sign>
   </soap:Body>
</soap:Envelope>
```

JSON Static Certificate SignRequest

```
{ "SignRequest": {
    "@RequestID": "YOUR_UNIQUE_ID",
    "@Profile": "http://ais.swisscom.ch/1.1",
    "OptionalInputs": {
      "ClaimedIdentity": {
        "Name": "YOUR_CUSTOMER_NAME:YOUR_KEY_ENTITY"
      },
      "SignatureType": "urn:ietf:rfc:3369",
      "AdditionalProfile": "http://ais.swisscom.ch/1.0/profiles/batchprocessing",
      "AddTimestamp": {"@Type": "urn:ietf:rfc:3161"},
      "sc.AddRevocationInformation": {"@Type": "BOTH"}
    },
    "InputDocuments": {
      "DocumentHash": [
      {
        "@ID": "YOUR_DOCID_#1",
        "dsig.DigestMethod": { "@Algorithm": "DIGEST_ALGO" },
        "dsig.DigestValue": "HASH_VALUE_#1"
      },
     {
        "@ID": "YOUR_DOCID_#2",
        "dsig.DigestMethod": { "@Algorithm": "DIGEST_ALGO" },
        "dsig.DigestValue": "HASH_VALUE_#2"
      }
     ]
    }
  }
}
```

5.3.2 CMS SignRequest for On Demand Signatures

Synchronous signature request for an On Demand signature without additional step-up authentication.

Grey = SOAP specific; **Blue** = Optional batch processing; ***Italic*** = Optional but highly recommended
Orange = Optional On Demand certificate request (instead of a static certificate)

SOAP/XML On Demand Certificate SignRequest

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
               xmlns:ais="http://service.ais.swisscom.com/">
   <soap:Body>
      <ais:sign>
         <SignRequest RequestID="YOUR_UNIQUE_ID"
                      Profile="http://ais.swisscom.ch/1.1"
                      xmlns="urn:oasis:names:tc:dss:1.0:core:schema"
                      xmlns:dsig="http://www.w3.org/2000/09/xmldsig#"
                      xmlns:sc="http://ais.swisscom.ch/1.0/schema">
            <OptionalInputs>
               <ClaimedIdentity>
                  <Name>YOUR_CUSTOMER_NAME:YOUR_KEY_ENTITY</Name>
               </ClaimedIdentity>
               <SignatureType>urn:ietf:rfc:3369</SignatureType>
               <AdditionalProfile>http://ais.swisscom.ch/1.0/profiles/batchprocessing</AdditionalProfile>
               <AdditionalProfile>http://ais.swisscom.ch/1.0/profiles/ondemandcertificate</AdditionalProfile>
               <sc:CertificateRequest>
                  <sc:DistinguishedName>YOUR_DN</sc:DistinguishedName>
               </sc:CertificateRequest>
               <AddTimestamp Type="urn:ietf:rfc:3161"/>
               <sc:AddRevocationInformation Type="BOTH"/>
            </OptionalInputs>
            <InputDocuments>
               <DocumentHash ID="YOUR_DOCID_#1">
                  <dsig:DigestMethod Algorithm="DIGEST_ALGO"/>
                  <dsig:DigestValue>HASH_VALUE_#1</dsig:DigestValue>
               </DocumentHash>
               <DocumentHash ID="YOUR_DOCID_#2">
                  <dsig:DigestMethod Algorithm="DIGEST_ALGO"/>
                  <dsig:DigestValue>HASH_VALUE_#2</dsig:DigestValue>
               </DocumentHash>
            </InputDocuments>
         </SignRequest>
      </ais:sign>
   </soap:Body>
</soap:Envelope>
```

JSON On Demand Certificate SignRequest

```
{ "SignRequest": {
    "@RequestID": "YOUR_UNIQUE_ID",
    "@Profile": "http://ais.swisscom.ch/1.1",
    "OptionalInputs": {
      "ClaimedIdentity": {
        "Name": "YOUR_CUSTOMER_NAME:YOUR_KEY_ENTITY"
      },
      "SignatureType": "urn:ietf:rfc:3369",
      "AdditionalProfile": [
          "http://ais.swisscom.ch/1.0/profiles/batchprocessing",
          "http://ais.swisscom.ch/1.0/profiles/ondemandcertificate"
      ],
      "sc.CertificateRequest": {
        "sc.DistinguishedName": "YOUR_DN",
      },
      "AddTimestamp": {"@Type": "urn:ietf:rfc:3161"},
      "sc.AddRevocationInformation": {"@Type": "BOTH"}
    },
    "InputDocuments": {
      "DocumentHash": [
      {
        "@ID": "YOUR_DOCID_#1",
        "dsig.DigestMethod": { "@Algorithm": "DIGEST_ALGO" },
        "dsig.DigestValue": "HASH_VALUE_#1"
      },
      {
        "@ID": "YOUR_DOCID_#2",
        "dsig.DigestMethod": { "@Algorithm": "DIGEST_ALGO" },
        "dsig.DigestValue": "HASH_VALUE_#2"
      }
     ]
    }
  }
}
```

5.3.3 CMS SignResponse

Signature response for synchronous signature requests for both static and On Demand signatures without additional step-up authentication (5.3.1 and 5.3.2).

Grey = SOAP specific; **Blue** = Optional batch processing; ***Italic*** = Optional but highly recommended

SOAP/XML Static/On Demand Certificate SignResponse

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body>
      <ais:signResponse xmlns="urn:oasis:names:tc:dss:1.0:core:schema"
                        xmlns:dsig="http://www.w3.org/2000/09/xmldsig#"
                        xmlns:ais="http://service.ais.swisscom.com/"
                        xmlns:sc="http://ais.swisscom.ch/1.0/schema"
                        xmlns:sas="http://sas.swisscom.ch/1.0/schema"
                        xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion">
         <SignResponse RequestID="YOUR_UNIQUE_ID"
                       Profile="http://ais.swisscom.ch/1.1"
                       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
            <Result>
               <ResultMajor>urn:oasis:names:tc:dss:1.0:resultmajor:Success</ResultMajor>
            </Result>
            <OptionalOutputs>
               <sc:RevocationInformation>
                  <sc:CRLs><sc:CRL>CRL_#1</sc:CRL></sc:CRLs>
                  <sc:OCSPs><sc:OCSP>OCSP_#1</sc:OCSP></sc:OCSPs>
               </sc:RevocationInformation>
            </OptionalOutputs>
            <SignatureObject>
               <Other>
                  <sc:SignatureObjects>
                     <sc:ExtendedSignatureObject WhichDocument="YOUR_DOCID_#1">
                        <Base64Signature Type="urn:ietf:rfc:3369">SIGNATURE_RESPONSE_#1</Base64Signature>
                     </sc:ExtendedSignatureObject>
                     <sc:ExtendedSignatureObject WhichDocument="YOUR_DOCID_#2">
                        <Base64Signature Type="urn:ietf:rfc:3369">SIGNATURE_RESPONSE_#2</Base64Signature>
                     </sc:ExtendedSignatureObject>
                  </sc:SignatureObjects>
               </Other>
            </SignatureObject>
         </SignResponse>
      </ais:signResponse>
   </soap:Body>
</soap:Envelope>
```

JSON Static/On Demand Certificate SignResponse

```
{ "SignResponse": {
    "@RequestID": "YOUR_UNIQUE_ID",
    "@Profile": "http://ais.swisscom.ch/1.1",
    "Result": {
      "ResultMajor": "urn:oasis:names:tc:dss:1.0:resultmajor:Success"
    },
    "OptionalOutputs": {
      "sc.RevocationInformation": {
        "sc.CRLs": { "sc.CRL": "CRL_#1" },
        "sc.OCSPs": { "sc.OCSP": "OCSP_#1" }
      }
    },
    "SignatureObject": {
      "Other": {
        "sc.SignatureObjects": {
          "sc.ExtendedSignatureObject": [
            {
              "@WhichDocument": "YOUR_DOCID_#1",
              "Base64Signature": {
                "@Type": "urn:ietf:rfc:3369",
                "$": "SIGNATURE_RESPONSE_#1"
              }
            },
            {
              "@WhichDocument": "YOUR_DOCID_#2",
              "Base64Signature": {
                "@Type": "urn:ietf:rfc:3369",
                "$": "SIGNATURE_RESPONSE_#2"
              }
            }
          ]
        }
      }
    }
  }
}
```

5.4 Asynchronous Mode

Asynchronous mode is supported for both static and On Demand signatures and mandatory for On Demand certificate signature requests where step-up authentication is enforced.

5.4.1 SignRequest

In order to indicate that the process should take place asynchronously, the following additional profile must be included in the request:

```
SOAP/XML:    <AdditionalProfile>
                urn:oasis:names:tc:dss:1.0:profiles:asynchronousprocessing
             </AdditionalProfile>
JSON:        "AdditionalProfile": "urn:oasis:names:tc:dss:1.0:profiles:asynchronousprocessing"
```

The asynchronous scenario (flow):

1. Send a "signRequest" with the aforementioned AdditionalProfile[11] to indicate support for asynchronous mode
2. Receive a "signResponse" (see 5.4.2) that contains the Response ID of the ongoing transaction
3. Poll the Response ID (from step 2) until you get a final response (signature or error)
    a. Send a "pendingRequest" (see 5.4.3) that contains the Response ID (from step 2)
    b. Receive a "pendingResponse" (5.4.4) and check the ResultMajor value
        i. The ResultMajor may indicate a pending[12] result (repeat step 3) or it may be the final result: success (=signature) or an error.

5.4.2 SignResponse

Grey = SOAP specific

SOAP/XML SignResponse (asynchronous)

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body>
      <ais:signResponse xmlns="urn:oasis:names:tc:dss:1.0:core:schema"
                        xmlns:dsig="http://www.w3.org/2000/09/xmldsig#"
                        xmlns:async="urn:oasis:names:tc:dss:1.0:profiles:asynchronousprocessing:1.0"
                        xmlns:ais="http://service.ais.swisscom.com/"
                        xmlns:sc="http://ais.swisscom.ch/1.0/schema"
                        xmlns:sas="http://sas.swisscom.ch/1.0/schema"
                        xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion">
         <SignResponse RequestID="YOUR_UNIQUE_ID"
                       Profile="http://ais.swisscom.ch/1.1"
                       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
            <Result>
               <ResultMajor>
                  urn:oasis:names:tc:dss:1.0:profiles:asynchronousprocessing:resultmajor:Pending
               </ResultMajor>
            </Result>
            <OptionalOutputs>
               <async:ResponseID>RESPONSE_ID</async:ResponseID>
            </OptionalOutputs>
         </SignResponse>
      </ais:signResponse>
   </soap:Body>
</soap:Envelope>
```

[11] urn:oasis:names:tc:dss:1.0:profiles:asynchronousprocessing
[12] urn:oasis:names:tc:dss:1.0:profiles:asynchronousprocessing:resultmajor:Pending

JSON SignResponse (asynchronous)

```
{ "SignResponse": {
    "@RequestID": "YOUR_UNIQUE_ID",
    "@Profile": "http://ais.swisscom.ch/1.1",
    "Result": {
      "ResultMajor": "urn:oasis:names:tc:dss:1.0:profiles:asynchronousprocessing:resultmajor:Pending"
    },
    "OptionalOutputs": {
      "async.ResponseID": "RESPONSE_ID"
    }
  }
}
```

5.4.3 PendingRequest

Grey = SOAP specific

SOAP/XML PendingRequest

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
               xmlns:ais="http://service.ais.swisscom.com/">
   <soap:Body>
      <ais:pending>
         <async:PendingRequest Profile="http://ais.swisscom.ch/1.1"
                               xmlns:async="urn:oasis:names:tc:dss:1.0:profiles:asynchronousprocessing:1.0"
                               xmlns="urn:oasis:names:tc:dss:1.0:core:schema" >
            <OptionalInputs>
               <ClaimedIdentity>
                  <Name>YOUR_CUSTOMER_NAME</Name>
               </ClaimedIdentity>
               <async:ResponseID>RESPONSE_ID</async:ResponseID>
            </OptionalInputs>
         </async:PendingRequest>
      </ais:pending>
   </soap:Body>
</soap:Envelope>
```

JSON PendingRequest

```
{ "async.PendingRequest": {
    "@Profile": "http://ais.swisscom.ch/1.1",
    "OptionalInputs": {
      "ClaimedIdentity": {
        "Name": "YOUR_CUSTOMER_NAME"
      },
      "async.ResponseID": "RESPONSE_ID"
    }
  }
}
```

5.4.4 PendingResponse

Grey = SOAP specific

SOAP/XML PendingResponse

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body>
      <ais:pendingResponse xmlns="urn:oasis:names:tc:dss:1.0:core:schema"
                        xmlns:dsig="http://www.w3.org/2000/09/xmldsig#"
                        xmlns:async="urn:oasis:names:tc:dss:1.0:profiles:asynchronousprocessing:1.0"
                        xmlns:ais="http://service.ais.swisscom.com/"
                        xmlns:sc="http://ais.swisscom.ch/1.0/schema"
                        xmlns:sas="http://sas.swisscom.ch/1.0/schema"
                        xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion">
         <SignResponse  RequestID="YOUR_UNIQUE_ID"
                       Profile="http://ais.swisscom.ch/1.1"
                       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
            <Result>
               <ResultMajor>
                  urn:oasis:names:tc:dss:1.0:profiles:asynchronousprocessing:resultmajor:Pending
               </ResultMajor>
            </Result>
            <OptionalOutputs>
               <async:ResponseID
                               xmlns:ns3="urn:oasis:names:tc:dss:1.0:profiles:asynchronousprocessing:1.0"
                               xmlns:ns2=http://www.w3.org/2000/09/xmldsig#
                               xmlns:ns4="http://ais.swisscom.ch/1.0/schema">
                  RESPONSE_ID
               </async:ResponseID>
            </OptionalOutputs>
         </SignResponse>
      </ais:signResponse>
   </soap:Body>
</soap:Envelope>
```

JSON PendingResponse

```
{ "SignResponse": {
    "@RequestID": "YOUR_UNIQUE_ID",
    "@Profile": "http://ais.swisscom.ch/1.1",
    "Result": {
      "ResultMajor": "urn:oasis:names:tc:dss:1.0:profiles:asynchronousprocessing:resultmajor:Pending"
    },
    "OptionalOutputs": {
      "async.ResponseID": "RESPONSE_ID"
    }
  }
}
```

5.5 CMS On Demand Signatures with Step-Up Authentication

The following request and response samples show the On Demand signature cases where a Step-Up method is defined.

5.5.1 SignRequest

Signature requests for On Demand signatures with step-up authentication must always be asynchronous.

Following elements must be included in the signature request in order to enforce the additional authorisation step:

- Since step-up authentication is only supported for asynchronous requests, the *asynchronous* additional profile must be included in the signature request (see additional profiles in 5.1.5.3)
- In order to allow the aforementioned client redirect, a *redirect* additional profile must be included in the Signing Request (see additional profiles in 5.1.5.3).
- The <StepUpAuthentication> element including the required information to perform the step-up authentication.

Grey = SOAP specific; **Blue** = Optional batch processing; ***Italic*** = Optional but highly recommended
Orange = Additonal Profiles and Step Up information

SOAP/XML On Demand Certificate SignRequest with Step-Up Authentication

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
               xmlns:ais="http://service.ais.swisscom.com/">
   <soap:Body>
      <ais:sign>
         <SignRequest RequestID="YOUR_UNIQUE_ID"
                      Profile="http://ais.swisscom.ch/1.1"
                      xmlns="urn:oasis:names:tc:dss:1.0:core:schema"
                      xmlns:dsig="http://www.w3.org/2000/09/xmldsig#"
                      xmlns:sc="http://ais.swisscom.ch/1.0/schema">
            <OptionalInputs>
               <ClaimedIdentity>
                  <Name>YOUR_CUSTOMER_NAME:YOUR_KEY_ENTITY</Name>
               </ClaimedIdentity>
               <SignatureType>urn:ietf:rfc:3369</SignatureType>
               <AdditionalProfile>http://ais.swisscom.ch/1.0/profiles/batchprocessing</AdditionalProfile>
               <AdditionalProfile>http://ais.swisscom.ch/1.0/profiles/ondemandcertificate</AdditionalProfile>
               <AdditionalProfile>urn:oasis:names:tc:dss:1.0:profiles:asynchronousprocessing</AdditionalProfile>
               <AdditionalProfile>http://ais.swisscom.ch/1.1/profiles/redirect</AdditionalProfile>
               <sc:CertificateRequest>
                  <sc:DistinguishedName>YOUR_DN</sc:DistinguishedName>
                  <sc:StepUpAuthorisation>
                     <sc:Phone
                        <sc:MSISDN>MOBILE_NUMBER</sc:MSISDN>
                        <sc:Message>MESSAGE</sc:Message>
                        <sc:Language>LANGUAGE_CODE</sc:Language>
                        <sc:SerialNumber>UNIQUE_SERIAL_NUMBER</sc:SerialNumber>
                     </sc:Phone>
                  </sc:StepUpAuthorisation>
               </sc:CertificateRequest>
               <AddTimestamp Type="urn:ietf:rfc:3161"/>
               <sc:AddRevocationInformation Type="BOTH"/>
            </OptionalInputs>
            <InputDocuments>
               <DocumentHash ID="YOUR_DOCID_#1">
                  <dsig:DigestMethod Algorithm="DIGEST_ALGO"/>
                  <dsig:DigestValue>HASH_VALUE_#1</dsig:DigestValue>
               </DocumentHash>
               <DocumentHash ID="YOUR_DOCID_#2">
                  <dsig:DigestMethod Algorithm="DIGEST_ALGO"/>
                  <dsig:DigestValue>HASH_VALUE_#2</dsig:DigestValue>
               </DocumentHash>
            </InputDocuments>
         </SignRequest>
      </ais:sign>
   </soap:Body>
</soap:Envelope>
```

JSON On Demand Certificate SignRequest with Step-Up Authentication

