eCH-0113: SuisseID specification

Name	SuisseID specification
Standard-Nummer	eCH-0113
Kategorie	Standard
Reifegrad	Verbreitet
Version	1.5
Status	Genehmigt
Genehmigt am	2012-06-27
Ausgabedatum	2012-06-28
Ersetzt Standard	
Sprachen	Englisch
Autoren	Die vorliegende Version der SuisseID Spezifikation wurde durch die “Arbeitsgruppe Spezifikation” des “Trägerschaftsverein SuisseID” erstellt. Die Autoren waren: Michael Doujak, Die Schweizerische Post (Editor); Gerhard Hassenstein, Berner Fachhochschule; Markus Limacher, Swisscom; Marcel Vinzens, AdNovum Informatik; Marc Zweiacker, Zweiacker IT Management; Thomas Moretti, QuoVadis; Urs Bürge, Urs Bürge Beratung GmbH
Herausgeber / Vertrieb	Verein eCH, Mainaustrasse 30, Postfach, 8034 Zürich T 044 388 74 64, F 044 388 71 80 www.ech.ch / info@ech.ch

Summary

The SuisseID is the first standardized product for an electronic proof of identity in Switzerland that can be used for both qualified digital signatures and secure strong authentication.

The SuisseID system provides three functions:

- Strong authentication
- Qualified digital signature
- Electronic identity provider

The components required to provide these functions are defined in this specification.

SuisseID incorporates the ability to create qualified digital signatures according to Swiss digital signature law. All aspects related to the creation or validation of qualified digital signatures are governed by Swiss digital signature law and are considered as integral part of this specification. In case of differences, Swiss digital signature law will override this specification in this aspect and without implications on the remainder of this specification.

The SuisseID is available nation-wide since May 2010 in the form of a smart card or as a token.

Acknowledgements

There would be no SuisseID without the relentless effort, dedication and determination of those who contributed to the success of this specification. Special thanks go to:

Rudolf Brügger, Urs Bürge, Christof Dornbierer, René Eberhard, Nick Hangartner, Manuel Hilty, Freddy Kaiser, Peter Keller, Markus Limacher, Daniel Markwalder, Igor Metz, Thomas Moretti, Rolf Oppliger Carl Rosenast,, Stephan Röthlisberger, Benjamin Schnell, Tom Sprenger, Marcel Vinzens, Hans-Peter Waldegger, Andreas Zürcher, Marc Zweiacker, Reto Zwyssig

Contributing organisations include: AdNovum, AWK Group, Bundesamt für Informatik und Telekommunikation, Enlight-It, Glue Engineering, Keyon AG, QuoVadis, Schweizerische Post, Swisscom, Urs Bürge Beratung GmbH, Zweiacker IT Management.

1 About this Specification

1.1 SuisseID Specifications

This document embraces the specification of two digital certificates along with a service framework in which those certificates can be used for the delivery of personal information by the user.

Chapter 3 is a detailed technical specification of the SuisseID identity and authentication certificate and the SuisseID qualified certificate.

Chapter 0 is a detailed technical specification of the core infrastructure services which comprise the identity provider service (IdP) and the claim assertion service (CAS). Using the core infrastructure services, users can submit personal data in a secure and reliable fashion to the Service Providers that require them. On the other hand, Service Providers can verify the origin and integrity of personal data they obtain from users.

1.2 Privacy and Data Protection

The design of the SuisseID specification was guided by stringent privacy and data protection requirements all along:

- SuisseID certificates contain a minimum of personal data stored on-card;
- SuisseID certificates are exclusively issued by Certificate Authorities which are accredited providers according to ZertES;
- Storage of personal data by the Certificate Authorities follows current practice in accordance to ZertES;
- A subset of the personal data from the identification document (e.g. a passport) is stored in the identity provider service (IdP) operated by the Certificate Authority. The only way to retrieve that data is by strong authentication with the IdP using the appropriate SuisseID authentication certificate;
- Using the IdP, the user will always explicitly acknowledge the submission of personal data to a Service Provider (SP) and he or she can always prevent those data from being submitted.

1.3 Notation of Requirement Levels

The keywords "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this specification are to be interpreted as described in [RFC 2119]. These keywords are thus capitalized when used to unambiguously specify requirements over protocol and application features and behavior that affect the interoperability and security of implementations. When these words are not capitalized, they are meant in their natural-language sense.

1.4 Disclaimer

This specification was developed by a voluntary group whose aim was to establish an industry standard for a digital authentication and signature token in a purely self-regulatory manner. This specification is published as-is, neither the publisher nor any of the group members provide warranties and they do not assume any liabilities with respect to the use of this specification.

Table of contents

2 Status of this Document

The present document has been approved by the eCH committee of experts. It has normative power in the defined domain.

3 SuisseID Digital Certificates Specification

3.1 Purpose of this Chapter

This chapter specifies the profiles of the SuisseID qualified digital certificate QC and the rules pertaining to the issuance and management of the (non-qualified) SuisseID identification and authentication certificate IAC.

3.2 Basic Principles

Unless stated otherwise, the guidelines in the TAV-ZertES [3] apply to the specification of both the QC and the IAC.

The following guidelines are complementary to the specifications in TAV-ZertES when dealing with the issuance and administration of SuisseID certificates[1].

For the issuance and life cycle management of the SuisseID IAC, the Certification Service Provider (CSP) **MUST** adhere to the same organizational and operational procedures and use the same technical infrastructure as they would with the ZertES-compliant qualified certificate.

CSPs issuing digital certificates according to the SuisseID specifications **MUST** be registered with the directory of accredited providers at least for the following standards: ZertES [1], VZertES [2] and TAV-ZertES [3].

3.3 Definitions

3.3.1 SuisseID Number

The SuisseID number is a novel concept to identify certificate owners easily. The following is a list of properties of the SuisseID number:

a) The SuisseID number is a unique number assigned by the CSP to one person exclusively, the certificate owner;

b) The SuisseID number is unique within the scope of all SuisseID certificates;

c) The SuisseID number is assigned to the certificate owner regardless of what other attributes there are in the SuisseID certificate (like organisation unit, for example);

d) A SuisseID certificate set – one QC and one IAC – is always assigned a common SuisseID number;

e) If the certificate owner obtains another SuisseID certificate set, he can ask for one of his SuisseID numbers to be re-allocated to the new certificate set;

f) The certificate owner can ask for several SuisseID certificate sets, each having a different SuisseID number to allow usage of SuisseIDs in different contexts;

g) Renewal of an existing SuisseID certificate set means to re-use the SuisseID number from that set.

[1] Since August 2011 TAV-ZertES refers to RFC 5280.

The SuisseID Number is provided using RDN `serialNumber`[2] of the Subject DN in the SuisseID QC / SuisseID IAC. It **MUST** be provided in the following format:

SuisseID Number Format:

`cspId[0-9]{4}-part1[0-9]{4}-part2[0-9]{4}-part3[0-9]{4}`

Example: 1100-4567-8901-2345

A user may ask for more than one SuisseID and he may ask for them to have equal SuisseID numbers. When this happens, the CSP **MUST** issue the requested certificate sets using a common SuisseID number allocated to all of them.

Certificate owners switching to another SuisseID vendor (CSP) may ask for their current SuisseID number to be transferred to the new CSP. The CSP issuing the new set **MUST** ensure that the number was assigned to the right person beyond doubt in an auditable fashion.

`cspId`	Unique number assigned to the CSP. Within the scope of this specification, `cspId`s are assigned to the CSPs as follows[3]:		
	cspId	**CSP**	**URL**
	1100	Swisscom (Schweiz) AG	www.swissdigicert.ch
	1200	QuoVadis Trustlink Schweiz AG	www.quovadisglobal.ch
	1300	SwissSign AG	www.swisssign.com
	1400	Bundesamt f. Informatik und Telekommunikation BIT	www.pki.admin.ch
`part1-part2-part3`	Unique number: Three blocks of four digits [0-9] each, separated by a dash sign (“-“). The number is assigned by the issuing CSP.		

3.3.2 Certificate Properties

Certificate properties are defined in the RDN `CN` or in RDN `pseudonym` of the subject DN of a SuisseID certificate as follows:

SuisseID QC	(Qualified Signature)
SuisseID IAC	(Authentication)

The property is appended to the last character of the certificate owner's name with a leading blank character. For example:

```
CN=Hans Muster (Qualified Signature)
CN=Hans Muster (Authentication)
pseudonym=Roger Rabbit (Qualified Signature)
pseudonym=Roger Rabbit (Authentication)
```

[2] RFC 3739 [9], chapter 3.1.2. SuisseID certificates use the `serialNumber` attribute solely for the SuisseID number.

[3] Order of appearance acc. to the directory of certified bodies that comply with ZertES (Bundesgesetz über die elektronische Signatur). See http://www.seco.admin.ch/sas/00229/00251/index.html?lang=en

3.3.3 Names

3.3.3.1 Subject DN

With the exception of a certificate's *property*, Subject DN is identical in both certificates of the set. It consists at least of the certificate owner's name and SuisseID number or the certificate owner's pseudonym and SuisseID number.

The name is provided in RDN `CN`[4], the pseudonym in RDN `pseudonym`[5], and SuisseID number in RDN `serialNumber`[6]. Additional RDNs can be added according to the guidelines of the CSPs. They can appear in any order.

Example certificate set 1:

```
CN=Hans Muster (Qualified Signature), serialNumber=1000-9384-9341-8453
CN=Hans Muster (Authentication), serialNumber=1000-9384-9341-8453
```

Example certificate set 2:

```
pseudonym=Roger Rabbit (Qualified Signature),
serialNumber=1000-2948-2300-0077

pseudonym=Roger Rabbit (Authentication),
serialNumber=1000-2948-2300-0077
```

Example certificate set 3:

```
CN=Hans Muster (Qualified Signature),
serialNumber=1000-2284-9341-8489,
C=CH, O=Fabro SA, emailAddress=muster@mail.ch

CN=Hans Muster (Authentication),
serialNumber=1000-2284-9341-8489,
C=CH, O=Fabro SA, emailAddress=muster@mail.ch
```

3.3.3.2 Email Addresses

Usage of the email address is **OPTIONAL**. If they are used, email information **MUST** be identical in both certificates of the set as far as naming, quantity and coding are concerned. They **MAY** be written into RDN `emailAddress`[7] within Subject DN (deprecated) and / or as a `rfc822Name` within `subjectAltName (recommended)`.

3.3.4 Dealing with Representatives

If the certificate owner happens to be a representative of a legal body or organisation according to article 5, paragraph 2 of VZertES [2], then the CSP **MUST** make sure that the legal entity specified in the appropriate attributes is capable of revoking the SuisseID certificates at any time.

Note that other SuisseID certificate sets having the same SuisseID number as the revoked one are not affected by the revocation.

[4] OID = 2.5.4.3
[5] OID = 2.5.4.65
[6] OID = 2.5.4.5
[7] OID=1.2.840.113549.1.9.1

3.4 Technical and Administrative Guidelines for the SuisseID QC

Restrictions, amendments and extensions stated in this chapter apply to the SuisseID QC.

3.4.1 CA Hierarchy – Issuing CA of the SuisseID QC

The CSP **MAY** issue the SuisseID QC using an issuing CA that has been issuing qualified certificates according to ZertES [1] before.

Alternatively, the CSP **MAY** issue SuisseID QC using a separate issuing CA just for that purpose. If so, the issuing CA **MUST NOT** issue anything but SuisseID QC and designated OCSP responder certificates[8]. In this case the issuing CA's certificate object identifier for `CertPolicyId` is `anyPolicy`[9].

3.4.2 SuisseID QC Format

Amendment to chapter 3.4.2, paragraph b) of TAV-ZertES [3] ("Format der Zertifikate der Inhaberinnen und Inhaber")

According to article 7, ZertES [1] and RFC 5280 [6], chapter 4.1, the CSP **MUST** append the fields below to the `tbsCertificate` sequence:

Description	Field	Content
Validity of the certificate	`validity`	According to RFC 5280 [6], chapter 4.1.2.5: a maximum of three years.
Name or pseudonym	`subject`	According to RFC 3739 [9], chapter 3.1.2. See also 3.3.3.1 for certificate properties (qualified signature).
SuisseID number		
specific attributes about the owner, if required		

3.4.3 Tagging of the SuisseID QC

SuisseID QC issued on the basis of the guidelines in this document **MUST** provide the following `explicitText` in the field `UserNotice` of `PolicyInformation`:

```
SuisseID qualified certificate
```

as either an ASN.1 IA5String or an ASN.1 VisibleString.

The object identifier for `CertPolicyId` is managed by the organisation responsible of governing SuisseID[10] and usage of it is restricted to identify SuisseID QCs that comply with the guidelines in this document.

```
OID = 2.16.756.5.26.1.1.1
```

The CSP **MAY** add more `PolicyInformation` to `CertificatePolicies`.

[8] According to RFC 2560, chapter 4.2.2.2
[9] OID=2.5.29.32.0
[10] By the time of this writing, SECO is that organisation

3.4.4 SuisseID QC Format Extensions (REQUIRED)

Amendment to chapter 3.4.2, paragraph c) of TAV-ZertES [3] ("Format der Zertifikate der Inhaberinnen und Inhaber")

According to RFC 5280 [8], chapter 4.2, the CSP **MUST** add the following to the `tbsCertificate` sequence:

Description	Critical	Extension Name	Content
Revoked certificates distribution point	no	cRLDistributionPoints	According to ITU-T X.509 [8], chapter 8.6.2.1 and RFC 5280 [6], chapter 4.2.1.13: Field `reasons` **MUST NOT** appear. DistributionPoints **MUST** specify `DistributionPointName` of type `uniformResourceIdentifier` using the HTTP protocol. Field `cRLIssuer` **MUST NOT** appear. More DistributionPoints **MAY** be added.
Certificate policy and scope, if required	no	certificatePolicies	According to RFC 5280 [6], chapter 4.2.1.4. Use of the `certificatePolicies` extension is specified in 3.4.3.
Identifier of subject public key	no	subjectKeyIdentifier	According to RFC 5280 [6], chapter 4.2.1.2.

3.4.5 SuisseID QC Format Extensions (OPTIONAL)

Amendment to chapter 3.4.2, paragraph c) of TAV-ZertES [3] ("Format der Zertifikate der Inhaberinnen und Inhaber")

According to RFC 5280 [8], chapter 4.2, the CSP **MAY** append the fields below to the `tbsCertificate` sequence:

Description	Critical	Extension Name	Content
Email Address	no	subjectAltName	According to RFC 5280 [6], chapter 4.2.1.6: `rfc822Name {0..n}`
Admission	no	admission	According to Common PKI Specification V2.0 [5], chapter 3.1, Table 29b: `admission {0, 1}` Usage of this extension is defined in 3.4.6

The CSP **MAY** append other extensions.

The CSP **MUST** verify the content of each extension in an auditable fashion.

3.4.6 Admission

The SuisseID QC **MAY** contain an admission. It **MUST NOT** contain more than one admission. It is up to the admission authority to define the appropriate attributes.