```
{ "SignRequest": {
    "@RequestID": "YOUR_UNIQUE_ID",
    "@Profile": "http://ais.swisscom.ch/1.1",
    "OptionalInputs": {
      "ClaimedIdentity": {
        "Name": "YOUR_CUSTOMER_NAME:YOUR_KEY_ENTITY"
      },
      "SignatureType": "urn:ietf:rfc:3369",
      "AdditionalProfile": [
          "http://ais.swisscom.ch/1.0/profiles/batchprocessing",
          "http://ais.swisscom.ch/1.0/profiles/ondemandcertificate",
          "urn:oasis:names:tc:dss:1.0:profiles:asynchronousprocessing",
          "http://ais.swisscom.ch/1.1/profiles/redirect"
      ],
      "sc.CertificateRequest": {
        "sc.DistinguishedName": "YOUR_DN",
        "sc.StepUpAuthorisation": {
          "sc.Phone": {
            "sc.MSISDN": "MOBILE_NUMBER",
            "sc.Message": "MESSAGE",
            "sc.Language": "LANGUAGE_CODE",
            "sc.SerialNumber": "UNIQUE_SERIAL_NUMBER"
            }
          }
        },
      "AddTimestamp": {"@Type": "urn:ietf:rfc:3161"},
      "sc.AddRevocationInformation": {"@Type": "BOTH"}
    },
    "InputDocuments": {
      "DocumentHash": [
      {
        "@ID": "YOUR_DOCID_#1",
        "dsig.DigestMethod": { "@Algorithm": "DIGEST_ALGO" },
        "dsig.DigestValue": "HASH_VALUE_#1"
      },
      {
        "@ID": "YOUR_DOCID_#2",
        "dsig.DigestMethod": { "@Algorithm": "DIGEST_ALGO" },
        "dsig.DigestValue": "HASH_VALUE_#2"
      }
     ]
    }
  }
}
```

5.5.2 SignResponse

If the enforced step-up authentication method requires redirection, the consent URL must be included among the “optional outputs” of the asynchronous signing response. Otherwise, the sign response does not differ from the one returned for On Demand asynchronous requests without step-up authentication (see 5.4.2).

After receiving the response, the application must make the following:

1) If the response contains a consent URL, redirect the user to the provided URL. The consent URL will be included in the response in 2 cases:
    a. either if the Step-Up is defined with Mobile ID and the MobileID-check returns an error (fallback case)
    b. or if the Step-Up is defined with Password & SMS-Challenge is enforced by configuration
2) Start polling the AIS for a final signature response, including in each poll request the provided transaction ID.

Grey = SOAP specific
Orange = Additional consent URL for user redirection for non-Mobile ID users

SOAP/XML Asynchronous SignResponse with Consent URL (pending)

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body>
      <ais:signResponse xmlns="urn:oasis:names:tc:dss:1.0:core:schema"
                        xmlns:dsig="http://www.w3.org/2000/09/xmldsig#"
                        xmlns:async="urn:oasis:names:tc:dss:1.0:profiles:asynchronousprocessing:1.0"
                        xmlns:ais="http://service.ais.swisscom.com/"
                        xmlns:sc="http://ais.swisscom.ch/1.0/schema"
                        xmlns:sas="http://sas.swisscom.ch/1.0/schema"
                        xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion">
         <SignResponse RequestID="YOUR_UNIQUE_ID"
                       Profile="http://ais.swisscom.ch/1.1"
                       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
            <Result>
               <ResultMajor>
                  urn:oasis:names:tc:dss:1.0:profiles:asynchronousprocessing:resultmajor:Pending
               </ResultMajor>
          </Result>
          <OptionalOutputs>
               <async:ResponseID>RESPONSE_ID</async:ResponseID>
               <sc:StepUpAuthorisationInfo>
                 <sc:Result>
                   <sc:ConsentURL>CONSENT_URL</sc:ConsentURL>
                 </sc:Result>
               </sc:StepUpAuthorisationInfo>
            </OptionalOutputs>
         </SignResponse>
      </ais:signResponse>
   </soap:Body>
</soap:Envelope>
```

JSON Asynchronous SignResponse with Consent URL (pending)

```
{ "SignResponse": {
    "@RequestID": "YOUR_UNIQUE_ID",
    "@Profile": "http://ais.swisscom.ch/1.1",
    "Result": {
      "ResultMajor": "urn:oasis:names:tc:dss:1.0:profiles:asynchronousprocessing:resultmajor:Pending"
        }
    "OptionalOutputs": {
      "async.ResponseID": "RESPONSE_ID"
      "sc.StepUpAuthorisationInfo": {
        "sc.Result": {
          "sc.ConsentURL":"CONSENT_URL"
        }
      }
    }
  }
}
```

5.5.3 PendingRequest

The pending request for On Demand signatures with step-up authentication does not differ from the ones of other asynchronous requests (see 5.4.3).

5.5.4 PendingResponse

Responses returned by AIS to PendingRequests before the signature has been computed will have the state "pending" as result major (see 5.4.4).

5.5.5 SignResponse (SUCCESS)

At soon as the signature is available, the next polling request from the application will return the final response containing the requested signature, assuming the step-up authentication succeeded. The response will include step-up information.

Grey = SOAP specific; **Blue** = Optional batch processing; ***Italic*** = Optional but highly recommended
Orange = Optional[13] information for step-up authentication

SOAP/XML Asynchronous SignResponse with Step-Up Information (SUCCESS)

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body>
      <ais:signResponse xmlns="urn:oasis:names:tc:dss:1.0:core:schema"
                        xmlns:dsig="http://www.w3.org/2000/09/xmldsig#"
                        xmlns:async="urn:oasis:names:tc:dss:1.0:profiles:asynchronousprocessing:1.0"
                        xmlns:ais="http://service.ais.swisscom.com/"
                        xmlns:sc="http://ais.swisscom.ch/1.0/schema"
                        xmlns:sas="http://sas.swisscom.ch/1.0/schema"
                        xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion">
         <SignResponse RequestID="YOUR_UNIQUE_ID"
                       Profile="http://ais.swisscom.ch/1.1"
                       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
            <Result>
               <ResultMajor>urn:oasis:names:tc:dss:1.0:resultmajor:Success</ResultMajor>
            </Result>
            <OptionalOutputs>
               <sc:APTransID>AP_TRANSACTION_ID</sc:APTransID>
               <sc:StepUpAuthorisationInfo>
                 <sc:Result>
                   <sc:SerialNumber>UNIQUE_SERIAL_NUMBER</sc:SerialNumber>
                 </sc:Result>
               </sc:StepUpAuthorisationInfo>
               <sc:RevocationInformation>
                  <sc:CRLs><sc:CRL>CRL_#1</sc:CRL></sc:CRLs>
                  <sc:OCSPs><sc:OCSP>OCSP_#1</sc:OCSP></sc:OCSPs>
               </sc:RevocationInformation>
            </OptionalOutputs>
            <SignatureObject>
               <Other>
                  <sc:SignatureObjects>
                     <sc:ExtendedSignatureObject WhichDocument="YOUR_DOCID_#1">
                        <Base64Signature Type="urn:ietf:rfc:3369">SIGNATURE_RESPONSE_#1</Base64Signature>
                     </sc:ExtendedSignatureObject>
                     <sc:ExtendedSignatureObject WhichDocument="YOUR_DOCID_#2">
                        <Base64Signature Type="urn:ietf:rfc:3369">SIGNATURE_RESPONSE_#2</Base64Signature>
                     </sc:ExtendedSignatureObject>
                  </sc:SignatureObjects>
               </Other>
            </SignatureObject>
         </SignResponse>
      </ais:signResponse>
   </soap:Body>
</soap:Envelope>
```

[13] Optional - but the customer specific certificate profile may require step up authorisation

JSON On Demand asynchronous SignResponse with Step-Up Information (SUCCESS)