The specification of these attributes **MUST** be published at no charge.

3.4.6.1 Admission Authority

The certificate owner's admission **MUST** be confirmed by a competent body, the admission authority. The admission authority **MUST** appear in `directoryName` of the `admissionAuthority` attribute (see [5], Table 29b, #4) using the following attributes in the order listed below:

- `organizationName`: Name of admission authority;
- `countryName`: Country of the admission authority;
- `postalAddress`: Address of the admission authority.

The admission authority **MAY** define additional attributes.

3.4.6.2 Admission

The certificate owner's admission **MUST** be provided as an UTF8String using the `professionItems` attribute (see [5], Table 29b, #16) within `directoryName`. Additionally, the admission's OID **MUST** be provided using the `professionOIDs` attribute (see [5], Table 29b, #17)

The admission authority **MAY** define additional attributes.

3.4.6.3 Dealing with outdated admissions

In order to handle outdated admissions the CSP **MUST** make sure that the admission authority specified in the appropriate attributes is capable of revoking the SuisseID certificates at any time.

3.4.7 Example

The following is a sample extract of a SuisseID QC using ASN.1. The certificate owner is a notary. The admission is approved by an admission authority called *Sample Notary Admission*.

```
SEQUENCE {
  OBJECT IDENTIFIER admission (1 3 36 8 3 3)
--id-commonpki-at-admission
  OCTET STRING, encapsulates {
    SEQUENCE {
--admissionAuthority, directoryName [4]
      [4] {
        SEQUENCE {
          SET {
--Name of the admission authority
            SEQUENCE {
              OBJECT IDENTIFIER organizationName (2 5 4 10)
              UTF8String 'Sample Notary Admission'
              }
            }
          SET {
--Country of the admission authority
            SEQUENCE {
              OBJECT IDENTIFIER countryName (2 5 4 6)
              PrintableString 'CH'
              }
            }
          SET {
--Address of the admission authority
            SEQUENCE {
              OBJECT IDENTIFIER postalAddress (2 5 4 16)
              SEQUENCE {
                UTF8String 'Strassenname 7'
```

```
                UTF8String '8645 Jona'

                UTF8String 'CH'

                }

              }

            }

          }

        }

      SEQUENCE {

        SEQUENCE {

          SEQUENCE {

            SEQUENCE {

              SEQUENCE {

                UTF8String 'Notar'
--Admission as a text

                }

              SEQUENCE {

                OBJECT IDENTIFIER '1 3 36 8 3 11 1 9'
--Admission - professionOIDs11

                }

              }

            }

          }

        }

      }

    }

  }
```

3.5 Technical and Administrative Guidelines for the SuisseID IAC

Restriction, amendments and extensions stated in this chapter apply to the SuisseID IAC.

[11] id-commonpki-at-namingAuthorities-RechtWirtschaftSteuern 9

3.5.1 Audit

Amendment to chapter 3.2, paragraph d) of TAV-ZertES [3] ("Organisation und operative Grundsätze")

Internal audit reports **MUST** capture any deviations from the above documents.

Audit reports **MUST** be archived.

In addition to the documents listed in TAV-ZertES [3] chapter 3.2 paragraph c), the CSP **MUST** include the technical and administrative guidelines into their yearly internal compliance audits. Detected shortcomings **MUST** be corrected using adequate measures.

Audit reports including all references may be demanded by the organisation responsible of governing SuisseID[12] at any time.

3.5.2 Secure Signature Creation Device

Supersedes chapter 3.3.3, paragraph b) of TAV-ZertES [3] ("Sichere Signaturerstellungseinheiten")

Certification of a secure signature creation device requires either

- conformance to FIPS 140-1 [10] or FIPS 140-2 [11], level 3 or above;
- or alternatively, cover examination level EAL 4 of ISO/IEC 15408:2005 [12], increased by the vulnerability assessment element AVA_MSU.3 (vulnerability assessment, analysis and testing of insecure states) and AVA_VLA.4 (vulnerability assessment, highly resistant);
- or alternatively, cover examination level E3 high of ITSEC [13];
- or alternatively, comply with the guidelines of article 6, paragraph 2 of ZertES [1], along with the guidelines defined in chapter 3.3.3, paragraph a) of TAV-ZertES [3] in an auditable fashion.

3.5.3 CA Hierarchy – Issuing CA of the SuisseID IAC

The CSP issues the SuisseID IAC using a SuisseID IAC issuing CA. That SuisseID IAC issuing CA **MUST NOT** issue anything but SuisseID IAC and designated OCSP responder certificates[13]. The issuing CA's certificate object identifier for `CertPolicyId` is `anyPolicy`[14].

The CSP **MAY** add more `PolicyInformation` to `CertificatePolicies`.

[12] By the time of this writing, SECO is that organization.
[13] Accorting to RFC 2560, section 4.2.2.2
[14] OID=2.5.29.32.0

3.5.4 SuisseID IAC Format

Amendment to chapter 3.4.2, paragraph b) of TAV-ZertES [3] ("Format der Zertifikate der Inhaberinnen und Inhaber")

According to article 7, ZertES [1] and RFC 5280 [6], chapter 4.1, the CSP **MUST** append the fields below to the `tbsCertificate` sequence:

Description	Field	Content
Object identifier of the signing algorithm used for signing the certificate	`signature`	According to RFC 5280 [6], chapter 4.1.2.3 and RFC 3279 [7]: `sha-1WithRSAEncryption` or stronger algorithm **MUST** be used according to RFC 4055 [14], chapter 5 [15].
Validity of the certificate	`validity`	According to RFC 5280 [6], chapter 4.1.2.5: a maximum of three years.
Name or pseudonym	`subject`	According to RFC 3739 [9], chapter 3.1.2. See also 3.3.3.1 for certificate properties (identification and authentication).
SuisseID number		
specific attributes about the owner, if required		
Key and algorithm for validating the signature of the certificate owner	`subjectPublicKeyInfo`	According to RFC 5280 [6], chapter 4.1.2.7 and RFC 3279 [7]: `rsaEncryption` with a minimal modulus-length of 2048 bit.

3.5.5 Tagging of the SuisseID IAC

SuisseID IAC issued on the basis of the guidelines in this document **MUST** provide the following `explicitText` in the field `UserNotice` of `PolicyInformation`:

```
SuisseID identity and authentication certificate
```

as either an ASN.1 IA5String or an ASN.1 VisibleString.

The object identifier for `CertPolicyId` is managed by the organisation responsible of governing SuisseID[16] and usage is restricted to identify SuisseID IACs that comply with the guidelines in this document.

```
OID = 2.16.756.5.26.1.1.2
```

The CSP **MAY** add more `PolicyInformation` to `CertificatePolicies`.

[15] `sha256WithRSAEncryption` is recommended.
[16] By the time of this writing, SECO is that organization.

3.5.6 SuisseID IAC Format Extensions (REQUIRED)

Supersedes chapter 3.4.2, paragraph c) of TAV-ZertES [3] ("Format der Zertifikate der Inhaberinnen und Inhaber")

According to RFC 5280 [6], chapter 4.2, the CSP **MUST** append the fields below to the `tbsCertificate` sequence:

Description	Critical	Extension Name	Content
Key identifier of the CSP's key used to sign the certificate	no	authorityKeyIdentifier	According to RFC 5280 [6], chapter 4.2.1.1.
Identifier of subject public key	no	subjectKeyIdentifier	According to RFC 5280 [6], chapter 4.2.1.2.
Certificate scope	yes	keyUsage	According to ITU-T X.509 [7], chapter 8.2.2.3 and RFC 5280 [6], chapter 4.2.1.3: Set bit 0 alone (digital signature).
Certificate policy and scope, if required	no	certificatePolicies	According to RFC 5280 [6], chapter 4.2.1.4. Use of the `certificatePolicies` extension is specified in 3.5.5.
Revoked certificates distribution point	no	cRLDistributionPoints	According to ITU-T X.509 [8], chapter 8.6.2.1 and RFC 5280 [6], chapter 4.2.1.13: Field `reasons` **MUST NOT** appear. DistributionPoints **MUST** specify `DistributionPointName` of type `uniformResourceIdentifier` using the HTTP protocol. Field `cRLIssuer` **MUST NOT** appear. More DistributionPoints **MAY** be added.
CSP certificate access	no	AuthorityInformation Access	According to RFC 5280 [6], chapter 4.2.2.1.
Extended certificate scope	no	ExtendedKeyUsage	According to RFC 5280 [6], chapter 4.2.1.3, the following OID **MUST** be set: `id-kp-clientAuth` according to RFC 5280 [6], chapter 4.2.1.12

Description	Critical	Extension Name	Content
			If a Microsoft UPN for Windows Logon is set in subjectAltName, the following **MUST** be set: `Smart Card Logon`[17] `{0, 1}`
			The CSP **MAY** set additional OIDs.

3.5.7 SuisseID IAC Format Extensions (OPTIONAL)

Amendment to chapter 3.4.2, paragraph c) of TAV-ZertES [3] ("Format der Zertifikate der Inhaberinnen und Inhaber")

According to RFC 5280 [8], chapter 4.2, the CSP **MAY** append the fields below to the `tbsCertificate` sequence:

Description	Critical	Extension Name	Content
Email Address	no	subjectAltName	According to RFC 5280 [6], chapter 4.2.1.6: `rfc822Name {0..n}`
Microsoft UPN for Windows Logon	no	subjectAltName	According to RFC 5280 [6], chapter 4.2.1.6: `User Principal Name`[18] `{0, 1}` Usage of User Principal Name extension is specified in 3.5.8

The CSP **MAY** append other extensions and it **MUST** verify the content of each extension in an auditable fashion.

3.5.8 Microsoft UPN for Windows Logon

Within this specification the Microsoft UPN for Windows Logon is specified as the concatenation of the SuisseID number (as found in the RDN "Serial" of the Subject DN) and the constant string "@upn.suisseid.ch".

The CSP **MUST NOT** use Microsoft User Principal Name in any other way.

The following is an example of a valid Microsoft User Principal Name according to the rules outlined above:

```
1000-2284-9341-8489@upn.suisseid.ch
```

The owner of the domain suisseid.ch **MUST** ensure that there will never be a DNS record of any type for upn.suisseid.ch.

[17] OID = 1.3.6.1.4.1.311.20.2.2
[18] Encoded as `otherName` using OID 1.3.6.1.4.1.311.20.2.3.

3.6 Administration of the IAC issuing certificate with the CSP

Supersedes chapter 3.4.3, paragraph d) of TAV-ZertES [3] ("Verwaltung des für die Ausstellung qualifizierter Zertifikate verwendeten Zertifikats der CSP")

With regard to its own certificates, the CSP **MUST** ensure presence of the following non-critical extensions in the `tbsCertificate` sequence in compliance with RFC 5280 [6], chapter 4.2:

- authorityKeyIdentifier;
- subjectKeyIdentifier;
- certificatePolicies;
- cRLDistributionPoints.

Supersedes chapter 3.4.3, paragraph e) of TAV-ZertES [3] ("Verwaltung des für die Ausstellung qualifizierter Zertifikate verwendeten Zertifikats der CSP")

Section e) is not relevant to the SuisseID IAC.

4 Core Infrastructure Services Specification

4.1 Purpose of this Chapter

This chapter specifies the SuisseID core infrastructure, a set of protocols and services to enable users of the SuisseID to authenticate and disclose personal data in a highly secure and reliable way. The core infrastructure is a partial implementation of the SuisseID Claim Assertion Infrastructure CAI.

4.2 SuisseID Claim Assertion Infrastructure

4.2.1 Overview

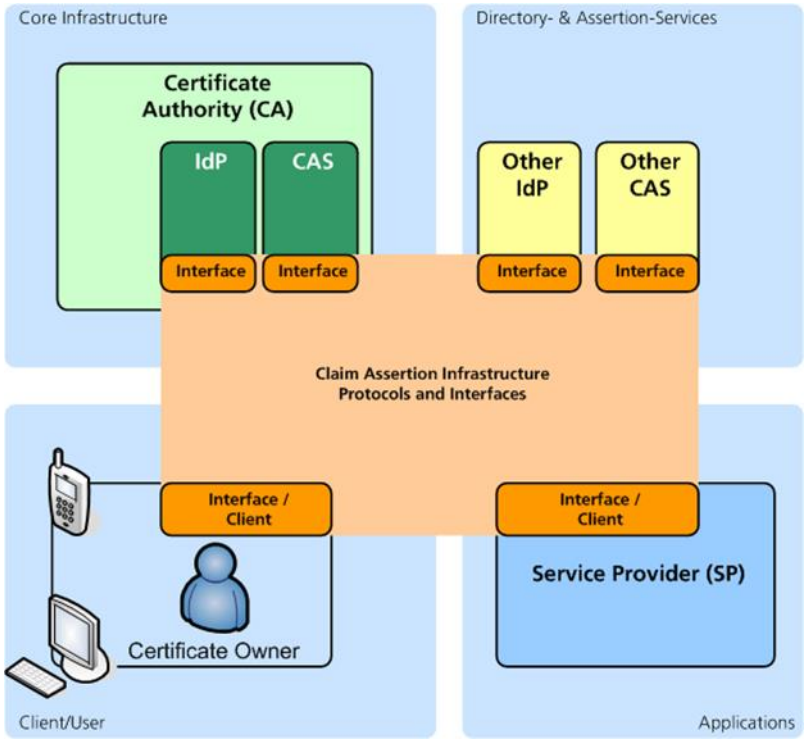

Figure: Claim Assertion Infrastructure

SuisseID certificates contain only little personal data. In order to provide the particulars of the certificate owner to services and applications in the internet, additional services are required. The SuisseID specifications comprise the definition of a set of protocols and interfaces for the disclosure of personal data, called the *Claim Assertion Infrastructure CAI*.

4.2.2 Building Blocks

The CAI is a framework comprised of four building blocks:

1) The ***Core Infrastructure*** consists of two distinct services:

   a) *Identity Provider (IdP).* An authentication authority according to the SAML 2.0 standards [17]. Its purpose is to confirm identity by means of SAML 2.0 assertions;

b) *Claim Assertion Service (CAS)*. Attribute authorities according to the SAML 2.0 standards acting as distributors of SAML 2.0 attribute assertions.

2) ***Client/User***: Recent browser technology is required to support the web frontend usage pattern (Web SSO and web-based attribute query). Other usage patterns can be applied, such as document signing and *Identity Selectors* that support the information card paradigm using WS-Trust identity and attribute assertions [18];

3) ***Applications***: Service Providers integrate services and applications into the CAI, possibly taking advantage of the rich functionality offered by the IdP/CAS. For Web SSO and web-based attribute queries, applications will use SAML 2.0 HTTP POST profile;

4) ***Directory- & Assertion Services***: Directory and database providers may introduce additional IdP and CAS services to provide business-relevant assertions about the certificate owner, like "the person is a registered notary" or "the person is a registered medical doctor".

4.2.3 Design Guidelines

A number of basic principles have guided the design of the CAI.

- *User centric approach* – Access to confidential personal data stored in the IdP and CAS is granted to the user exclusively. It is impossible for an application to discover personal data or get access to them without the certificate owner being actively involved in the retrieval process;
- *Use of open standards* – The CAI defines a set of interfaces and protocols with no prescription about the implementation. Protocols and interfaces are based on open standards such as SAML 2.0 [17] and WS-Trust [18];
- *Platform independency* – The CAI is a platform-independent framework. There is no demand for a specific computer system, architecture, processor, or operating system;
- *Decentralised approach* – The CAI is designed to work as a decentralised system with no requirement for a common, central service. This approach is key to the architecture and guarantees maximum decoupling. Clients, applications and directories can join or leave the CAI without breaking it. The providers of the CAI services constitute a loosely coupled system with only few direct links (interfaces and communication channels). Instead, format conventions and trust relations are being used;
- *Extensibility* – The CAI requires at least one IdP/CAS. However, the CAI is designed such that it can incorporate further IdP and CAS providers. They may be integrated freely with no need for accreditation or special validation procedures.

4.3 Core Infrastructure

4.3.1 Overview

The core infrastructure comprises one IdP and one CAS. There **MUST** be at least one IdP/CAS. The IdP/CAS **MAY** be shared by several CSPs, each of which supplies the particulars of its customers – the certificate owners – to feed the database. Alternatively, a CSP **MAY** operate a separate IdP/CAS of its own, thus adding another IdP/CAS to the core infrastructure landscape.

In support of the SuisseID core infrastructure, CSPs issuing qualified certificates according

to ZertES propably re-use elements of the PKI they operate to a large extent.

As of today, each CSP runs a sophisticated infrastructure comprising a Certification Authority (CA), a Registration Authority (RA), LDAP-based directories, a Time Stamping Authority (TSA) and card management processes to handle and release tokens, cards and certificates.

In order for the core infrastructure to function properly, the CSP …

- **MUST** adapt or extend their RA, CA and card management infrastructure to accomplish full conformance to the specifications in this document;
- **MUST** run a Certificate Owner Database of its own (see below);
- **MUST** either operate an IdP of its own (see 4.4) or supply customer data to an IdP operated elsewhere;
- **MUST** either operate a CAS of its own or supply customer data to a CAS operated elsewhere.

The *Certificate Owner Database* is used to store the particulars of SuisseID customers – the certificate owners – as a set of attributes. The Certificate Owner Database is the data foundation to feed the core IdP/CAS no matter where the IdP/CAS is being operated.

Information stored in the Certificate Owner Database must be comprehensible and correct. The CSP is responsible of taking appropriate measures to ensure correctness and consistency of the data (see 4.6.3.2).

The core infrastructure (combined IdP/CAS) **MUST** be supported by the CSP along with compliance to the SuisseID certficate specifications outlined in Chapter 3 of this document. It can do so by operating a separate IdP/CAS of its own or join a shared IdP/CAS.

4.3.2 Core Components

The core Identity Provider (IdP)

- **MUST** be based on SAML 2.0;
- **MUST** provide SAML tokens (SAML 2.0 assertions) that confirm identity;
- **REQUIRES** the SuisseID IAC to authenticate;
- **MUST** provide SAML 2.0 Web Browser SSO profile with HTTP POST binding;
- **MUST** provide a WS-Trust 1.3 Security Token Service (STS).

The core Claim Assertion Service (CAS) …

- **MUST** be operated in the presence of a core IdP (no stand-alone);
- **MUST** be based on SAML 2.0;
- **MUST** provide attribute assertions (SAML tokens to approve certain attributes);
- **REQUIRES** the SuisseID IAC to authenticate;
- **MUST** provide SAML 2.0 Web-based attribute authority with HTTP POST binding;
- **MUST** provide a WS-Trust 1.3 Security Token Service (STS).

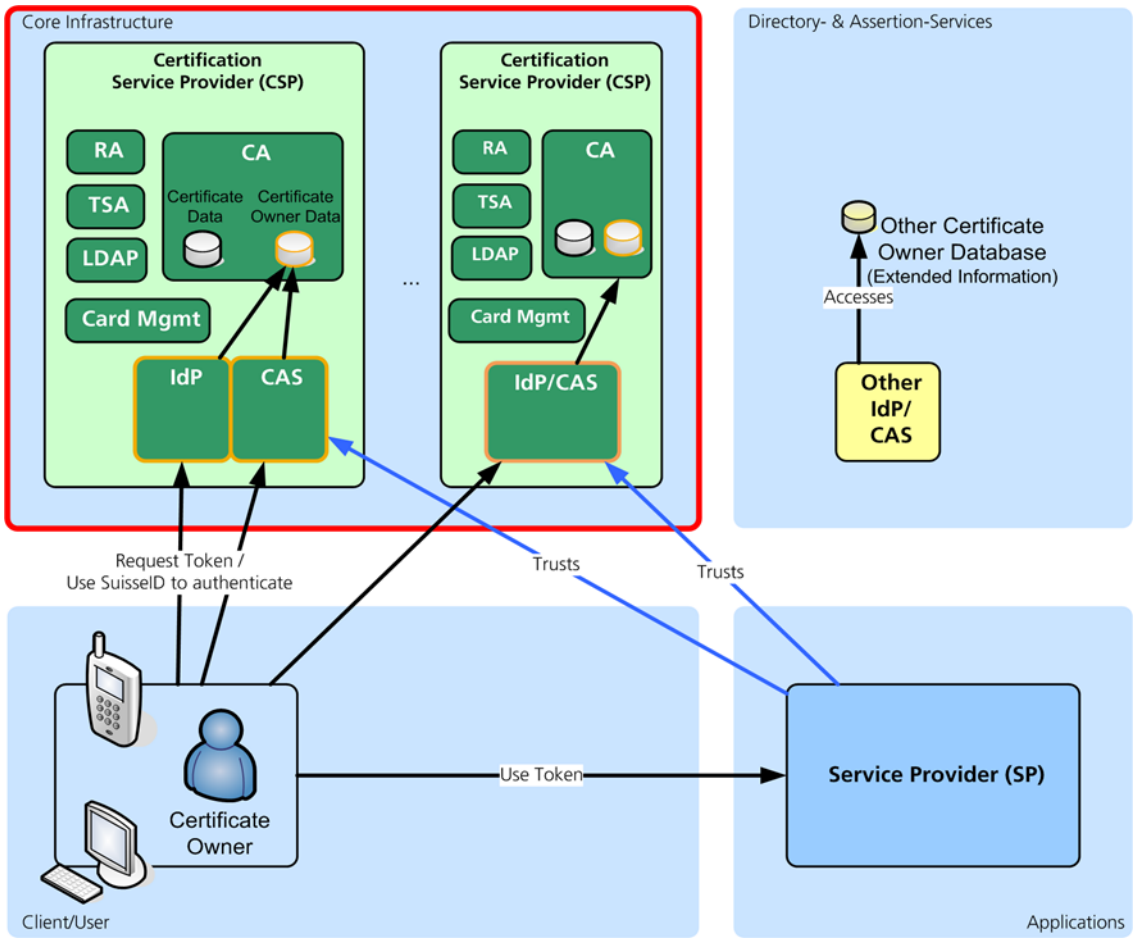

Figure: Overview of Core Infrastructure

The core infrastructure **MUST** support SAML 2.0 Web Browser SSO and a SAML 2.0 Web based attribute authority including the appropriate Web interfaces.

A WS-Trust 1.3 based security token service (STS) **MUST** be provided in addition.

Core IdP and core CAS are two distinct concepts. However, they will most likely be combined into a single service, called the *Extended IdP*.

For both core interfaces, core IdP and core CAS, SuisseID certificates **MUST** be used for authentication.

Core IdP and core CAS **MUST NOT** use anything but the SuisseID IAC for authentication.

The architecture is capable of dealing with a single, shared core IdP instance as well as with many core IdP instances running at the same time.

A shared IdP may serve many CAs. If this is the case, each CSP **MUST** supply the registration data – the particulars of the certificate owners – to the operator of the shared IdP.

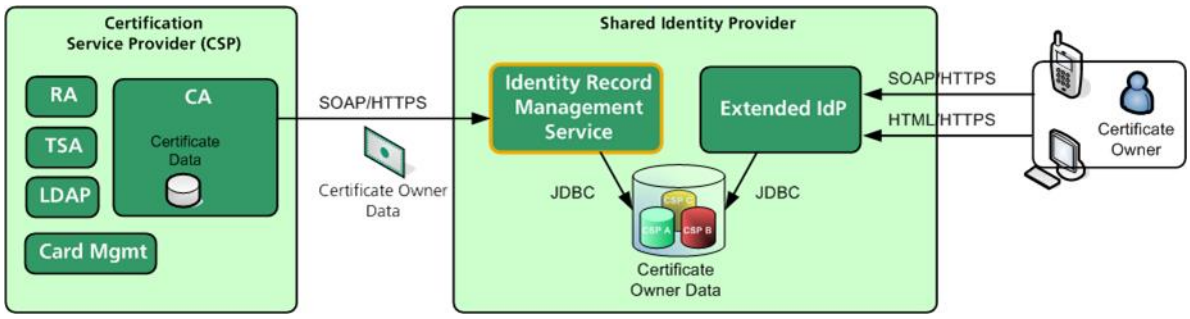

Figure: Sample architecture with shared Identity Provider (IdP)

4.4 Core Identity Provider (IdP)

This chapter describes a sample IdP component architecture.

4.4.1 Overview

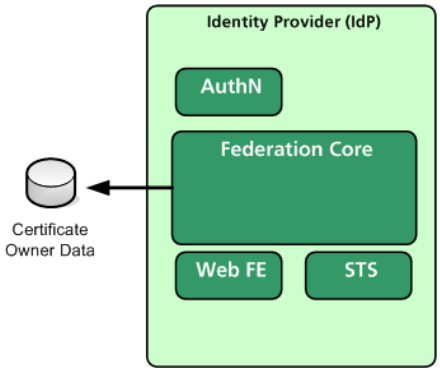

Figure: Identity Provider (IdP) component architecture

The IdP (Identity Provider) is based on SAML 2.0. As a minimal requirement, it consists of the following sub-components (see figure above):

- *Federation core* – Central federation subcomponent for the creation of SAML assertions and the administration of trust relationships;
- *Web FE* – Web frontend interface for Web-SSO, interactive and non-interactive;
- *STS* – WS-Trust based Security Token Service;
- *AuthN* - Authentication subcomponent used by STS as well as by web frontend;
- *Certificate Owner Data* – The source of all IdP attributes.

An IdP can be combined with a CAS to constitute a so-called *Extended IdP*. In this case the following components are added:

- *Attribute* – attribute subcomponent issuing attribute assertions that can be used by the STS and the web frontend;
- *Other Certificate Owner Data* – one or several databases or directories containing additional verified user information (SuisseID assertion attributes, see 4.6.3).

4.4.2 Functionality

4.4.2.1 Issuance of Authentication Assertions

Authentication assertions are used to authenticate against a Service Provider. The IdP provides authentication assertions using either of the two interfaces below after successful authentication with the SuisseID IAC:

- Web frontend for SAML 2.0 Web Browser SSO
- WS-Trust based STS

In either case, Web frontend for SAML 2.0 Web Browser SSO and WS-Trust based STS, authentication with the SuisseID IAC MUST use SSL/TLS client certificate authentication.

4.4.2.2 Issuance of Combined Assertions

When requesting an authentication assertion, other attributes may be requested in addition. In such a case the IdP sends a combined assertion (see 4.6.1.2). This usage pattern constitutes the Extended IdP.

The core infrastructure MUST support combined assertions.

4.4.3 Authentication Scenario

As a sample use case, let the certificate owner authenticate at a Service Provider that requires additional user data not delivered with the IdP authentication assertion. Working in extended IdP mode, the IdP provides the requested attributes along with the authentication statement to the Service Provider using a combined assertion.

In order to use the assertion, the Service Provider needs to trust the issuing IdP. To every SuisseID certificate there is exactly one IdP capable of providing attribute assertions for it.

In a world where there is one IdP shared by all CSPs, every SuisseID certificate, regardless of the CSP issuing it, is bound to the shared IdP and assertions will always be obtained by this one and there is only one core IdP to trust for the Service Provider.

In the presence of more than one IdP Service Providers are almost forced to trust all of them. Another challenge Service Providers face is to find out which of the IdPs to use for authentication. One option is to let the user specify the IdP (see 4.10.1.2). Another way is to let the user authenticate with the Service Provider in the first place. Doing so will disclose the necessary information from the certificate to find out the appropriate IdP.

The predominant use case is to start the user session with the Service Provider. This is called *SP first* according to SAML 2.0.

Each core IdP MUST support the *SP first* scenario.

The user is required to authenticate with the IdP. After authentication with the IdP, the user is re-directed to the URL that was used in the initial service request with the Service Provider. To the user, external authentication with the IdP and re-direction to the Service Provider are handled in a fully transparent way.

4.4.4 Interfaces

4.4.4.1 SAML2.0 based Web Frontend for Authentication and Combined Assertions

The web frontend **MUST** present the contents of the assertion to the user and it **MUST** support SAML 2.0 Web Browser SSO profile with POST binding as a minimal requirement. See 4.9.1 for the technical interface specifications using SAML 2.0 Web Browser SSO.

4.4.4.2 WS-Trust STS for Authentication and Combined Assertions

The WS-Trust service is responsible of providing authentication and combined assertions. See 4.9.2 for technical interface specifications using WS-Trust 1.3 STS.

4.5 Claim Assertion Service (CAS)

4.5.1 Overview

This chapter describes a sample Claim Assertion Service sub-component architecture.

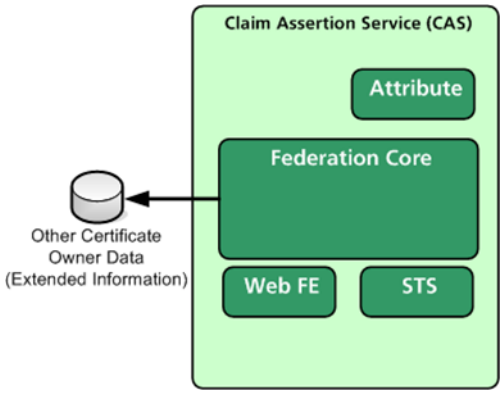

Figure: Claim Assertion Service (CAS) component architecture

The Claim Assertion Service (CAS) is based on SAML 2.0. As a minimal requirement, it consists of the following sub-components (see figure above):

- *Federation core* – central federation subcomponent for the creation of SAML assertions and the administration of trust relationships;
- *Web FE* – Web frontend interface for *interactive* Web-SSO;
- *STS* – WS-Trust based Security Token Service;
- *Other certificate owner data* – one or several databases or directories containing user information.
- A CAS is an additional component to the IdP. It can be combined with the IdP to constitute the so-called *Extended IdP* (see 4.4.1).