```
{ "SignResponse": {
    "@RequestID": "YOUR_UNIQUE_ID",
    "@Profile": "http://ais.swisscom.ch/1.1",
    "Result": {
      "ResultMajor": "urn:oasis:names:tc:dss:1.0:resultmajor:Success"
    },
    "OptionalOutputs": {
      "sc.APTransID ": "AP_TRANSACTION_ID",
      "sc.StepUpAuthorisationInfo":{
        "sc.Result":{
          "sc.SerialNumber":"UNIQUE_SERIAL_NUMBER"
        }
      }
      "sc.RevocationInformation": {
        "sc.CRLs": { "sc.CRL": "CRL_#1" },
        "sc.OCSPs": { "sc.OCSP": "OCSP_#1" }
      }
    },
    "SignatureObject": {
      "Other": {
        "sc.SignatureObjects": {
          "sc.ExtendedSignatureObject": [
            {
              "@WhichDocument": "YOUR_DOCID_#1",
              "Base64Signature": {
                "@Type": "urn:ietf:rfc:3369",
                "$": "SIGNATURE_RESPONSE_#1"
              }
            },
            {
              "@WhichDocument": "YOUR_DOCID_#2",
              "Base64Signature": {
                "@Type": "urn:ietf:rfc:3369",
                "$": "SIGNATURE_RESPONSE_#2"
              }
            }
          ]
        }
      }
    }
  }
}
```

5.6 Static Plain Signatures (PKCS#1)

Following 2 example requests for static plain signatures in PKCS#1 format.

Grey = SOAP specific; **Blue** = Optional batch processing; ***Italic*** = Optional but highly recommended
Orange = Additonal Profiles and Step Up information

SOAP Static Plain PKCS#1 Certificate SignRequest

```
<soap:Envelope
    xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
    xmlns:ais="http://service.ais.swisscom.com/">
   <soap:Body>
      <ais:sign>
        <SignRequest RequestID="2017-04-13T13:50:25.655+0200" Profile="http://ais.swisscom.ch/1.1"
xmlns="urn:oasis:names:tc:dss:1.0:core:schema" xmlns:dsig="http://www.w3.org/2000/09/xmldsig#"
xmlns:sc="http://ais.swisscom.ch/1.0/schema">
           <OptionalInputs>
              <AdditionalProfile>http://ais.swisscom.ch/1.1/profiles/plainsignature</AdditionalProfile>
              <SignatureType>http://ais.swisscom.ch/1.1/signaturetype/plain</SignatureType>
              <ClaimedIdentity>
                 <Name>IAM-Test:kp1-iam-signer</Name>
              </ClaimedIdentity>
              <AddTimestamp Type="urn:ietf:rfc:3161"/>
              <sc:AddRevocationInformation Type="PLAIN"/>
           </OptionalInputs>
           <InputDocuments>
              <DocumentHash>
                 <dsig:DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/>
                 <dsig:DigestValue>BwcHBwcHBwcHBwcHBwcHBwcHBwcHBwcHBwcHBwcHBwc=</dsig:DigestValue>
              </DocumentHash>
           </InputDocuments>
        </SignRequest>
      </ais:sign>
    </soap:Body>
</soap:Envelope>
```

JSON Static Plain PKCS#1 Certificate SignRequest

```
{
  "SignRequest": {
    "@RequestID": "2017-04-13T13:50:28.030+0200",
    "@Profile": "http://ais.swisscom.ch/1.1",
    "OptionalInputs": {
      "ClaimedIdentity": {
        "Name": "IAM-Test:kp1-iam-signer"
      },
      "AdditionalProfile": "http://ais.swisscom.ch/1.1/profiles/plainsignature",
      "SignatureType": "http://ais.swisscom.ch/1.1/signaturetype/plain",
      "AddTimestamp": {"@Type": "urn:ietf:rfc:3161"},
      "sc.AddRevocationInformation": {"@Type": "PLAIN"}
    },
    "InputDocuments": {
      "DocumentHash": {
        "@ID": "256",
        "dsig.DigestMethod": { "@Algorithm": "http://www.w3.org/2001/04/xmlenc#sha256" },
        "dsig.DigestValue": "BwcHBwcHBwcHBwcHBwcHBwcHBwcHBwcHBwcHBwcHBwc="
      }
    }
  }
}
```

5.7 Fault Response Message

In case of an error a Fault SignResponse will be raised with the appropriate details. The details will contain a ResultMajor, ResultMinor and a ResultMessage. The list of codes can be found in chapter 6.4.

5.7.1 Wrong Digest Size (example)

It follows an example SignResponse in case the SignRequest content was invalid due to a wrong digest size. The SignResponse has the related error details embedded.

Grey = SOAP specific

SOAP/XML Fault SignResponse

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body>
      <ais:signResponse xmlns="urn:oasis:names:tc:dss:1.0:core:schema"
                        xmlns:dsig="http://www.w3.org/2000/09/xmldsig#"
                        xmlns:async="urn:oasis:names:tc:dss:1.0:profiles:asynchronousprocessing:1.0"
                        xmlns:ais="http://service.ais.swisscom.com/"
                        xmlns:sc="http://ais.swisscom.ch/1.0/schema"
                        xmlns:sas="http://sas.swisscom.ch/1.0/schema"
                        xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion">
         <SignResponse RequestID="YOUR_UNIQUE_ID"
                       Profile="http://ais.swisscom.ch/1.1"
                       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
            <Result>
               <ResultMajor>urn:oasis:names:tc:dss:1.0:resultmajor:RequesterError</ResultMajor>
               <ResultMinor>http://ais.swisscom.ch/1.0/resultminor/UnexpectedData</ResultMinor>
               <ResultMessage xml:lang="en">digest size doesn't match the expected size</ResultMessage>
            </Result>
         </SignResponse>
      </ais:signResponse>
   </soap:Body>
</soap:Envelope>
```

JSON Fault SignResponse

```
{ "SignResponse": {
    "@RequestID": "YOUR_UNIQUE_ID",
    "@Profile": "http://ais.swisscom.ch/1.1",
    "Result": {
      "ResultMajor": "urn:oasis:names:tc:dss:1.0:resultmajor:RequesterError",
      "ResultMinor": "http://ais.swisscom.ch/1.0/resultminor/UnexpectedData",
      "ResultMessage": {
        "@xml.lang": "en",
        "$": "digest size doesn't match the expected size"
      }
    }
  }
}
```

5.7.2 Step-Up Authentication: Mobile ID User Account Problem (example)

It follows an example SignResponse in case the user does not have an active Mobile ID account and when MobileID-StepUp is used without fallback to Password and SMS-Challenge.
The SignResponse contains a **Portal URL**[14] that the user should visit in order to resolve the problem.

Grey = SOAP specific; **Green** = Mobile ID Step Up authorisation specific

SOAP/XML Fault SignResponse

```
..
      <ais:signResponse xmlns="urn:oasis:names:tc:dss:1.0:core:schema"
                        xmlns:dsig="http://www.w3.org/2000/09/xmldsig#"
                        xmlns:async="urn:oasis:names:tc:dss:1.0:profiles:asynchronousprocessing:1.0"
                        xmlns:ais="http://service.ais.swisscom.com/"
                        xmlns:sc="http://ais.swisscom.ch/1.0/schema"
                        xmlns:sas="http://sas.swisscom.ch/1.0/schema"
                        xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion">
```

[14] Refer to [MIDSOAP] chapter 6.5 (Fault Code User Assistance)

```
          <SignResponse RequestID="YOUR_UNIQUE_ID"
                        Profile="http://ais.swisscom.ch/1.1"
                        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
             <Result>
                <ResultMajor>http://ais.swisscom.ch/1.0/resultmajor/SubsystemError</ResultMajor>
                <ResultMinor>http://ais.swisscom.ch/1.1/resultminor/subsystem/StepUp/service</ResultMinor>
                <ResultMessage xml:lang="en">mss:_404</ResultMessage>
             </Result>
             <OptionalOutputs>
               <sc:StepUpAuthorisationInfo>
                 <sc:Result>
                   <sc:MobileIDFault>
                     <sc:Subcode xmlns:mss="http://uri.etsi.org/TS102204/v1.1.2#">mss:_404</sc:Subcode>
                     <sc:Reason>NO_KEY_FOUND</sc:Reason>
                     <sc:Detail>
                        <ns1:detail xmlns:ns1="http://kiuru.methics.fi/mssp">
                           Mobile user account needs to be activated
                        </ns1:detail>
                        <ns2:UserAssistance xmlns:ns2="http://www.swisscom.ch/TS102204/ext/v1.0.0">
                           <ns2:PortalUrl xmlns="http://www.swisscom.ch/TS102204/ext/v1.0.0">
                              https://.../MobileId?reactivateMobileId=true&amp;msisdn=41791234567
                           </ns2:PortalUrl>
                        </ns2:UserAssistance>
                     </sc:Detail>
                  </sc:MobileIDFault>
                <sc:Result>
               <sc:StepUpAuthorisationInfo>
             </OptionalOutputs>
          </SignResponse>
       </ais:signResponse>
..
```

JSON Fault SignResponse

```
{ "SignResponse": {
    "@RequestID": "YOUR_UNIQUE_ID",
    "@Profile": "http://ais.swisscom.ch/1.1",
    "Result": {
      "ResultMajor": "http://ais.swisscom.ch/1.0/resultmajor/SubsystemError",
      "ResultMinor": "http://ais.swisscom.ch/1.1/resultminor/subsystem/StepUp/service",
      "ResultMessage": {
        "@xml.lang": "en",
        "$": "mss:_404"
      }
    },
    "OptionalOutputs": {
      "sc.StepUpAuthorisationInfo": {
        "sc.Result": {
          "sc.MobileIDFault": {
          "sc.Subcode": "mss:_404",
          "sc.Reason": "NO_KEY_FOUND",
          "sc.Detail": {
            "ns1.detail": "Mobile user account needs to be activated",
            "ns2.UserAssistance": {
              "ns2.PortalUrl": "https://.../MobileId?reactivateMobileId=true&msisdn=41791234567"
              }
            }
          }
        }
    }
  }
}
```

5.7.3 Step-Up Authentication: User Cancel (example)

During a step-up authentication, the user presses cancel interrupting in this way the consent process.