4.5.2 Issuance of Attribute Assertions

SAML 2.0 assertions with no authentication statement are called *attribute assertions*. They are used for the assertion of personal data, such as the date of birth or passport number, or others. Attribute assertions can be derived or aggregated from other information. For example, a derived assertion saying "the user is at least 18 years old" can be deduced from the date of birth and the current date.

4.5.3 Controlling Distribution of Attributes and Profiles

When the Claim Assertion Service (CAS) submits the SAML 2.0 assertion to the Service Provider, the SuisseID number is provided along with the attributes. Generally, the source of attributes is a database or directory accessible by the CAS. In case of the core CAS, those attributes contain the particulars of a person gathered during user registration.

A core CAS **MUST** provide the set of attributes according to chapter 4.6.3 of this document.

Service Providers may request some or all of the attributes available from the CAS. The ultimate decision to deliver them is up to the user (certificate owner). He can acknowledge them all at once or restrict to a smaller set or decline at all.

See 4.10.1.6 for a description of how Service Providers request specific attributes from a CAS, and 4.9.2.3 for the WS-Trust based STS.

In case of Web SSO and web based attribute authority, the core IdP/CAS **MUST** provide an appropriate user web interface for the purpose of controlling what core attributes to submit to the Service Provider and to acknowledge submission (see 4.10.1.4).

4.5.4 Interfaces

4.5.4.1 SAML 2.0 based Web Frontend for Attribute Assertions

The web frontend **MUST** present the contents of the assertion to the user. In addition it **MUST** support SAML 2.0 HTTP POST binding (see 4.9.1, SAML 2.0 Web Browser SSO and Attribute Requests with HTTP POST).

4.5.4.2 WS-Trust STS

The WS-Trust STS (Secure Token Service) of the CAS is expected to provide attribute assertions (see 4.9.2, WS-Trust 1.3 STS for a technical interface specification).

In practice, the WS-Trust STS of the SuisseID core CAS will always be combined with the SuisseID core IdP to provide combined assertions (see 4.9.2, WS-Trust 1.3 STS for a technical interface specification).

4.6 Protocols and Interfaces

4.6.1 SAML 2.0

4.6.1.1 Overview

SAML (Security Assertion Markup Language) is an XML framework for exchanging authentication- and authorisation information. It provides functions for describing and sending security related information.

OASIS SAML 2.0 is the technological basis of the Claim Assertion Infrastructure.

4.6.1.2 Types of Assertions

SAML specifies a set of assertion statements that can be issued by a SAML authority (IdP/CAS), including Authentication Statement, Attribute Statement, and Authz Decision Statement. From these assertion statements, new assertion types can be built and combined with a subject. In a SuisseID context, the following assertion types are relevant: authentication assertion, attribute assertion, and combined assertion.

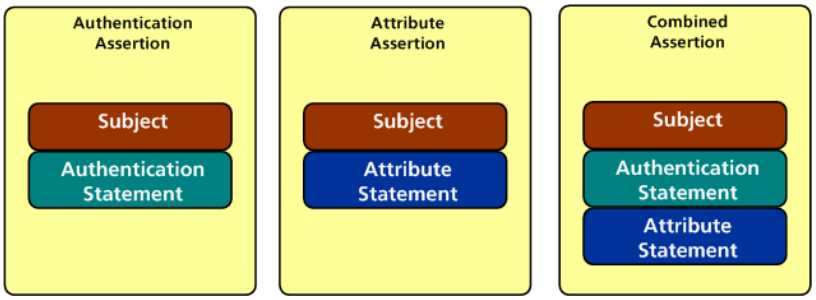

Figure: Types of Assertions

IdPs are providers of authentication assertions.

CAS are providers of attribute assertions.

Extended IdPs (EIdP) provide both authentication assertions and combined assertions.

4.6.1.3 Username in the SAML Assertion - NameID Format

An important constituent of SAML assertions is the *UserID*. SAML 2.0 supports a couple of ways to manage UserID in the IdP and the Service Provider, anonymous and temporary users being among the options. As they are of no use in a SuisseID context they are not elaborated further. Instead, SuisseID specifies its own definition of NameID, based on the "unspecified" NameID format of SAML 2.0.

SuisseID uses a NameID with the following syntax:

`XXXX-XXXX-XXXX-XXXX` (for example: `1234-5678-9012-3456`)

The term "unspecified" indicates that there is a NameID format agreement between IdPs and Service Providers and that none of the predefined NameID formats (transient, persistent, emailAddress, …) has been used.

Service Providers operating in a SuisseID environment use the SuisseID number as the user identification in SAML requests.

Usage of NameID is illustrated in the example below:

```
<saml:Subject>
   <saml:NameID
      Format="urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified">
      1234-5678-9012-3456
   </saml:NameID>
</saml:Subject>
```

4.6.2 XML Namespaces

4.6.2.1 Referenced Namespaces

The CAI specifications reference the following prefixes and XML namespaces:

Prefix	XML Namespace	Reference
eCH-0113	http://www.ech.ch/xmlns/eCH-0113/1	Schema of this specification (see 0)
ic	http://schemas.xmlsoap.org/ws/2005/05/identity	Identity Metasystem Interoperability Schema
icc	http://schemas.xmlsoap.org/ws/2005/05/identity/claims	Claims Schema
md	urn:oasis:names:tc:SAML:2.0:metadata	SAML V2.0 metadata namespace
saml	urn:oasis:names:tc:SAML:2.0:assertion	SAML V2.0 assertion namespace [SAMLCore]
samlp	urn:oasis:names:tc:SAML:2.0:protocol	SAML V2.0 protocol namespace
sp	http://schemas.xmlsoap.org/ws/2005/07/securitypolicy	WS-SecurityPolicy Schema
wsa	http://www.w3.org/2005/08/addressing	WS-Addressing Schema
wsp	http://schemas.xmlsoap.org/ws/2004/09/policy http://www.w3.org/ns/ws-policy	WS-Policy Schema
wsse	http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd	WS-Security Schema
wst	http://schemas.xmlsoap.org/ws/2005/02/trust	WS-Trust Schema
wsu	http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd	WS-Security Schema (Utility)
wsx	http://schemas.xmlsoap.org/ws/2004/08/mex	WS-MetadataExchange
xs	http://www.w3.org/2001/XMLSchema	XML Schema
xsi	http://www.w3.org/2001/XMLSchema-instance	XML Schema: Structures
XAdES	http://uri.etsi.org/01903/v1.1.1#	XML Advanced Electronic Signatures Schema

Table: Referenced XML Namespaces

The prefixes listed in the table above are used in the examples of the CAI specifications.

In the example CAI specifications the declarations that map XML namespaces to prefixes have been omitted to improve readability.

4.6.2.2 SuisseID XML Namespace

The following XML is applied: `http://www.ech.ch/xmlns/eCH-0113/1`

The SuisseID XML namespace is referenced as follows:
`xmlns:eCH-0113=http://www.ech.ch/xmlns/eCH-0113/1`

XML schema location: `http://www.ech.ch/xmlns/eCH-0113/1/eCH-0113-1-0.xsd`

4.6.3 Assertion Attributes

4.6.3.1 Overview

Attributes are used to describe the characteristics of an identity. Attributes have a type and a well defined syntax and semantic. *SuisseID assertion attributes* are issued by the IdP/CAS as part of a SAML 2.0 assertion. In order to map them to local attributes, Service Providers rely on the naming convention specified in this section.

4.6.3.2 SuisseID IdP/CAS Core Assertion Attributes

The SuisseID core assertion attributes are attributes that have been validated by the CSP during the registration process.

Core assertion attributes represent the following personal data:

- Identity card data (ICD): data found in a Swiss ID card or passport and stored in the Certificate Owner's Database.
- Registration process data (RPD): a well-known set of additional attributes gathered during the registration process (e.g. email address). These attributes are provided in the CAS and are added to the subject of the SuisseID certificates. The attribute values MUST be identical.
  RPD has been introduced in the version 1.5 of the SuisseID specification to support use cases, where SP trust the IdP for authentication. In this case, the SP never receives the IAC of the SuisseID and therefore cannot read the attributes from the certificate through the core CAS.

The core IdP/CAS **MUST** support *all of the* core assertion attributes specified in this specification.

Core assertion attributes are divided into three categories:

- Plain core assertion attributes (see 4.6.3.3)
- QC signed core assertion attributes (see 4.6.3.4)
- Derived core assertion attributes (see 4.6.3.5)

Derived core attributes are used to assert properties that are derived from the source attributes, like "IsOver18". The specifications define five derived core attributes and the core IdP/CAS **MUST** provide them all.

Each of the *non-derived* identity card based core assertion attributes **MUST** be provided in two flavors:

1) As a *plain core assertion attribute*. For improved interoperability, the attribute value as such is not signed (see 4.6.3.3);
2) As a *QC signed core assertion attribute*. The attribute value **MUST** be signed using the CA's QC (see 4.6.3.4);

As both versions of an attribute originate from the same source, their contents **MUST** be the same (except for the signature).

The core CAS **MAY** provide more attributes using an appropriate naming convention.

The following rules apply for the definition of attribute names, namespaces and syntax:

- Names used in applications **MUST** have a well defined type and namespace;
- *FriendlyName* **SHALL** be short and descriptive. It is translated into German, French, and Italian for the display of the core assertion attributes (for example in the IdP attribute selection and confirmation dialog box, see 4.10.1.4). For each attribute there is an additional description available in German, French, Italian, and English;
- For interoperability reasons, attributes are taken from the Identity Metasystem Interoperability (IMI) specification [19], whenever possible;
- There are attributes for which standard practices are established or whose definitions in the eCH standards are different from those in the Identity Metasystem Interoperability standard. Those attributes have been defined redundantly using separate namespaces. "Date of birth" is an example of a redundant attribute;
- Encoding is UTF-8 unless stated otherwise.

4.6.3.3 Plain Core Assertion Attributes

The table below defines the plain IdP/CAS core assertion attributes. The referenced XML schema eCH-0113 is attached in the appendix of this document.

Friendly Name	Description	Name	Type	Source
Given Names	Given names	http://www.ech.ch/xmlns/eCH-0113/1/givenNames	eCH-0113:givenNamesType (sequence of xs:string)	ICD
First Name	Preferred name or first name of a Subject. Every IdP/CAS **MUST** use the first name appearing in `givenNames` for this purpose.	http://schemas.xmlsoap.org/ws/2005/05/identity/claims/givenname	icc:StringMaxLength255MinLength1 (xs:string)	ICD
Last Name	Surname, Family name	http://schemas.xmlsoap.org/ws/2005/05/identity/claims/surname	icc:StringMaxLength255MinLength1 (xs:string)	ICD
Date of Birth	If the date is only partially known, this attribute **MUST NOT** be returned.	http://schemas.xmlsoap.org/ws/2005/05/identity/claims/dateofbirth	xs:date	ICD
Date of Birth	May be returned in any of the following formats: YYYY or YYYY-MM or YYYY-MM-DD	http://www.ech.ch/xmlns/eCH-0113/1/dateOfBirthPartiallyKnown	eCH-0113:datePartiallyKnownType (xs:choice)	ICD
Place of Birth	Place of birth according to an official identification document. This attribute is not applicable for a Swiss citizen.	http://www.ech.ch/xmlns/eCH-0113/1/placeOfBirth	eCH-0113:stringMaxLength255Type (xs:string)	ICD

Friendly Name	Description	Name	Type	Source
Origin	Place of origin according to Swiss ID card or passport[19]. Not applicable for foreigners.	http://www.ech.ch/xmlns/eCH-0113/1/origin	eCH-0113:stringMaxLength255Type (xs:string)	ICD
Gender	0: unspecified 1: male 2: female	http://schemas.xmlsoap.org/ws/2005/05/identity/claims/gender	icc:GenderType (xs:token)	ICD
Nationality	ISO 3166-1 alpha-3 codes with modifications (use 000 for stateless persons, use RKS for Kosovars[20])	http://www.ech.ch/xmlns/eCH-0113/1/nationality	eCH-0113:countryIdISO3Type (xs:token)	ICD
Identification Number	Number of the identification document, limited to 9 characters, in accordance to the machine readable zone MRZ as defined in [22] (trailing filler characters must be removed).	http://www.ech.ch/xmlns/eCH-0113/1/identificationNumber	eCH-0113:stringMaxLength9Type (xs:string, according to [22] the maximum length is 9)	ICD
Identification NumberFull	Number of the identification document, limited to 24 characters, in accordance to the visual inspection zone VIZ as defined in [22]	http://www.ech.ch/xmlns/eCH-0113/1/identificationNumberFull	eCH-0113:stringMaxLength24Type (xs:string, according to [22] the maximum length is 24)[21]	ICD
Identification Kind	0: Passport 1: ID 2: Stateless	http://www.ech.ch/xmlns/eCH-0113/1/identificationKind	eCH-0113:identificationKindType (xs:token)	ICD
Issuing Country	Issuing country for the identification document (see Nationality except for 000)	http://www.ech.ch/xmlns/eCH-0113/1/issuingCountry	eCH-0113:countryIdISO3Type	ICD
Issuing Office	Issuing Office	http://www.ech.ch/xmlns/eCH-0113/1/issuingOffice	eCH-0113:stringMaxLength255Type (xs:string)	ICD

[19] Swiss ID cards and passports issued before 2003 may contain more than one origin. In such a case, the attribute would contain all of them separated by a semicolon, e.g. "Sion VS; Oberems VS".
[20] There is no ISO 3166-1 alpha-3 code defined for Kosovo.
[21] According to [22] chapter 6.6 the document number can have variable length. The following text is an exact quote from [22]:

> The number of characters in the VIZ may be variable; however, if the document number has more than 9 characters, the 9 principal characters shall be shown in the MRZ in character positions 6 to 14. They shall be followed by a filler character instead of a check digit to indicate a truncated number. The remaining characters of the document number shall be shown at the beginning of the field reserved for optional data elements (character positions 16 to 30 of the upper machine readable line) followed by a check digit and a filler character.

Friendly Name	Description	Name	Type	Source
Identification Issued On	Issuance date of the identification document	http://www.ech.ch/xmlns/eCH-0113/1/identificationIssuedOn	xs:date	ICD
Identification Valid Until	Valid-through date of the identification document	http://www.ech.ch/xmlns/eCH-0113/1/identificationValidUntil	xs:date	ICD
E-mail Address	Validated e-mail address of the subscriber. MUST be the preferred address for the 'To:' field of email to be sent to the subject and MUST match the email information of the corresponding certificate if they are used.	http://schemas.xmlsoap.org/ws/2005/05/identity/claims/emailaddress	icc:StringMaxLength255MinLength1 (xs:string)	RPD
Organization Name	Authorized name of the organization	http://www.ech.ch/xmlns/eCH-0113/1/organizationName	eCH-0113:stringMaxLength255Type (xs:string)	RPD
Title	Validated title of the subscriber	http://www.ech.ch/xmlns/eCH-0113/1/title	eCH-0113:stringMaxLength255Type (xs:string)	RPD

Table: Plain Core Assertion Attributes

4.6.3.3.1 Example Core Assertion Attribute

The following is an example SAML 2.0 attribute used to assert `eCH-0113:identificationNumber`:

```
<saml:Attribute
   NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri"
   Name="http://www.ech.ch/xmlns/eCH-0113/1/identificationNumber"
   FriendlyName="Identification Number">
   <saml:AttributeValue xsi:type="eCH-0113:stringMaxLenght9Type">
      C01745261
   </saml:AttributeValue>
</saml:Attribute>
```

4.6.3.3.2 Handling of Unknown Values

The core IdP/CAS **MUST** provide all of the plain core assertion attributes specified in this specification.