Grey = SOAP specific

SOAP/XML Fault SignResponse

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body>
      <ais:signResponse xmlns="urn:oasis:names:tc:dss:1.0:core:schema"
                        xmlns:dsig="http://www.w3.org/2000/09/xmldsig#"
                        xmlns:async="urn:oasis:names:tc:dss:1.0:profiles:asynchronousprocessing:1.0"
                        xmlns:ais="http://service.ais.swisscom.com/"
                        xmlns:sc="http://ais.swisscom.ch/1.0/schema"
                        xmlns:sas="http://sas.swisscom.ch/1.0/schema"
                        xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion">
         <SignResponse RequestID="YOUR_UNIQUE_ID"
                       Profile="http://ais.swisscom.ch/1.1"
                       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
            <Result>
               <ResultMajor>http://ais.swisscom.ch/1.0/resultmajor/SubsystemError</ResultMajor>
               <ResultMinor>
                  http://ais.swisscom.ch/1.1/resultminor/subsystem/StepUp/cancel
               </ResultMinor>
               <ResultMessage/>
            </Result>f
         </SignResponse>
      </ais:signResponse>
   </soap:Body>
</soap:Envelope>
```

JSON Fault SignResponse

```
{ "SignResponse": {
    "@RequestID": "YOUR_UNIQUE_ID",
    "@Profile": "http://ais.swisscom.ch/1.1",
    "Result": {
      "ResultMajor": "http://ais.swisscom.ch/1.0/resultmajor/SubsystemError",
      "ResultMinor": "http://ais.swisscom.ch/1.1/resultminor/subsystem/StepUp/cancel",
      "ResultMessage": {
        "@xml.lang": "en",
        "$":
      }
    }
  }
}
```

5.7.4 Step-Up Authentication: SerialNumber Mismatch (example)

It follows an example of the case where the serial number included in the signature request does not match the serial number registered to the provided mobile phone number. (see 5.8.2).

Grey = SOAP specific

SOAP/XML Fault SignResponse

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body>
      <ais:signResponse xmlns="urn:oasis:names:tc:dss:1.0:core:schema"
                        xmlns:dsig="http://www.w3.org/2000/09/xmldsig#"
                        xmlns:async="urn:oasis:names:tc:dss:1.0:profiles:asynchronousprocessing:1.0"
                        xmlns:ais="http://service.ais.swisscom.com/"
                        xmlns:sc="http://ais.swisscom.ch/1.0/schema"
                        xmlns:sas="http://sas.swisscom.ch/1.0/schema"
                        xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion">
         <SignResponse RequestID="YOUR_UNIQUE_ID"
                       Profile="http://ais.swisscom.ch/1.1"
                       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
            <Result>
               <ResultMajor>http://ais.swisscom.ch/1.0/resultmajor/SubsystemError</ResultMajor>
               <ResultMinor>
                  http://ais.swisscom.ch/1.1/resultminor/subsystem/StepUp/SerialNumberMismatch
               </ResultMinor>
               <ResultMessage xml:lang="en">
                  SerialNumber mismatch. We strongly advise to go through the Pre-Signing Process in order
                  to retrieve the actual SerialNumber
               </ResultMessage>
            </Result>
         </SignResponse>
      </ais:signResponse>
   </soap:Body>
</soap:Envelope>
```

JSON Fault SignResponse

```
{ "SignResponse": {
    "@RequestID": "YOUR_UNIQUE_ID",
    "@Profile": "http://ais.swisscom.ch/1.1",
    "Result": {
      "ResultMajor": "http://ais.swisscom.ch/1.0/resultmajor/SubsystemError",
      "ResultMinor": "http://ais.swisscom.ch/1.1/resultminor/subsystem/StepUp/SerialNumberMismatch",
      "ResultMessage": {
        "@xml.lang": "en",
        "$": "SerialNumber mismatch. We strongly advise to go through the Pre-Signing Process in order to
              retrieve the actual SerialNumber"
      }
    }
  }
}
```

5.8 Best Practices

5.8.1 Example of Codes and Scripts

Swisscom has published a set of tools and scripts on the AIS repository on GitHub at http://git.io/nJ6BeA

The GitHub repository sub-content:

- **shell** - A set of shell scripts that use the AIS SOAP and RESTful interface
- **services** - Schemas and WSDL/WADL service description files
- **soapui** - A sample project for SoapUI that contains example of requests and a test suite

Additional repositories are available on http://git.io/Fcp7 in relation to AIS:

- **itext-ais** - Java source code and command line tool to digitally sign PDF with iText
- **setapdfsigner-ais** - SetaPDF-Signer Module Interface to digitally sign PDF with PHP

5.8.2 On Demand Step-Up Pre-Signing Process

Step-up authentication is mandatory for qualified signatures and most advanced signatures. Every On Demand Signature with step-up authentication of will contains a unique serial number in the signature response.

The Serial Number is a unique attribute associated to a user's mobile phone number. The Application Provider (AP) can rely to have the same physical body behind the step-up authentication as long as the serial number remains the same.

5.8.2.1 Pre-Signing Process

If an AP would like to track the Serial Number, the Pre-Signing process could look like as follows:

1. A user is authenticated to a portal or application of the AP
2. The AP displays an "Enrolment Document" that the user has to sign and whose signature must be approved through step-up authentication.
3. The user invokes the signature and the AP sends an AIS On Demand Signature with Step Up authentication where there is no Serial Number is provided in the Request.
4. The user confirms the Signing request on the corresponding device
5. The AP receives the AIS On Demand Signature Response that contains the serial number
6. The AP stores the serial number and uniquely binds it to the user's identity.

Once the Pre-Signing process has been completed successfully, the AP should always set the known Serial Number in every Step Up authentication as described in the chapter below (Signing Process).

5.8.2.2 Signing Process

1. The AP provides the Serial Number in the Step Up authentication
2. After a successful Step Up authentication, AIS checks whether the Serial Number provided by the AP and the one being returned by the Step-up Authentication System are equal.
    - In case they match the signature process proceeds as usual

- In case they mismatch the signature process is aborted and a fault response is returned (see 5.7.2). It is strongly advised to go through the Pre-Signing process again in order to retrieve the correct (actual) Serial Number.

5.8.3 Signing Requests

5.8.3.1 General

Those major aspects need to be considered when any signing request is being constructed:

1) Define a unique RequestID (type string) attribute for every <SignRequest> element. Note that from the DSS Standard it is optional – but we highly recommend to always define it.
2) On Demand certificate signature requests will take more time compared to static certificate requests because it takes time to generate a key pair for each request.
3) When signing multiple hashes consider to use the batch mode. That because the AIS authorisation (which may include the optional step up authorisation) is applied only once per request.

5.8.3.2 Step Up Authorisation

1) Refer to [MIDSOAP] chapter 5.1 to properly define the Step Up content.
2) Since a step up can take significant longer time due to the user interaction, you may consider handling the AIS signature requests in asynchronous mode. In case of synchronous mode, ensure that the client (AP) is waiting long enough for the response.
3) If step-up authentication must be enforced, asynchronous mode is the only option available.
4) The Serial Number is an optional element to increase the security of the authorisation. In case of a SerialNumber mismatch fault response, we strongly advise you go through the pre-signing process again (see 5.8.2.1).

5.8.4 Response handlings

It is under the responsibility of the AP to verify the response of the All-in Signing Service (AIS). The following points are crucial:

1. Verify that the RequestID attribute of the response message matches the RequestID attribute of the request message
2. Verify the signature (see 0).
3. Ensure proper exception handling for error response messages (see 6.4).
4. If step-up authentication is enforced, ensure proper handling of status and fault codes as described in [MIDSOAP], chapter 6.

5.8.5 Adobe PDF

Please refer also to the iText and PHP code examples on GitHub at http://git.io/Fcp7

5.8.5.1 Validation with On Demand Certificates

Usually a PDF Signature is based on three main steps:

1) A new PDF document is created and the signing time (local time) is set.
2) A signature is requested to the All-in Signing Service.
3) The signature is embedded in the new PDF document which has been created in step 1)

For a successful signature validation, it is important that you either have a trusted timestamp information in the signature (see 5.1.5.4) or the signing time set within the 10 Minutes validity period of the On Demand certificate (see 2.2).

For the latter case we recommend to add +3 Minutes to the local time when setting the signing time in step 1).

In order to have LTV-enabled signatures, the CMS signature must include the timestamp and the revocation information. You also need to add to the PDF document the timestamp revocation information, which for the PAdES Signature Standard is delivered separately in the OptionalOutputs element of the SignResponse (see 5.1.5.4).

5.8.5.2 Estimating the size of the signature content

In some cases, you may need to estimate the size of the signature content, i.e. when you want to embed a digital signature into a PDF document.

With the All-in Signing Service, it is relatively easy to make an educated guess. The size of the document to be signed has no impact on the signature size. Only the following request specific options have an impact on the size of the signature content.

<u>Options that have a fixed length:</u>

- Length of the customer name
- Digest Algorithm (the length of the hash value)
- Additional signing options such as Revocation Information or Timestamp

<u>Options that have a variable length:</u>

- Use and length of the Distinguished Name (DN)
- Use and length of the Step-Up message

We recommend to send a few example Signature Requests with your preferred options first, in order to get the actual size of the signature content. This should give you a good indication for the estimation of the signature size.

Please take into consideration that the size of the signature might change due to different factors which are not always easy to foresee. Make sure to let some margin when estimating the signature size.

6 Appendix

6.1 Create self-signed certificate with openssl

Below are some examples on how to create a self-signed certificate with OpenSSL, valid for 5 years.

6.1.1 Generate Key and CSR

```
$ openssl req -new -newkey rsa:2048 -nodes -rand /dev/urandom -keyout mycert.key
  -out mycert.csr -sha256 -subj '/CN=ais.company.ch/O=Company/C=CH'
```

6.1.2 Self-sign it and create your certificate

```
$ openssl x509 -req -days 1825 -sha256 -in mycert.csr -signkey mycert.key -out mycert.crt
```

6.1.3 Convert into PKCS#12 (if needed)

If you need a PKCS#12 file you can convert the Key and Certificate with this command:

```
$ openssl pkcs12 -export -in mycert.crt -inkey mycert.key -out mycert.p12
```

Provide the **mycert.crt** file to Swisscom and keep your *.key file securely stored.

6.2 Create self-signed certificate with Java keytool

Below are some examples on how to create a self-signed certificate with the Java Keytool, valid for 5 years.

6.2.1 Generate KeyStore & export the self-signed certificate

```
$ keytool -genkey -alias <alias-name> -keyalg RSA -keysize 2048 -validity 1825
  -dname 'CN=ais.company.ch,O=Company,C=CH' -keystore mycert.jks

$ keytool -export -alias <alias-name> -keystore mycert.jks -file mycert.crt
```

6.2.2 Root CA and Intermediate CA certificate import

```
$ keytool -keystore truststore.jks -import -file Swisscom_Root_CA_2_der.crt
$ keytool -keystore truststore.jks -import -file Swisscom_Rubin_CA_2_der.crt
```

6.2.3 Verification

```
$ keytool -printcert -v -file mycert.crt
$ keytool -list -v -keystore keystore.jks
$ keytool -list -v -keystore keystore.jks -alias <alias-name>
```

Provide the **mycert.crt** file to Swisscom and keep your *.jks file securely stored.

6.3 Swisscom signed certificate

Any client certificate issued by an official CA like Swisscom SDCS http://www.swissdigicert.ch may be used as an alternative to the self-signed certificate.

An example of a Swisscom CA Certificate file for the AIS service can be found here http://git.io/OUL3HA

The Swisscom CA Certificates can be downloaded from the Swisscom SDCS website:
http://www.swissdigicert.ch/sdcs/portal/page?node=download_ca

6.4 Result Major and ResultMinor list

The list of ResultMajor and ResultMinor status codes that might appear. Additionally, details will be set in the ResultMessage. Example:

```
<Result>
  <ResultMajor>http://ais.swisscom.ch/1.0/resultmajor/SubsystemError</ResultMajor>
  <ResultMinor>http://ais.swisscom.ch/1.1/resultminor/subsystem/StepUp/service</ResultMinor>
  <ResultMessage xml:lang="en">mss:_401</ResultMessage>
</Result>
```

This list may not be complete and additional status codes may be returned by the platform.

6.4.1 ResultMajor Status Codes

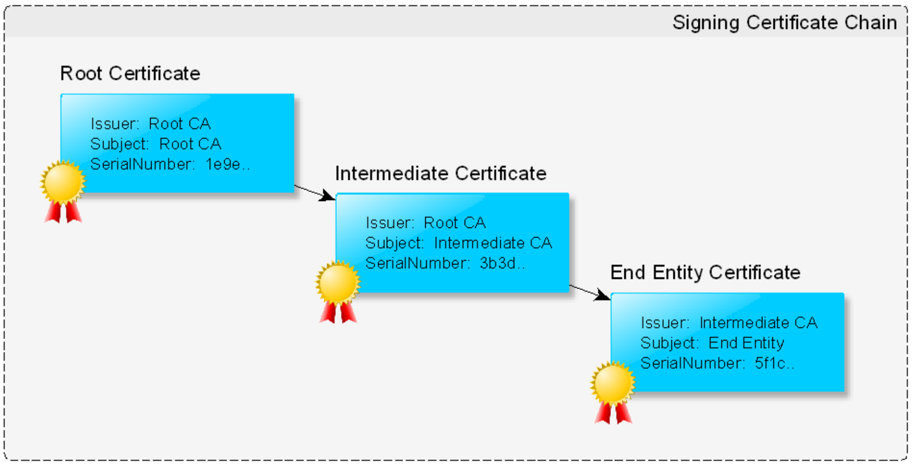
URN / URI	Description
http://ais.swisscom.ch/1.0/resultmajor/SubsystemError	Some subsystem of the server produced an error. Details are included in the minor status code.
urn:oasis:names:tc:dss:1.0:profiles:asynchronousprocessing: resultmajor:Pending	Asynchronous request was accepted. It is pending now.
urn:oasis:names:tc:dss:1.0:resultmajor:RequesterError	It is assumed that the caller has made a mistake. Details are included in the minor status code.
urn:oasis:names:tc:dss:1.0:resultmajor:ResponderError	The server could not process the request. Details are included in the minor status code.
urn:oasis:names:tc:dss:1.0:resultmajor:Success	Request was successfully executed

6.4.2 ResultMinor Status Codes

URN / URI	Description
http://ais.swisscom.ch/1.0/resultminor/AuthenticationFailed	Request authentication failed. For example, the customer used an unknown certificate.
http://ais.swisscom.ch/1.0/resultminor/CantServeTimely	The request could not be processed on time. The subsystem might be overloaded.
http://ais.swisscom.ch/1.0/resultminor/InsufficientData	The request could not be completed, because some information is missing.
http://ais.swisscom.ch/1.0/resultminor/ServiceInactive	The requested service is inactive (or not defined at all).
http://ais.swisscom.ch/1.0/resultminor/SignatureError	An error occurred, while creating a signature.
http://ais.swisscom.ch/1.1/resultminor/subsystem/StepUp/ SerialNumberMismatch	During a step-up authentication, the optional unique serial number was provided in the request but did not match the one of the user's mobile number.
http://ais.swisscom.ch/1.1/resultminor/subsystem/StepUp/service	A service error occurred during the step-up authentication. Error details are included in the error message.
http://ais.swisscom.ch/1.1/resultminor/subsystem/StepUp/status	An unknown status code of the step-up subsystem was included in the fault response.
http://ais.swisscom.ch/1.1/resultminor/subsystem/StepUp/timeout	The transaction expired before the step-up authorization was completed.
http://ais.swisscom.ch/1.1/resultminor/subsystem/StepUp/cancel	The user canceled the step-up authorization.
http://ais.swisscom.ch/1.0/resultminor/TimestampError	An error occurred, while creating a timestamp.
http://ais.swisscom.ch/1.0/resultminor/UnexpectedData	The request contains unexpected (wrong or misleading) data.
http://ais.swisscom.ch/1.0/resultminor/UnknownCustomer	The customer is unknown.
http://ais.swisscom.ch/1.0/resultminor/UnknownServiceEntity	The service entity (static key pair or the On Demand CA server) could not found. Maybe the customer does not have access to it.
http://ais.swisscom.ch/1.0/resultminor/UnsupportedDigestAlgorithm	The request contains a document hashed with unsupported or weak digest algorithms.
http://ais.swisscom.ch/1.0/resultminor/UnsupportedProfile	The request contained unknown profile URI.
http://ais.swisscom.ch/1.1/resultminor:subsystem/StepUp/transport	A subsystem transport error occurred.
urn:oasis:names:tc:dss:1.0:resultminor:GeneralError	A general internal error occurred.

6.5 Swisscom CA Hierarchy

The certificate chain provides a way to verify that all certificates related to the certificate being validated are trustworthy. A certificate-validation software walks through the signing certificate's chain starting with the end entity (EE) certificate, through the intermediate CA (ICA) certificate, until it finds a trusted Root CA (RCA) certificate. The Root CA certificate is the trust anchor.