If an attribute value is unknown or not applicable, the core IdP/CAS **MUST** return the defined exceptional value if there is one defined for it, e.g. "0" in case of icc:gender. If there is no such value defined for the attribute, the IdP/CAS **MUST** omit the `AttributeValue` element. The following is an overview of the rules that apply to unknown values of plain core assertion attributes:

Attribute	Exceptional Value	Handling in the IdP
http://www.ech.ch/xmlns/eCH-0113/1/givenNames	The value is always known for this attribute (no exception here).	The IdP MUST always return an attribute value.
http://schemas.xmlsoap.org/ws/2005/05/identity/claims/givenname	The value is always known for this attribute (no exception here).	The IdP MUST always return an attribute value.
http://schemas.xmlsoap.org/ws/2005/05/identity/claims/surname	The value is always known for this attribute (no exception here).	The IdP MUST always return an attribute value.
http://schemas.xmlsoap.org/ws/2005/05/identity/claims/dateofbirth	-	The IdP MUST omit the `AttributeValue` if `dateofbirth` is not known.
http://www.ech.ch/xmlns/eCH-0113/1/dateOfBirthPartiallyKnown	-	The IdP MUST omit the `AttributeValue` if `dateOfBirthPartiallyKnown` is not known.
http://www.ech.ch/xmlns/eCH-0113/1/placeOfBirth	-	The IdP MUST omit the `AttributeValue` if `placeOfBirth` is not applicable.
http://www.ech.ch/xmlns/eCH-0113/1/origin	-	The IdP MUST omit the `AttributeValue` if `origin` is not applicable.
http://schemas.xmlsoap.org/ws/2005/05/identity/claims/gender	0: unspecified	The IdP MUST return the exceptional value 0 in the `AttributeValue`.
http://www.ech.ch/xmlns/eCH-0113/1/nationality	The value is always known for this attribute (no exception here).	The IdP MUST always return an attribute value.
http://www.ech.ch/xmlns/eCH-0113/1/identificationNumber	The value is always known for this attribute (no exception here).	The IdP MUST always return an attribute value.
http://www.ech.ch/xmlns/eCH-0113/1/identificationNumberFull	The value is always known for this attribute (no exception here) for all SuisseID issued under version 1.5 or higher.	For all SuisseID issued under version 1.5 or higher, the IdP MUST always return an attribute value. For all SuisseID issued prior to version 1.5, the IdP MUST omit the `AttributeValue`.
http://www.ech.ch/xmlns/eCH-0113/1/identificationKind	The value is always known for this attribute (no exception here).	The IdP MUST always return an attribute value.
http://www.ech.ch/xmlns/eCH-0113/1/issuingCountry	The value is always known for this attribute (no exception here).	The IdP MUST always return an attribute value.

Attribute	Exceptional Value	Handling in the IdP
http://www.ech.ch/xmlns/eCH-0113/1/issuingOffice	-	The IdP MUST omit the `AttributeValue` if `issuingOffice` is not known.
http://www.ech.ch/xmlns/eCH-0113/1/identificationIssuedOn	The value is always known for this attribute (no exception here).	The IdP MUST always return an attribute value.
http://www.ech.ch/xmlns/eCH-0113/1/identificationValidUntil	The value is always known for this attribute (no exception here).	The IdP MUST always return an attribute value.
http://schemas.xmlsoap.org/ws/2005/05/identity/claims/emailaddress	-	The IdP MUST omit the `AttributeValue` if `emailaddress` is not known.
http://www.ech.ch/xmlns/eCH-0113/1/organizationName	-	The IdP MUST omit the `AttributeValue` if `organizationName` is not known.
http://www.ech.ch/xmlns/eCH-0113/1/title	-	The IdP MUST omit the `AttributeValue` if `title` is not known.

A dash-sign ("-") indicates that there is no exceptional value.

Table: Handling of unknown values for plain core assertion attributes by the EIdP

4.6.3.4 QC Signed Core Assertion Attributes

In addition to the core assertion attributes that are provided by the IdP/CAS "as-is", each of the identity card based, *non-derived* core assertion attributes **MUST** be provided in terms of a QC signed core assertion attribute. Those attributes are signed by the CA for increased traceability and trust.

For each certificate owner, the CSP delivers the QC signed assertion attributes to the IdP/CAS. The unsigned, plain core assertion attributes are then generated from the signed ones by the IdP/CAS.

The table below defines the mapping of the plain IdP/CAS core assertion attributes to QC signed IdP/CAS core assertion attributes. The referenced XML schema eCH-0113 can be found in the appendix of this document.

Name of Plain Attribute	Name of QC Signed Attribute
http://www.ech.ch/xmlns/eCH-0113/1/givenNames	http://www.ech.ch/xmlns/eCH-0113/1/givenNamesQc
http://schemas.xmlsoap.org/ws/2005/05/identity/claims/givenname	http://www.ech.ch/xmlns/eCH-0113/1/givenNameQc
http://schemas.xmlsoap.org/ws/2005/05/identity/claims/surname	http://www.ech.ch/xmlns/eCH-0113/1/surnameQc
http://schemas.xmlsoap.org/ws/2005/05/identity/claims/dateofbirth	http://www.ech.ch/xmlns/eCH-0113/1/dateOfBirthQc
http://www.ech.ch/xmlns/eCH-0113/1/dateOfBirthPartiallyKnown	http://www.ech.ch/xmlns/eCH-0113/1/dateOfBirthPartiallyKnownQc

Name of Plain Attribute	Name of QC Signed Attribute
http://www.ech.ch/xmlns/eCH-0113/1/placeOfBirth	http://www.ech.ch/xmlns/eCH-0113/1/placeOfBirthQc
http://www.ech.ch/xmlns/eCH-0113/1/origin	http://www.ech.ch/xmlns/eCH-0113/1/originQc
http://schemas.xmlsoap.org/ws/2005/05/identity/claims/gender	http://www.ech.ch/xmlns/eCH-0113/1/genderQc
http://www.ech.ch/xmlns/eCH-0113/1/nationality	http://www.ech.ch/xmlns/eCH-0113/1/nationalityQc
http://www.ech.ch/xmlns/eCH-0113/1/identificationNumber	http://www.ech.ch/xmlns/eCH-0113/1/identificationNumberQc
http://www.ech.ch/xmlns/eCH-0113/1/identificationNumberFull	http://www.ech.ch/xmlns/eCH-0113/1/identificationNumberFullQc
http://www.ech.ch/xmlns/eCH-0113/1/identificationKind	http://www.ech.ch/xmlns/eCH-0113/1/identificationKindQc
http://www.ech.ch/xmlns/eCH-0113/1/issuingCountry	http://www.ech.ch/xmlns/eCH-0113/1/issuingCountryQc
http://www.ech.ch/xmlns/eCH-0113/1/issuingOffice	http://www.ech.ch/xmlns/eCH-0113/1/issuingOfficeQc
http://www.ech.ch/xmlns/eCH-0113/1/identificationIssuedOn	http://www.ech.ch/xmlns/eCH-0113/1/identificationIssuedOnQc
http://www.ech.ch/xmlns/eCH-0113/1/identificationValidUntil	http://www.ech.ch/xmlns/eCH-0113/1/identificationValidUntilQc

Table: QC Signed Core Assertion Attributes

Service Providers **MAY** use either version of the assertion attribute. The QC signed assertion attribute may be considered safer, whereas a plain assertion attribute taken from the Identity Metasystem Interoperability (IMI) specification [19] is certainly better suited for interoperability.

Service Providers having special security needs may not trust the plain attribute although the SAML assertion was signed using the IdP's advanced certificate. They **SHOULD** request the QC signed version of the attribute instead of the plain one (e.g. use *eCH-0113:identificationValidUntilQc* instead of *eCH-0113:identificationValidUntil*).

4.6.3.4.1 Attribute Binding

In order to be verifiable, QC signed core assertion attributes are bound to the appropriate SuisseID number using the following combined data set:

- *name*: Name of the QC core assertion attribute;
- *suisseIdNo*: SuisseID number of the QC;
- *value*: The actual value of the attribute.

Furthermore, QC signed core assertion attributes are linked to the appropriate QC and IAC. Therefore the following information is part of the attribute as well:

- *certIssuerDnQc* and *certSerialNoQc*: A reference to the QC;
- *certIssuerDnIac* and *certSerialNoIac*: A reference to the IAC.

QC signed core assertion attributes contain the actual QC XML signature and a signed timestamp token according to RFC 3161. This timestamp token is used as a proof that the core assertion attribute existed *before* the time of stamping.

The QC signed core assertion attributes **MUST** include the X.509 certificate in the `<ds:KeyInfo>` section.

4.6.3.4.2 Example QC Signed Core Assertion Attribute

The following XML document is a sample QC signed core assertion attribute used to assert the `eCH-0113:givennameQc` attribute:

```
<eCH-0113:signedAttribute>
   <eCH-0113:attribute
      certIssuerDnQc="C=CH, O=SECO, OU=QC, CN=SuisseId-CSP"
      certSerialNoQc="123456"
      certIssuerDnIac="C=CH, O=SECO, OU=IAC, CN=SuisseId-CSP"
      certSerialNoIac="123456"
      name="http://www.ech.ch/xmlns/eCH-0113/1/givennameQc"
      suisseIdNo="1234-1234-1234-1234">
      <icc:givenname>Hans</icc:givenname>
   </eCH-0113:attribute>
   <ds:Signature Id="signature1">
      <ds:SignedInfo>
         <ds:CanonicalizationMethod
            Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
         <ds:SignatureMethod
            Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1" />
         <ds:Reference URI="">
            <ds:Transforms>
            <ds:Transform
               Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature" />
            <ds:Transform
               Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
            </ds:Transforms>
      <ds:DigestMethod
               Algorithm="http://www.w3.org/2000/09/xmldsig#sha1" />
            <ds:DigestValue>VGVzdA==</ds:DigestValue>
         </ds:Reference>
      </ds:SignedInfo>
      <ds:SignatureValue Id="signatureValue1">VGVzdA==</ds:SignatureValue>
      <ds:KeyInfo>
         <ds:X509Data>
            <ds:X509Certificate>[base64 encoded QC]</ds:X509Certificate>
         </ds:X509Data>
      </ds:KeyInfo>
      <ds:Object Id="object1">
         <XAdES:QualifyingProperties Target="#signature1"
            xmlns="http://uri.etsi.org/01903/v1.1.1#">
            <XAdES:UnsignedProperties>
               <XAdES:UnsignedSignatureProperties>
                  <XAdES:SignatureTimeStamp>
                     <XAdES:HashDataInfo uri="#signatureValue1" />
                     <XAdES:EncapsulatedTimeStamp>
               [base64 encoded RFC 3161 timestamp token]
                     </XAdES:EncapsulatedTimeStamp>
                  </XAdES:SignatureTimeStamp>
               </XAdES:UnsignedSignatureProperties>
            </XAdES:UnsignedProperties>
         </XAdES:QualifyingProperties>
      </ds:Object>
   </ds:Signature>
</eCH-0113:signedAttribute>
```

Attribute Reference@URI	In this context, URI="" identifies the node-set (minus any comment nodes) of the element `eCH-0113:signedAttribute` containing the signature.
Element ds:X509Certificate	The X509Certificate element contains the base64-encoded QC of the CA used to sign this core assertion attribute.
Element XAdES:QualifyingProperties	XML advanced electronic signatures (XAdES) [20] defines properties to qualify the whole signature. In case of QC signed core assertion attributes, this is used to incorporate the RFC 3161 timestamp token.
Element XAdES:SignatureTimeStamp	Encapsulates the timestamp over the ds:SignatureValue element (base64 encoded RFC 3161 timestamp token).

4.6.3.4.3 Handling of Unknown Values

The core IdP/CAS **MUST** provide *all of the* QC signed core assertion attributes specified in this specification.

If the attribute value is unknown or not applicable, the core IdP **MUST** *always* return a SAML attribute value of type `eCH-0113:signedAttributeType` for QC signed core assertion attributes, i.e. the attribute value *always* contains an `eCH-0113:attribute` element. The value contained in that particular `eCH-0113:attribute` element is as specified in the "Exceptional Value" column in 4.6.3.3.2.

If no exceptional value is defined for an attribute, the core IdP **MUST** provide an empty `eCH-0113:attribute` element. Hereby the signature and the attribute binding of the QC signed core assertion attribute are retained.

The following example shows how the `eCH-0113:attribute` element of the QC signed core assertion attribute `eCH-0113:placeOfBirthQc` is embedded into the `saml:AttributeValue` element as an empty element:

```
<saml:AttributeValue xsi:type="eCH-0113:signedAttributeType">
   <eCH-0113:attribute
      certIssuerDnQc="C=CH, O=SECO, OU=QC, CN=SuisseId-CSP"
      certSerialNoQc="123456"
      certIssuerDnIac="C=CH, O=SECO, OU=IAC, CN=SuisseId-CSP"
      certSerialNoIac="123456"
      name="http://www.ech.ch/xmlns/eCH-0113/1/placeOfBirthQc"
      suisseIdNo="1234-1234-1234-1234"/>
   <ds:Signature>
      ...
   </ds:Signature>
```

For the attribute identificationNumberFull the IdP/CAS MUST omit the AttributeValue for all SuisseID issued prior to Version 1.5 of this specification.

4.6.3.4.4 Validation Checks

The IdP/CAS **SHOULD** check the qualified signature of QC signed core assertion attributes as well as the integrity of the requested attributes based on the QC signed core assertion attributes before issuing attribute assertions. Additionally, it **SHOULD** check the OID for `CertPolicyId` according to 4.6.3.4.2. Furthermore the IdP/CAS **SHOULD** check coherence of the QC signed core assertion attribute timestamp and the certificate's `ValidFrom` date.

If the date coherence check failed or integrity is broken or an invalid certificate was used to sign a QC signed core assertion attribute, the IdP **MUST NOT** return the assertion.

Service Providers **MAY** perform the above checks for their own safety.

4.6.3.5 Quality of Core Assertion Attributes

As a basic rule, Service Providers are responsible of their own assessment for evaluating the quality (whatever that means) of the assertions issued by a CAS.

Trustworthiness is enhanced by the fact that for each core attribute, the Service Provider can ask for the QC signed assertion in order to verify the attribute's origin and integrity.