In the sub chapter below, you will find an overview about the certificate hierarchy related to the Swisscom All-in Signing Service and its available revocation information (see chapter 5.1.5.5).

The Swisscom Root CA 2 certificate is included in the Adobe Approved Trust List (AATL)[15]

NAME	FINGERPRINT ALGORITHM	FINGERPRINT
Swisscom Root CA 2	SHA1	77 47 4f c6 30 e4 0f 4c 47 64 3f 84 ba b8 c6 95 4a 8a 41 ec

6.5.1 CMS Signature

CERTIFICATE	SUBJECT	ISSUER	AVAILABLE RI	RETURNED REVOCATION INFO
EE Cert	CN = <Username>	ICA: CN = Swisscom Saphir CA 2	OCSP CRL	OCSP-Response
ICA Cert	CN = Swisscom Saphir CA 2	RCA: CN = Swisscom Root CA 2	CRL	http://crl.swissdigicert.ch/sdcs-root2.crl
ICA Cert	CN = Swisscom Diamant CA 2	RCA: CN = Swisscom Root CA 2	CRL	http://crl.swissdigicert.ch/scds-root2.crl
RCA Cert	CN = Swisscom Root CA 2	RCA: CN = Swisscom Root CA 2	-	-

6.5.2 Timestamp Signature

CERTIFICATE	SUBJECT	ISSUER	AVAILABLE RI	RETURNED REVOCATION INFO
TSA Cert	CN = Swisscom TSA 3	ICA: CN = Swisscom TSS CA 2	OCSP CRL	OCSP-Response
ICA Cert	CN = Swisscom TSS CA 2	RCA: CN = Swisscom Root CA 2	CRL	http://crl.swissdigicert.ch/sdcs-root2.crl
RCA Cert	CN = Swisscom Root CA 2	RCA: CN = Swisscom Root CA 2	-	-

[15] http://helpx.adobe.com/acrobat/kb/approved-trust-list2.html

7 Migration Guide

This chapter summarizes all syntax differences between the version 1.5 (subsequently referred as “legacy”) and the version 2.0 (subsequently referred as “new”) of the All-In Signing Reference Guide.

The new interface syntax:

- Allows to use a combination of personal password and One-Time Password (PwdOTP) as an alternative method to MobileID (MID) for the declaration of will.
- Requires the use of the new 1.1 profile for the /signing and /pending requests.

7.1 New Profile

For all requests (for both /sign and /pending), a new profile version must be included in the requests:

Legacy

```
SOAP/XML: <SignRequest Profile="http://ais.swisscom.ch/1.0">
JSON:     "SignRequest": {"@Profile": "http://ais.swisscom.ch/1.0"}
```

New

```
SOAP/XML: <SignRequest Profile="http://ais.swisscom.ch/1.1">
JSON:     "SignRequest": {"@Profile": "http://ais.swisscom.ch/1.1"}
```

7.2 On-Demand Signatures with Declaration of Will

Timestamping, Static signatures and On-Demand signatures without declaration of will, are not affected by the interface modifications, being the only modification in the version profile mentioned above.

7.2.1 Additional Profiles in Signing Request

On-Demand signatures with Declaration of Will are ONLY supported in asynchronous mode. Therefore, all signing requests including step-up (declaration of will) information must include the asynchronous additional profile. In addition to this, if the PwdOTP method is going to be used, the redirect profile must be included as well.

Legacy

```
SOAP/XML:
<SignRequest Profile="http://ais.swisscom.ch/1.0">
…
  <AdditionalProfile>
      http://ais.swisscom.ch/1.0/profiles/ondemandcertificate
  </AdditionalProfile>
…
</SignRequest>

JSON:
"SignRequest": {
  "@Profile": "http://ais.swisscom.ch/1.0"
…
  "AdditionalProfile": "http://ais.swisscom.ch/1.0/profiles/ondemandcertificate
  }
…
}
```

New

```
SOAP/XML:
<SignRequest Profile="http://ais.swisscom.ch/1.1">
…
  <AdditionalProfile>
```

```
      http://ais.swisscom.ch/1.0/profiles/ondemandcertificate
  </AdditionalProfile>
  <AdditionalProfile>
      urn:oasis:names:tc:dss:1.0:profiles:asynchronousprocessing:1.0
  </AdditionalProfile>
  <AdditionalProfile>
      http://ais.swisscom.ch/1.1/profiles/redirect
  </AdditionalProfile>
…
</SignRequest>

JSON:
"SignRequest": {
 "@Profile": "http://ais.swisscom.ch/1.1"
 …
 "AdditionalProfile": [
  "http://ais.swisscom.ch/1.0/profiles/ondemandcertificate",
  "urn:oasis:names:tc:dss:1.0:profiles:asynchronousprocessing",
  "http://ais.swisscom.ch/1.1/profiles/redirect"
 ]
 }
…
}
```

NOTE: if trying to invoke a signature request with the new profile without the asynchronous additional profile, an error will be returned by AIS.

For On-Demand signatures WITH STEP-UP, only ASYNCHRONOUS MODE is supported. On-Demand signatures with step-up using MobileID CAN'T use the synchronous mode anymore. If trying to do so, an error will be returned (RequesterError: InsufficientData).

7.2.2 Step-Up Information in Signing Request

The syntax of the <StepUpAuthorisation> element changes slightly. Since MobileID is not the only method available for accomplishing the Declaration of Will anymore, the <MobileID> element is replaced for the more generic <Phone> element:

Legacy

```
SOAP/XML:
<SignRequest Profile="http://ais.swisscom.ch/1.0">
  …
  <sc:StepUpAuthorisation>
    <sc:MobileID Type="http://ais.swisscom.ch/1.0/auth/mobileid/1.0">
      <sc:MSISDN>MOBILE_NUMBER</sc:MSISDN>
      <sc:Message>MESSAGE</sc:Message>
      <sc:Language>LANGUAGE_CODE</sc:Language>
      <sc:SerialNumber>MobileIDSerialNumber</sc:SerialNumber>
    </sc:MobileID>
  </sc:StepUpAuthorisation>
  …
</SignRequest>

JSON:
"SignRequest": {
  "@Profile": "http://ais.swisscom.ch/1.0"
  …
  "sc.StepUpAuthorisation": {
    "sc.MobileID": {
    "@Type": "http://ais.swisscom.ch/1.0/auth/mobileid/1.0",
    "sc.MSISDN": "MOBILE_NUMBER",
    "sc.Message": "MESSAGE",
    "sc.Language": "LANGUAGE_CODE",
    "sc.SerialNumber": "MobileIDSerialNumber"
    }
  }
  …
}
```

New

```
SOAP/XML:
<SignRequest Profile="http://ais.swisscom.ch/1.1">
  …
  <sc:StepUpAuthorisation>
    <sc:Phone>
      <sc:MSISDN>MOBILE_NUMBER</sc:MSISDN>
      <sc:Message>MESSAGE</sc:Message>
      <sc:Language>LANGUAGE_CODE</sc:Language>
      <sc:SerialNumber>SerialNumber</sc:SerialNumber>
    </sc:Phone>
  </sc:StepUpAuthorisation>
  …
</SignRequest>

JSON:
"SignRequest": {
  "@Profile": "http://ais.swisscom.ch/1.1"
  …
  "sc.StepUpAuthorisation": {
    "sc.Phone": {
    "sc.MSISDN": "MOBILE_NUMBER",
    "sc.Message": "MESSAGE",
    "sc.Language": "LANGUAGE_CODE",
    "sc.SerialNumber": "SerialNumber"
    }
  }
  …
}
```

Taking a closer look, we find the following differences:

- The element MobileID has been replaced by the more generic <Phone> element.
- There is no attribute “Type” in the “Phone” element.
- The serial number is not necessarily a MID serial number anymore, being also possible to be the SNofDN (Serial Number of Distinguished Name) uniquely associated with each user using PwdOTP as Declaration of Will mechanism.
- The Mobile Phone Number can be any mobile phone from any country, it must not be a MID-capable mobile phone number.
- Profile Version is 1.1. instead of 1.0.

7.2.3 Signing Response (pending)

Since asynchronous mode is enforced for on-demand signatures with declaration of will, the first response to the signing request will be a “pending” signing response. If MID is not available and PwdOTP is being used, the pending response will include the consent URL as depicted in the table below.