Operators of the core IdP/CAS **MUST** conclusively ascertain the correct capturing and immutable storage by providing documented, auditable procedures.

Some of the information in an identification document, like a passport, may change over time and there is a possibility that some of the personal data in a core assertion attribute have become outdated (e.g. last name may have changed after the marriage).

Once stored in the IdP/CAS, core assertion attributes cannot be changed as they are bound to the SuisseID certificate set persistently. Users can obtain a new certificate set at any time, which means they will register anew, yielding a new set of core assertion attributes according to their up-to-date identity card or passport.

Note that obtaining a new certificate set does not affect the assertions that the IdP/CAS issues on the basis of an outdated SuisseID certificate set.

4.6.3.6 Derived Core Assertion Attributes

In addition to the core assertion attributes defined in 4.6.3.2 the core IdP/CAS **MUST** provide the following *derived* core assertion attributes:

Friendly Name	Derived from	Name	Type
Age	eCH-0113:dateOfBirthPartiallyKnown	http://www.ech.ch/xmlns/eCH-0113/1/age	xs:unsignedInt
isOver16	eCH-0113:dateOfBirthPartiallyKnown (return true, iff Age >= 16)	http://www.ech.ch/xmlns/eCH-0113/1/isOver16	xs:boolean
isOver18	eCH-0113:dateOfBirthPartiallyKnown (return true, iff Age >= 18)	http://www.ech.ch/xmlns/eCH-0113/1/isOver18	xs:boolean
age-18-or-over	eCH-0113:dateOfBirthPartiallyKnown (return true, iff Age >= 18) 0 = False 1 = True 2 = Unknown	http://schemas.information-card.net/@ics/age-18-or-over/2008-11	xs:token
isSwissCitizen	`eCH-0113:nationality` (return true, iff nationality == "CHE")	http://www.ech.ch/xmlns/eCH-0113/1/isSwissCitizen	xs:boolean

Table: Derived Core Assertion Attributes

The following rules apply when computing the derived attribute Age:

Age is the number of years between the date of birth and the person's last birthday.

In case the date of birth is not exactly known, the following rules apply in addition:

- If YYYY and MM have been provided, but not DD, then DD is assumed to be the month's last day on the calendar (either of 28, 29, 30 or 31);
- If YYYY has been provided, but neither MM nor DD, then MM is assumed to be 12 (December) and DD is assumed to be 31.

4.6.3.6.1 Handling of Unknown Values

If a derived core assertion attribute value is not deducible from a corresponding source core assertion attribute because the according value is not known, the IdP/CAS **MUST** return the defined expectional value if one has been defined for it (e.g. "2" in case of `ics:age-18-or-over`). If there is no such value defined for the derived attribute, the core IdP/CAS **MUST** omit the `saml:AttributeValue` element.

The following is an overview of the rules that apply to unknown values of derived attributes:

Attribute	Exceptional Value	Handling in the IdP
http://www.ech.ch/xmlns/eCH-0113/1/age	-	The IdP **MUST** omit the `AttributeValue` if the value of the attribute `eCH-0013:dateOfBirthPartiallyKnown` is not known.
http://www.ech.ch/xmlns/eCH-0113/1/isOver16	-	The IdP **MUST** omit the `AttributeValue` if the value of the attribute `eCH-0013:dateOfBirthPartiallyKnown` is not known.
http://www.ech.ch/xmlns/eCH-0113/1/isOver18	-	The IdP **MUST** omit the `AttributeValue` if the value of the attribute `eCH-0013:dateOfBirthPartiallyKnown` is not known.
http://schemas.informationcard.net/@ics/age-18-or-over/2008-11	2: Unknown	`The` IdP **MUST** return the exceptional value `2 in the AttributeValue` if the value of the attribute `eCH-0013:dateOfBirthPartiallyKnown` is not known.
http://www.ech.ch/xmlns/eCH-0113/1/isSwissCitizen	The value is always known for this attribute (no exception here).	The IdP **MUST** always return an attribute value.

A dash-sign ("-") indicates that there is no exceptional value.

4.6.3.7 Commonly Used Interoperable Attribute Statement Types

From a Service Provider's point of view, commonly used attribute statement types can be requested based on interoperable standard core assertion attributes:

- *Proof of name*: Based on `icc:givenname` and `icc:surname`
- *Proof of age*: Based on icc:dateofbirth or isOver16 or isOver18 or ics:age-18-or-over
- *Proof of birthdate*: Based on `icc:dateofbirth`
- *Proof of gender*: Based on `icc:gender`

4.7 Security

4.7.1 Digital Signatures in a SAML 2.0 Context

4.7.1.1 Common Requirements

The core IdP/CAS **MUST** support *RSA-SHA1 Signature Suite* (RSAwithSHA1, http://www.w3.org/2000/09/xmldsig#rsa-sha1)

Service Providers **SHOULD** support this algorithm. The core IdP/CAS **MAY** support other signature algorithms.

4.7.1.2 SAML 2.0 Assertions

Each QC signed core assertion attribute **MUST** contain a qualified signature of the CA.

This is to increase traceability and trust.

The QC used by the CA to sign the core assertion attributes **MUST** contain a `PolicyInformation` with the following object identifier for `CertPolicyId`:

```
OID = 2.16.756.5.26.1.1.3
```

This object identifier for `CertPolicyId` is managed by the organisation responsible of governing SuisseID[22] and usage of it is restricted to identify SuisseID QCs used to sign core assertion attributes.

The CSP **MAY** add more `PolicyInformation` to `CertificatePolicies`.

SAML assertions **MUST** be signed by the core IdP/CAS using an advanced signature.

This is to prevent unauthorized modifications.

The advanced certificate used by the IdP/CAS to sign the SAML assertions **MUST** contain a `PolicyInformation` with the following object identifier for `CertPolicyId`:

```
OID = 2.16.756.5.26.1.1.4
```

This object identifier for `CertPolicyId` is managed by the organisation responsible of governing SuisseID[23] and usage is restricted to identify advanced certificates used to sign SAML assertions that comply with the guidelines in this document.

The CSP **MAY** add more `PolicyInformation` to `CertificatePolicies`.

The IdP/CAS has to include the complete chain of the advanced signature certificate into the signed SAML Assertion for certificate path validation according to RFC 5280 [6]

The IdP/CAS **SHOULD** create the advanced signature on a secure signature creation device according to chapter 3.5.2.

The core IdP/CAS **MUST** protect SAML 2.0 assertions using `<ds:signature>`.

[22] By the time of this writing, SECO is that organisation
[23] By the time of this writing, SECO is that organisation

4.7.1.3 SAML 2.0 Protocol

SAML 2.0 responses **SHOULD** be signed using `<ds:signature>`.

```
Service Providers SHOULD digitally sign SAML 2.0 requests to the
IdP/CAS so they can be authenticated.
```

SAML 2.0 requests **SHOULD** be protected using `<ds:signature>`.

If the Service Provider's request is unsigned, the IdP/CAS **SHOULD** present an appropriate statement or warning about this to the user (see 4.10.1.4).

4.7.2 Digital Signatures in a WS-Trust Context

4.7.2.1 WS-Trust Protocol

WS-Trust request security token requests (RST) and WS-Trust request security token responses (RSTR) **SHOULD** be signed according to the WSS X.509 Certificate Token Profile ([23]).

4.7.2.2 SAML 2.0 Assertions

Issued tokens are SAML 2.0 assertions. See 4.7.1.2 for details on applied signatures.

4.7.3 Encryption

SuisseID requires SSL/TLS secured communication links between the IdP/CAS and the user as well as between the Service Provider and the user.

The core IdP/CAS **MUST** support the following cipher suites for the TLS protocol:

- DHE-RSA-AES256-SHA
- DHE-RSA-AES128-SHA

Service Providers **SHOULD** support these algorithms.

Assertions MUST NOT be encrypted.

4.7.4 Assertion Conditions

One time usage. Each assertion may only be used once. SuisseID requires "One Time Use" to be set. As a consequence of this, the Service Provider **MUST** check the Recipient attribute in the SAML response and make sure that the `InResponseTo` attribute matches the request of the Service Provider.

Quality of time. SAML assertions contain embedded timestamps to reduce the window of opportunity for attacks. Therefore core IdP/CAS **MUST** ensure time synchronization. The maximum clock drift from the reference time (together with whose inaccuracy) **MUST NOT** exceed 1 minute or 10 % of the minimal period of validity of SuisseID core assertions. Service Providers **SHOULD** ensure time synchronization as well. The use of NTP (Network Time Protocol) is recommended.

Short lifetime. Assertions **MUST** have limited lifetime, applying the following conventions:

- `NotBefore`: A maximum of 5 minutes in the past.
- `NotOnOrAfter`: A maximum of 5 minutes into the future.

The Service Provider **MUST** check the validity of assertions obtained from the IdP/CAS and refuse them if expired.

4.7.5 SAML 2.0 Metadata of the core IdP/CAS

SuisseID core IdP/CAS services share a common, small-scale set of functions. That set of functions **MUST** be specified according to SAML 2.0 metadata. The description of it **MUST** be published using a public URL.

The core IdP/CAS **MUST** publish the following configuration information as a minimum:

- entityID
- IDPSSODescriptor (IdP)
- AttributeAuthorityDescriptor (CAS)
- Organization

The XML below is a sample metadata description of core IdP/CAS:

```
<md:EntityDescriptor entityID="https://a-suisseid-idp.ch/saml-idp">    <md:IDPSSODescriptor
WantAuthnRequestsSigned="true"      protocolSupportEnumera-
tion="urn:oasis:names:tc:SAML:2.0:protocol">            <md:KeyDescriptor use="signing">
         <ds:KeyInfo>
            <ds:X509Data>
               <ds:X509Certificate>PC4uLiB0ZXN0IGNlcnRpZmljYXRlIC4uLj4=
               </ds:X509Certificate>
            </ds:X509Data>
         </ds:KeyInfo>
      </md:KeyDescriptor>
      <md:SingleSignOnService
         Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"
         Location="https://a-suisseid-idp.ch/saml-idp/login/" />
   </md:IDPSSODescriptor>
   <md:AttributeAuthorityDescriptor    protocolSupportEnumera-
tion="urn:oasis:names:tc:SAML:2.0:protocol">            <md:KeyDescriptor use="signing">
         <ds:KeyInfo>
            <ds:X509Data>
               <ds:X509Certificate>PC4uLiB0ZXN0IGNlcnRpZmljYXRlIC4uLj4=
               </ds:X509Certificate>
            </ds:X509Data>
         </ds:KeyInfo>
      </md:KeyDescriptor>
      <md:AttributeService
      Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"
      Location="https://a-suisseid-idp.ch/saml-idp/query/" />
   </md:AttributeAuthorityDescriptor>
   <md:Organization>
      <md:Extensions>
         <eCH-0113:suisseidRessourceUrl>https://a-suisseid-idp.ch/saml-idp/res/
         </eCH-0113:suisseidRessourceUrl>
      </md:Extensions>
      <md:OrganizationName xml:lang="de">IdP A</md:OrganizationName>
      <md:OrganizationName xml:lang="en">IdP A</md:OrganizationName>
      <md:OrganizationName xml:lang="fr">IdP A</md:OrganizationName>
      <md:OrganizationName xml:lang="it">IdP A</md:OrganizationName>
      <md:OrganizationDisplayName
         xml:lang="de">SuisseID IdP Firma A</md:OrganizationDisplayName>
      <md:OrganizationDisplayName
         xml:lang="en">SuisseID IdP A</md:OrganizationDisplayName>
      <md:OrganizationDisplayName
         xml:lang="fr">SuisseID IdP Entreprise A</md:OrganizationDisplayName>
      <md:OrganizationDisplayName
         xml:lang="it">SuisseID IdP Ditta A</md:OrganizationDisplayName>
      <md:OrganizationURL
         xml:lang="en">https://a-suisseid-idp.ch</md:OrganizationURL>
   </md:Organization>
   <md:ContactPerson contactType="technical">
      <md:Company>A SuisseID IdP Company</md:Company>
      <md:GivenName>Peter</md:GivenName>
      <md:SurName>Muster</md:SurName>
      <md:EmailAddress>pmu@a-suisseid-idp.ch</md:EmailAddress>
      <md:TelephoneNumber>096 743 45 35</md:TelephoneNumber>
   </md:ContactPerson>
</md:EntityDescriptor>
```

Attribute EntityDescriptor@entityID	Must match the SAML assertion issuer property.
Attribute SingleSignOnService@Location	Service Providers shall send auth requests to this URL.
Attribute AttributeService@Location	Service Providers shall send attribute requests to this URL.
Element eCH-0113:suisseidRessourceUrl	IdP specific base URL of SuisseID resources such as the favorite icon.

Element OrganizationDisplayName	Language-qualified names that are suitable for human consumption (e.g. displayed in the attribute selection/confirmation dialog, see 4.10.1.4).

4.8 Functional Requirements

The tables below provide an overview of the functional requirements imposed on the Identity Provider (IdP) and the Service Provider (SP).

4.8.1 Web Frontend

Function	IdP	SP	Chapter
SAML Web SSO AuthnRequest via HTTP POST	MUST	MUST	4.10.1.3
SAML AttributeQuery via HTTP POST	MUST	MUST	4.10.1.6
SAML Response via HTTP POST	MUST	MUST	0 / 4.10.1.7
NameID Format "unspecified" with SuisseID number support	MUST	MUST	4.6.1.3
Signing of all AuthNRequests and Attribute-Queries	-	SHOULD	4.7.1
Advanced signing of all SAML assertions	MUST	-	4.7.1
Signing of the responses	SHOULD	-	4.7.1
Metadata export/import	MUST	SHOULD	4.7.5
IdP selector dialog	-	MUST	4.10.1.2
Attribute selection/confirmation dialog	MUST	-	4.10.1.4

Table: Web Frontend Requirements

4.8.2 WS-Trust Use Cases

Function	IdP	SP	Chapter
Provisioning STS following the WS-Trust 1.3 (provides the same SAML 2.0 assertions as the web frontend)	MUST	-	4.9.2
NameID Format "unspecified" with support of the SuisseID number	MUST	MUST	4.6.1.3
Signing of all SecurityTokenRequests (RST)	-	SHOULD	4.7.2
Advanced signing of all SAML assertions	MUST	-	4.7.1
Signing of all responses (RSTR)	SHOULD	-	4.7.2
Metadata Export/Import	MUST	MUST	0

Table: WS-Trust Requirements

4.9 Application Profiles

4.9.1 SAML 2.0 Web Browser SSO and Attribute Requests with HTTP POST

4.9.1.1 Overview

SAML 2.0 specifies several ways to exchange SAML requests and assertions between an IdP and a Service Provider.

Service Providers **MUST** support HTTP POST binding for Web Browser SSO and Web-based attribute requests according to *Bindings for the OASIS Security Assertion Markup Language (SAML), V2.0*.

Using POST binding ensures that SAML requests and responses (assertions) are routed through the user's browser instead of being submitted to the Service Provider directly. This is achieved through XHTML documents containing an automatic HTTP POST request.

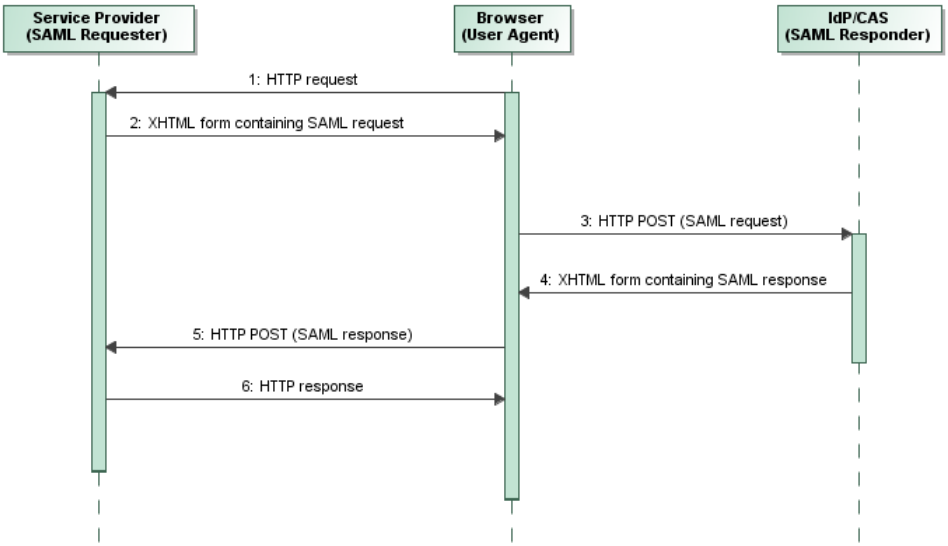

Chart: Attribute request based on HTTP POST binding

1	The user accesses an online application (operated by the Service Provider) using a web browser to make an online purchase order, for example. This is done using a HTTP request.
2	The Service Provider discovers that the user has been authenticated before. In our example, the Service Provider requires a specific assertion about the user (e.g. must be over 18). The Service Provider's answer is a Base-64-encoded SAML request using a XHTML document (see 4.9.1.2). `form action` contains the core IdP's address.
3	The SAML request contained in the HTML (HTTP POST) is transparently forwarded to the IdP/CAS by the browser.

4	The IdP/CAS performs the necessary checks about the user, whether or not he is over 18, and produces the assertion as a base-64-encoded SAML response. The response is sent in a HTML form of a XHTML document to the browser (see 4.9.1.3).
5	The browser submits the SAML response as a HTTP POST request to the Service Provider.
6	From the SAML assertion, the Service Provider has validated that the user is in fact over 18. The order can now be processed further.

An advantage of the HTTP POST binding is the absence of a direct communication link between the Service Provider and the IdP/CAS. Moreover, there is virtually no limit to the size of requests and responses.

Service Providers **MUST** apply HTTP POST binding for all kinds of web-based communication with the IdP/CAS.

4.9.1.2 XHTML Form with SAML Request

The following is a XHTML document with a base-64-encoded SAML request in a HTML form:

```
HTTP/1.1 200 OK
Date: 08 Dec 2009 14:00:59 GMT
Content-Type: text/html; charset=iso-8859-1

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
   <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="de">
   <head>
      <title>SAML Authentication Request POSTer</title>
   </head>
   <body onload="document.forms[0].submit()">
      <form action="https://a-suisseid-idp.ch/saml-idp/login/?lang=de-CH"
   method="post">
         <div>
            <input type="hidden" name="RelayState"
            value="0043beac1b455110dae17004005b13a2b" />
            <input type="hidden" name="SAMLRequest"
            value="&lt;base64 encoded SAML request&gt;" />
         </div>
         <noscript>
            <div>
               <input type="submit" value="Continue" />
            </div>
         </noscript>
      </form>
   </body>
</html>
```

The SAML request input parameter contains the base64 encoded SAML request. Examples SAML requests can be found in 4.10.1.5 (combined requests, element `<samlp:AuthnRequest>`) and 4.10.1.6 (requests to the CAS, element `<samlp:AttributeQuery>`)

The Service Provider can retrieve the action URL from the IdP's metadata (see 4.7.5) and **SHOULD** propagate the user's preferred language to the IdP (using the `lang` query attribute). If `lang` is not specified, the IdP **MUST** use the browser language.

Form encoded messages addressed to the core IdP/CAS **SHOULD** be signed before base64 encoding is applied to the request (see 4.7.1).

If a signature is applied, then the Destination XML attribute in the root SAML element of the protocol message **MUST** contain the URL to which the sender has initially addressed the message.

4.9.1.3 XHTML Form with SAML Response

The following is a XHTML document with base-64-encoded SAML response in a HTML form:

```
HTTP/1.1 200 OK Date: 08 Dec 2009 14:01:04 GMT
Content-Type: text/html; charset=iso-8859-1

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
   <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
   <head>
      <title>SAML Authentication Response POSTer</title>
   </head>
   <body onload="document.forms[0].submit()">
      <form
         action="https://www.a-suisseid-sp.ch/samlconsumer/authenticate"
         method="post">
         <div>
            <input type="hidden" name="RelayState"
               value="0043bfc1bc45110dae17004005b13a2b" />
            <input type="hidden" name="SAMLResponse"
               value="&lt;base64 encoded SAML response&gt;" />
         </div>
         <noscript>
            <div>
               <input type="submit" value="Continue" />
            </div>
         </noscript>
      </form>
   </body>
</html>
```

In a core IdP/CAS response, form-encoded messages **SHOULD** be signed before base64 encoding is applied (see 4.7.1).

4.9.1.4 Relay State

The core IdP/CAS **MUST** support RelayState parameters as indicated in the SAML binding specification.

According to the SAML 2.0 binding specification, *RelayState* data **MAY** be included in a SAML message submitted using the binding. The value **MUST NOT** exceed 80 bytes in length[24].

Relay state **SHOULD** be considered a handle from the point of view of the Service Provider and **SHOULD** be integrity protected by the entity creating the message.

Signing is not realistic given the space limitation, but because the value is exposed to third-party tampering, the entity **SHOULD** ensure that the value has not been tampered with by using a checksum, a pseudo-random value, or similar means.

When a SAML request is accompanied by RelayState data, the SAML responder **MUST** put the data it has received with the request into the corresponding RelayState parameter of the response.

[24] As a user's session always starts with the Service Provider (*SP first* usage pattern according to SAML 2.0), there is no need to transport data.

If no such value is included with a SAML request, or if the SAML response is being generated without prior request, then the SAML responder **MAY** include RelayState data.

4.9.1.5 HTTP and Caching

Standard best practices for HTTP and caching apply to the Claim Assertion Infrastructure, therefore HTTP proxies and the user agent intermediary **SHOULD NOT** cache SAML protocol messages. HTTP responders **SHOULD** apply the following rules when returning SAML protocol messages using HTTP 1.1:

- Include a Cache-Control header field set to "no-cache, no-store"
- Include a Pragma header field set to "no-cache"

There are no other restrictions on the use of HTTP headers.

4.9.1.6 Error Handling

An IdP/CAS may refuse to accept messages from a Service Provider, e.g. because it is black-listed. When this happens, the IdP **MUST** respond with `samlp:StatusCode` set to `urn:oasis:names:tc:SAML:2.0:status:RequestDenied`.

```
<samlp:Response ID="_b3edfeaa-d8b9-48a0-9492-331ab6f53020"
   Version="2.0"  xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"   IssueInstant="2009-
12-05T10:31:09Z"  InResponseTo="_E977CB899FF44B4E74CBDC81A7CC2213">
  <samlp:Status>
      <samlp:StatusCode Value="urn:oasis:names:tc:SAML:2.0:status:Responder">
         <samlp:StatusCode
               Value="urn:oasis:names:tc:SAML:2.0:status:RequestDenied"/>
      </samlp:StatusCode>
   </samlp:Status>
</samlp:Response>
```

HTTP interactions during message exchange **MUST NOT** use HTTP error status codes to indicate failures in SAML processing, since the user agent does not fully support the SAML protocol.

4.9.2 WS-Trust 1.3 STS

4.9.2.1 Overview

The core IdP/CAS **MUST** provide a WS-Trust 1.3 Security Token Service (STS).

To the Claim Assertion Infrastructure, WS-Trust is the general framework for token exchange based on Security Token Service (STS). The protocol specified in this document is a profile for SAML 2 assertions that use WS-Trust.

The sequence diagram below illustrates how client applications use WS-Trust to obtain STS authentication and attribute assertions.

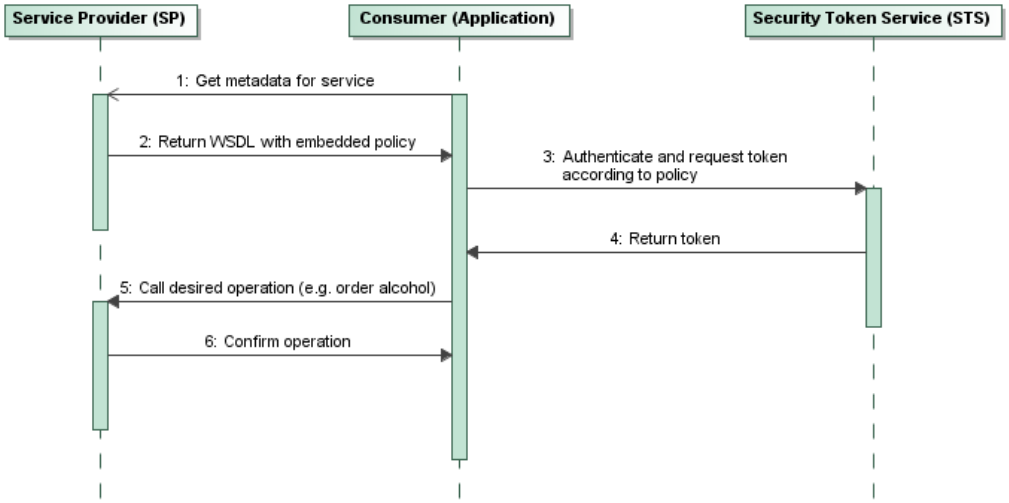

Chart: STS assertions using WS-Trust

1	Optional – The consumer contacts the Service Provider requesting metadata for a service, e.g. an order service.
2	Optional – The Service Provider returns a WSDL document with a policy embedded. The policy specifies what assertions the Service Provider needs.
3	The consumer contacts the STS, authenticates and requests the appropriate token (see 4.9.2.3).
4	The STS issues the token and returns it to the consumer (see 4.9.2.4).
5	The consumer forwards the token to the Service Provider.
6	The Service Provider receives the token and carries on with the order.

4.9.2.2 Metadata (Service Policy)

Service policy metadata specify the requirements of a service to the consumer using interoperable description standards WS-PolicyFramework and WS-PolicyAttachment. In order to invoke the service, users adhere to the prescriptions of the service policy embedded in the metadata of the WSDL document.

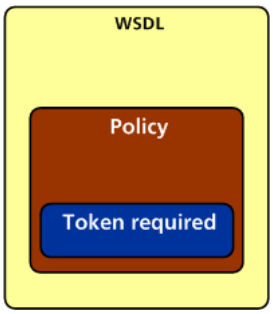

Figure: Service Policy embedded in WSDL

The policies embedded into the WSDL document describe which part of a service call have to be encrypted and signed and what kind of token is expected. The following is a sample policy of a Service Provider requiring SAML tokens in every call:

```
<sp:ProtectionToken>
   <wsp:Policy>
      <wsp:ExactlyOne>
         <wsp:All>
            <sp:IssuedToken sp:IncludeToken="http://schemas.xmlsoap.org/ws/2005/
↳07/securitypolicy/IncludeToken/AlwaysToRecipient">
               <sp:RequestSecurityTokenTemplate>
               <wst:TokenType>urn:oasis:names:tc:SAML:2.0:assertion</wst:TokenType>
                  <wst:Claims
                     Dialect="http://schemas.xmlsoap.org/ws/2005/05/identity">
                     <ic:ClaimType Uri="http://www.ech.ch/xmlns/eCH-0113/1/
↳identificationValidUntil" Optional="false" />
                  </wst:Claims>
               </sp:RequestSecurityTokenTemplate>
            </sp:IssuedToken>
            <ic:PrivacyNotice
               Version="1">http://www.a-suisseid-sp.ch/privacy_policy.html
            </ic:PrivacyNotice>
         </wsp:All>
      </wsp:ExactlyOne>
   </wsp:Policy>
</sp:ProtectionToken>
```

Element IssuedToken/Issuer	There can be many core IdPs, each of which can only serve owners of SuisseID certificates of its own brand. Therefore, element issuer is not set.
Element Claims	Claims are requested SAML attributes, specified using <ic:ClaimType> according to [19]. Attribute dialect MUST be http://schemas.xmlsoap.org/ws/2005/05/identity. This is the only dialect that the STS MUST support in a SuisseID environment.
Element PrivacyNotice	Service Providers SHOULD publish the Privacy Policy URL using this element.