Legacy

SOAP/XML:
`<SignResponse` **`xsi:type="SignResponse"`** `…>`
`  …`
`  <OptionalOutputs>`
`    <async:ResponseID>RESPONSE_ID</async:ResponseID>`
`  </OptionalOutputs>`
`  …`
`</SignResponse>`
JSON:
`"SignResponse": {`
`    "@RequestID": "YOUR_UNIQUE_ID",`
`    "@Profile": "http://ais.swisscom.ch/1.0",`
`    "Result": {`
`      "ResultMajor": "urn:oasis:names:tc:dss:1.0:resultmajor:Success"`
`    },`
`    "OptionalOutputs": {`
`      "async.ResponseID": "RESPONSE_ID"`
`    }`

New **(Orange: only if PwdOTP is used as declaration of will method)**

SOAP/XML:
`<SignResponse …>`
`  …`
`  <OptionalOutputs>`
`    <async:ResponseID>RESPONSE_ID</async:ResponseID>`
`    <sc:StepUpAuthorisationInfo>`
`      <sc:Result>`
`        <sc:ConsentURL>CONSENT_URL</sc:ConsentURL>`
`      </sc:Result>`
`    </sc:StepUpAuthorisationInfo>`
`  </OptionalOutputs>`
`  …`
`</SignResponse>`
 	
JSON:
`"SignResponse": {`
`    "@RequestID": "YOUR_UNIQUE_ID",`
`    "@Profile":` **`"http://ais.swisscom.ch/1.1"`**`,`
`    "Result": {`
`      "ResultMajor": "urn:oasis:names:tc:dss:1.0:resultmajor:Success"`
`    },`
`    "OptionalOutputs": {`
`      "async.ResponseID": "RESPONSE_ID"`
`      "sc.StepUpAuthorisationInfo": {`
`        "sc.Result": {`
`          "sc.ConsentURL":"CONSENT_URL"`
`        }`
`      }`
`    }`

- After a schema optimization, the attribute xsi :type is not needed in the Response anymore and it is therefore dropped.
- In the JSON response, the included profile is 1.1, same as in the request
- A new element <StepUpAuthorisationInfo> is included in the optional outputs in the response, containing the ConsentURL where the user must be redirected in order to acknowledge the declaration of will. Please notice that this output would be only be delivered if PwdOTP is used as declaration of will method.

7.2.4 Pending Request

For the /pending request, the only difference is the profile version, as described on [7.1].

7.2.5 Pending Response (pending/success)

As for the /pending response:
In case of "pending" (the transaction is not finished yet), the only difference is the profile version, as described on [7.1].

This affects only the JSON response, since the SOAP response does not include the profile version.

In case of "success" (the transaction is finished and the signature computed), the changes are shown in the following table:

Legacy

```
SOAP/XML:
<SignResponse xsi:type="SignResponse" …>
  …
  <OptionalOutputs>
    <sc:MobileIDAPTransID>MOBILEID_AP_TRANSACTION_ID</sc:MobileIDAPTransID>
    <sc:MobileIDSerialNumber>MOBILEID_SERIAL_NUMBER</sc:MobileIDSerialNumber>
    …
  </OptionalOutputs>
  …
</SignResponse>


JSON:
"SignResponse": {
    "@RequestID": "YOUR_UNIQUE_ID",
    "@Profile": "http://ais.swisscom.ch/1.0",
    "Result": {
      "ResultMajor": "urn:oasis:names:tc:dss:1.0:resultmajor:Success"
    },
    "OptionalOutputs": {
      "sc.MobileIDAPTransID": "MOBILEID_AP_TRANSACTION_ID",
      "sc.MobileIDSerialNumber": "MOBILEID_SERIAL_NUMBER",
    }
  …
}
```

New

```
SOAP/XML:
<SignResponse …>
  …
  <OptionalOutputs>
    <sc:APTransID>AP_TRANSACTION_ID</sc: APTransID>
    <sc:StepUpAuthorisationInfo>
      <sc:Result>
        <sc:SerialNumber>"UNIQUE_SERIAL_NUMBER"</sc:SerialNumber>
      </sc:Result>
    </sc:StepUpAuthorisationInfo>
    …
  </OptionalOutputs>
  …
</SignResponse>

JSON :
"SignResponse": {
    "@RequestID": "YOUR_UNIQUE_ID",
    "@Profile": "http://ais.swisscom.ch/1.0",
    "Result": {
      "ResultMajor": "urn:oasis:names:tc:dss:1.0:resultmajor:Success"
    },
    "OptionalOutputs": {
      "sc.APTransID": " AP_TRANSACTION_ID",
      "sc.StepUpAuthorisationInfo":{
        "sc.Result":{
          "sc.SerialNumber":"UNIQUE_SERIAL_NUMBER"
        }
      }
    }
  …
}
```

7.2.6 Error and Status Codes

There is some modifications in the returned error and status codes, which are shown in the following table:

	Legacy	
1	http://ais.swisscom.ch/1.0/resultminor/subsystem/MobileID/service	Service
2	http://ais.swisscom.ch/1.0/resultminor/subsystem/MobileID/verify	The validation of the Mobile ID Signature failed.
3	http://ais.swisscom.ch/1.0/resultminor/subsystem/MobileID/SerialNumberMismatch	The MID serial number was provided in the request and does not match with the one returned by MID.
4	http://ais.swisscom.ch/1.0/resultminor/subsystem/MobileID/status	An unknown status was returned by MobileID.
5	http://ais.swisscom.ch/1.0/resultminor/subsystem/MobileID/transport	A Mobile ID transport error occurred.
	New	
1	http://ais.swisscom.ch/**1.1**/resultminor/subsystem/**StepUp**/service	"StepUp" instead of "MobileID". It applies to both MID and PwdOTP.
2	There is no dedicated minor result for "verify" anymore.	"verify" is new a subcase of "service".
3	http://ais.swisscom.ch/**1.1**/resultminor/subsystem/**StepUp**/cancel	New dedicated minor result for user cancel. It applies to both MID and PwdOTP
4	http://ais.swisscom.ch/**1.1**/resultminor/subsystem/**StepUp**/timeout	New dedicated minor result for transaction timeout. It applies to both MID and PwdOTP
5	http://ais.swisscom.ch/**1.1**/resultminor/subsystem/**StepUp**/SerialNumberMismatch	"StepUp" instead of "MobileID". It applies to both MID and PwdOTP.
6	http://ais.swisscom.ch/**1.1**/resultminor/subsystem/**StepUp**/transport	"StepUp" instead of "MobileID". It applies to both MID and PwdOTP.

Same for "Status" and "SerialNumberMismatch" and "Transport"

The Minor result for subsystem/MobileID/verify does not exist anymore. "Verify" is a subcase of "Service".

7.2.7 User Cancel

In case a user cancels the declaration of will, the optional outputs don't include any soap fault, as it was the case for MID in the legacy version.

Legacy

```
SOAP/XML:
<SignResponse
     RequestID="YOUR_UNIQUE_ID"
     Profile="http://ais.swisscom.ch/1.0"
     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Result>
    <ResultMajor>
      http://ais.swisscom.ch/1.0/resultmajor/SubsystemError
    </ResultMajor>
    <ResultMinor>
        "http://ais.swisscom.ch/1.0/resultminor/subsystem/MobileID/service"
    </ResultMinor>
      <ResultMessage xml:lang="en">mss_401</ResultMessage>
    </Result>
    <OptionalOutputs>
      <sc:MobileIDFault>
        <sc:Subcode xmlns:mss="http://uri.etsi.org/TS102204/v1.1.2#">
          mss:_401
        </sc:Subcode>
        <sc:Reason>USER_CANCEL</sc:Reason>
        <sc:Detail>
          <ns1:detail>User Cancelled the request </ns1:detail>
        </sc:Detail>
      </sc:MobileIDFault>
    </OptionalOutputs>
</SignResponse>

JSON:
"SignResponse": {
  "@RequestID": "YOUR_UNIQUE_ID",
  "@Profile": "http://ais.swisscom.ch/1.0",
  "Result": {
    "ResultMajor": "urn:oasis:names:tc:dss:1.0:resultmajor:Success"
    "ResultMinor":
      "http://ais.swisscom.ch/1.0/resultminor/subsystem/MobileID/service",
    "ResultMessage": {
        "@xml.lang": "en",
        "$": mss_401
      }
    }
  },
  "OptionalOutputs": {
    "sc.MobileIDFault": {
      "sc.Subcode": "mss:_401",
      "sc.Reason": "USER_CANCEL",
      "sc.Detail": {
      "ns1.detail": "User Cancelled the request"
    }
  }
}
```

New

```
SOAP/XML:
<SignResponse Profile="http://ais.swisscom.ch/1.1">
   <Result>
    <ResultMajor>
      http://ais.swisscom.ch/1.0/resultmajor/SubsystemError
    </ResultMajor>
    <ResultMinor>
      **http://ais.swisscom.ch/1.1/resultminor/subsystem/StepUp/cancel**
    </ResultMinor>
    <ResultMessage xml:lang="en"/>
   </Result>
</SignResponse>


JSON:
"SignResponse": {
  "@RequestID": "YOUR_UNIQUE_ID",
  "@Profile": "http://ais.swisscom.ch/1.1",
  "Result": {
    "ResultMajor": "http://ais.swisscom.ch/1.0/resultmajor/SubsystemError"
    "ResultMinor": "http://ais.swisscom.ch/1.1/resultminor/subsystem/StepUp/cancel,
    "ResultMessage": {
        "@xml.lang": "en",
        "$": ""
      }
    }
  }
}
```