4.9.2.3 STS Token Request (RST)

The following is a sample STS request assuming the security policy in 0:

```
<wst:RequestSecurityToken>
   <wst:TokenType>urn:oasis:names:tc:SAML:2.0:assertion</wst:TokenType>
   <wst:RequestType>http://docs.oasis-open.org/ws-sx/ws-trust/200512/Issue
   </wst:RequestType>
   <wsp:AppliesTo>
      <wsa:EndpointReference>
         <wsa:Address>https://www.a-suisseid-sp.ch</wsa:Address>
      </wsa:EndpointReference>
   </wsp:AppliesTo>
   <wst:Lifetime>
      <wsu:Created>2009-11-01T09:58:29Z</wsu:Created>
      <wsu:Expires>2009-11-01T10:00:09Z</wsu:Expires>
   </wst:Lifetime>
   <wst:Claims Dialect="http://schemas.xmlsoap.org/ws/2005/05/identity">
      <ic:ClaimType
         Uri="http://www.ech.ch/xmlns/eCH-0113/1/identificationValidUntil"
         Optional="false" />
   </wst:Claims>
</wst:RequestSecurityToken>
```

Element TokenType	Describes the kind of tokens requested. In the example above, a SAML 2.0 assertion is requested. SAML 2.0 assertion is the only token type that MUST be supported in a SuisseID environment.
Element EndpointReference	Specifies the entity ID of the Service Provider for whom the assertion is being issued. This is a unique ID that does not reference a genuine WS Endpoint.
Claims	See 0

4.9.2.4 STS Token Response (RSTR)

The following is a sample STS response:

```
<wst:RequestSecurityTokenResponse>
   <wst:Lifetime>
      <wsu:Created>2010-11-17T04:26:59Z</wsu:Created>
      <wsu:Expires>2010-11-17T04:36:39Z</wsu:Expires>
   </wst:Lifetime>
   <wsp:AppliesTo>
      <wsa:EndpointReference>
         <wsa:Address>https://www.a-suisseid-sp.ch</wsa:Address>
      </wsa:EndpointReference>
   </wsp:AppliesTo>
   <wst:RequestedSecurityToken>
      <saml:Assertion ID="__a367dab9354eeb59a6b0299f1a9b3d73"
         IssueInstant="2010-11-02T09:50:26.902Z" Version="2.0">
         <saml:Issuer Format="urn:oasis:names:tc:SAML:2.0:nameid-format:entity">
            https://www.a-suisseid-idp.ch
         </saml:Issuer>
         <saml:Subject>
            <saml:NameID Format="urn:oasis:names:tc:SAML:1.1:nameid-
↳format:unspecified">001-8578-4583-4344</saml:NameID>
         </saml:Subject>
         <saml:Conditions
            NotBefore="2010-11-17T04:26:59.000Z"
            NotOnOrAfter="2010-11-17T04:36:59.000Z">
            <saml:AudienceRestriction>
               <saml:Audience>https://www.a-suisseid-sp.ch</saml:Audience>
            </saml:AudienceRestriction>
         </saml:Conditions>
         <saml:AuthnStatement
            AuthnInstant="2010-11-17T04:26:58.906Z"
            SessionIndex="_f7d9b353-3457-4c55-acb8-bdasga4f8a944b4570">
            <saml:AuthnContext>
               <saml:AuthnContextClassRef>
               urn:oasis:names:tc:SAML:2.0:ac:classes:SmartcardPKI
               </saml:AuthnContextClassRef>
            </saml:AuthnContext>
         </saml:AuthnStatement>
         <saml:AttributeStatement>
            <saml:Attribut
               NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri"
               Name="http://www.ech.ch/xmlns/eCH-0113/1/identificationValidUntil"
               FriendlyName="Identification Valid Until">
               <saml:AttributeValue xsi:type="xs:date">2009-11-10
               </saml:AttributeValue>
            </saml:Attribute>
         </saml:AttributeStatement>
         <ds:Signature>
            ...
         </ds:Signature>
      </saml:Assertion>
   </wst:RequestedSecurityToken>
</wst:RequestSecurityTokenResponse>
```

Element EndpointReference	Specifies the entity ID of the Service Provider for whom the assertion is being issued. This is a unique ID that does not reference a genuine WS Endpoint.
Element Assertion	See 0 and 4.10.1.7
Element AudienceRe-striction	If the request contains a `wsp:AppliesTo` element, then a `saml:AudienceRestriction` containing a `saml:Audience` element **MUST** be included in the response along with the value of that element.

4.9.2.5 Error Handling

Errors use the SOAP Fault mechanism specified by WS-Trust.

4.10 Example Scenarios and Use Cases

4.10.1 Web Login und Attribute Requests using SAML 2.0

This is a step-by-step explanation of the web login using SAML 2.0 attribute requests.

4.10.1.1 Sequence Chart

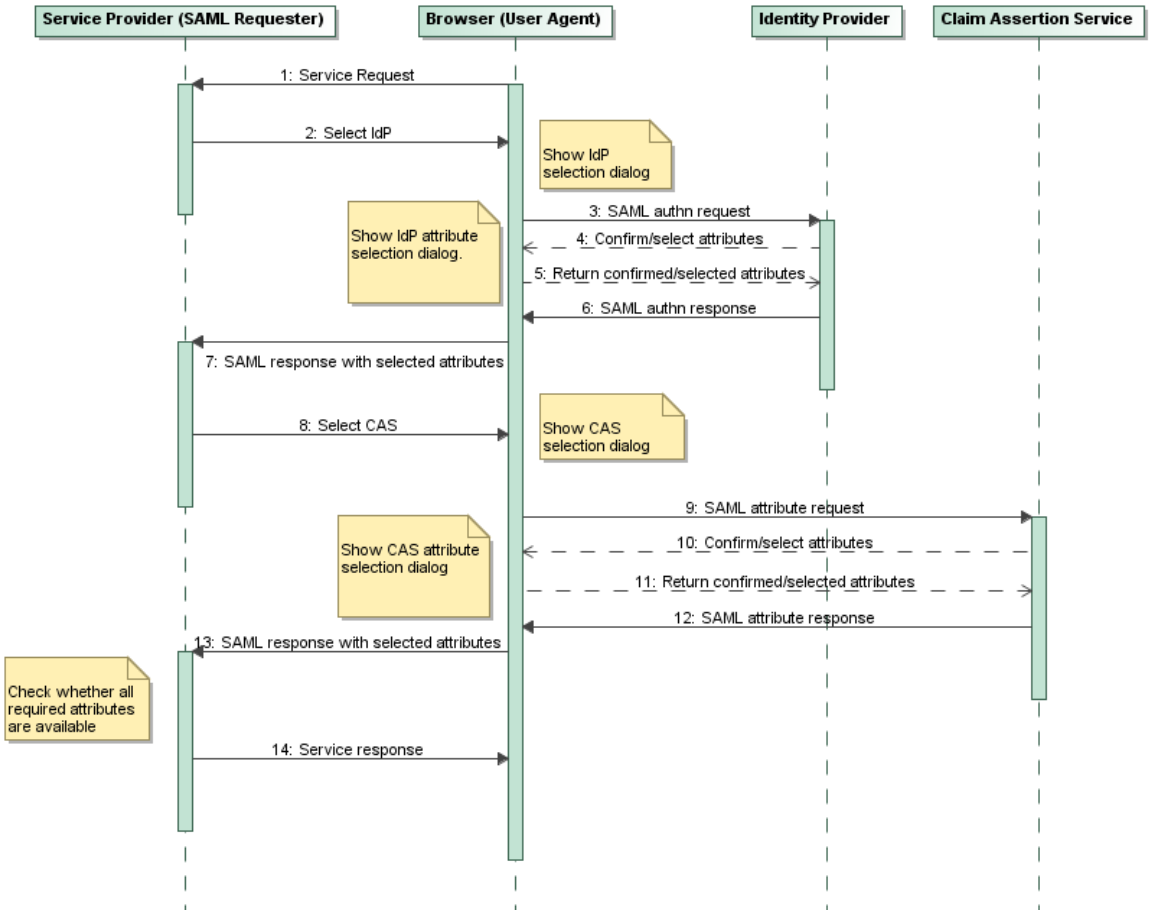

Chart: Web Login and Attribute Request using SAML 2.0

The relevant steps of the login are explained in greater detail below.

4.10.1.2 Step 2 – Display of the IdP Selection Dialog

In the presence of more than one core IdP/CAS, Service Providers may not know which one to use for authentication and assertion requests. Service Providers **MUST** provide a way for the user to specify the appropriate IdP/CAS for authentication and attribute requests.

The following is a sample IdP selection dialog provided by the Service Provider:

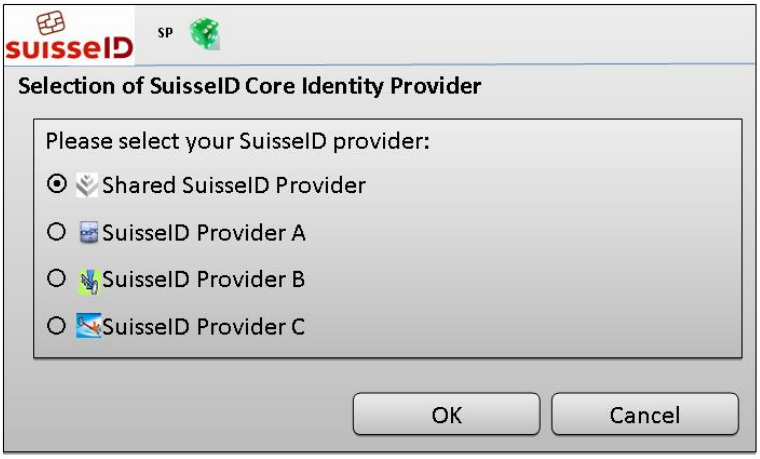

Figure: Sample IdP Selection Dialog

The Service Provider **SHOULD** use the SAML metadata element `OrganizationDisplay-Name` from the IdP and display that name along with the IdP's icon.

IdPs **MUST** provide the icon (favicon.ico, 32x32 pixels) located at the top level within the SuisseID resource directory as specified by SAML metadata (see 4.7.5).

Service Providers **MAY** cache the user's selection as a way to optimise further requests with the same user (using cookies, for example). If so, the IdP selection dialog will not be needed any longer.

Service Providers **MAY** authenticate the user locally using the SuisseID certificate prior to offering further services. Doing so would result in the exchange of certificate information from which the Service Provider can deduce the IdP-URL, thus find out what IdP to use ("authenticate first"). This is considered best practice in situations where the Service Provider has no urgent need for assertion-only authentication. (Assertion-only authentication is a way to implement SSO independent of smartcard PIN caching).

4.10.1.3 Step 3 – Extended AuthnRequest for Combined Assertions

A combined assertion request uses SAML Web SSO `AuthnRequest` extended by a SAML extension. The following example of an extended SAML `AuthnRequest` shows how profile data is demanded with the authentication request:

```
<samlp:AuthnRequest
   Destination="https://a-suisseid-idp.ch/samlprovider/authenticate"   Provider-
Name="SuisseID Service Provider AG"
   ForceAuthn="true"
   ID="identifier_1"
   IsPassive="false"
   IssueInstant="2009-11-21T11:43:08.3344Z"
   Version="2.0">
   <saml:Issuer>https://a-suisseid-sp.ch/saml/acs</saml:Issuer>
   <samlp:Extensions>
      <saml:Attribute
         NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri"
         Name="http://www.ech.ch/xmlns/eCH-0113/1/identificationNumber"
         eCH-0113:required="true" />
      <ic:PrivacyNotice Version="1">http://a-suisseid-sp.ch/privacy_policy.html
      </ic:PrivacyNotice>
   </samlp:Extensions>
</samlp:AuthnRequest>
```

Attribute ForceAuthn	In case of an AuthnRequest that is combined with attribute request (as in the example above) the Service Provider MUST set attribute ForceAuthn to true and the IdP MUST enforce usage of it. In case of a plain AuthnRequest that is not combined with an attribute request, the attribute ForceAuthn MAY be set to false to support SSO.
Element Extensions	The above AuthnRequest shows how attribute request can be packed into a SAML extension using SAML attributes with the XML attribute eCH-0113:required added, yielding a combined assertion issued by the IdP. eCH-0113:required is "false" by default, thus rendering an attribute optional if it is not explicitely set to "true". SAML extensions are a speciality of SuisseID – there is no SAML specification defining that particular mechanism. If the AuthnRe-quest does not have the attributes specified in the extension, then the IdP issues a plain authentication assertion.

Token requests **SHOULD** be signed by the Service Provider.

The idea behind this is to allow Service Provider ratings of various kinds for the future, e.g. to give a visual hint of whether the Service Provider is trusted by the IdP. The above example request does not contain a signature for better reading.

The core IdPs/CAS **MAY** reject requests from specific Service Providers.

One day, Service Providers may appear on black lists referenced by the IdP/CAS operators. However, blacklisting is out of scope of the SuisseID specifications.

4.10.1.4 Step 4 – Display / Confirmation of the Requested Attributes

The IdP/CAS **MUST** display a confirmation page showing the requested data in an attribute request to the IdP/CAS (attribute selection and approval dialog).

The confirmation page **MUST** show the name and value of each attribute that was requested.

Attributes labelled as "required" cannot be modified, nor waved by the users. In fact, the only choice the user has is to confirm the attributes by pressing OK or dismiss the request entirely by pressing the Cancel button.

In any case the Service Provider **MUST** check completeness of the attributes of type "required".

The following is a sample confirmation dialog provided by the IdP/CAS.

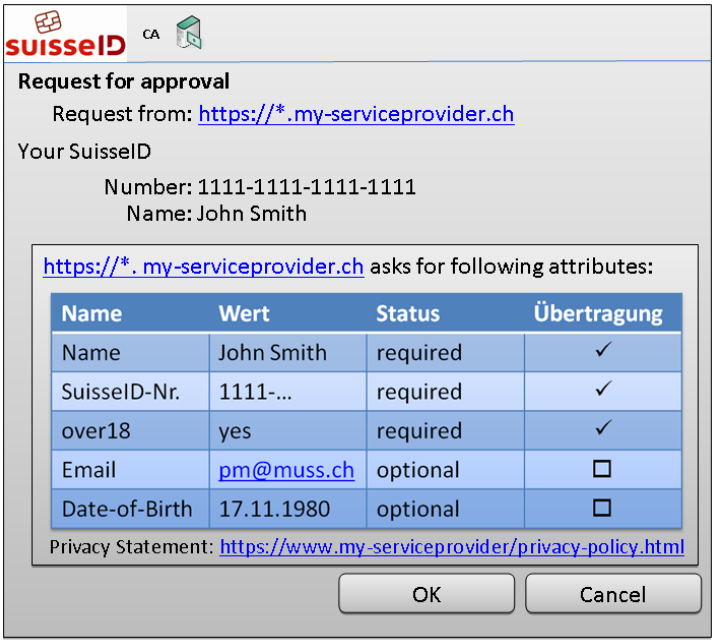

Figure: IdP Confirmation Dialog

Optional attributes **MUST** be de-selected by default. Users **MAY** select them deliberately one by one.

In case of a plain AuthnRequest that is not combined with an attribute query, only a minimal dialog for the attribute selection/confirmation is displayed. This is to assure user awareness when transmitting the SuisseID number (e.g. in an SSO scenario).

In the attribute selection and confirmation screen, the following **MUST** appear:

- IdP identification header;
- Service Provider identification data, either name or issuer information from the SAML request;
- SuisseID number;
- Name of the SuisseID owner, either the RDN "cn" or the RDN "pseudonym" of the IAC subject DN;
- Privacy policy of the Service Provider (from SAML request);

- If a QC signed attribute is requested, the IdP/CAS **SHOULD** display the actual value only. It **MUST** provide a way to dig into the details and view the whole attribute including the QC signature.

When the user presses the Cancel button, the IdP **MUST** return a SAML error response (see 4.9.1.6). The response **SHOULD** contain a second-level `<samlp:StatusCode>` value of `http://www.ech.ch/xmlns/eCH-0113/1/abortedByUser`. Service Providers **SHOULD** interpret this status code and display a user friendly message.

4.10.1.5 Step 6 – SAML Assertion for Combined Requests

The following is a simplified example of a SAML assertion showing the response to an AuthnRequest that was combined with a request for profile data.

Note that `<samlp:Response>` and signatures are skipped for better reading.

```
<saml:Assertion   ID="_896597f4-5abe-4181-b47a-b9c1dcc2d82d"
   IssueInstant="2009-08-21T14:11:08.739Z" Version="2.0">
   <saml:Issuer>SUISSE_ID_IDP_xyz</saml:Issuer>
   <saml:Subject>
      <saml:NameID Format="urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified">
      1234-5678-9012-3456
      </saml:NameID>
      <saml:SubjectConfirmation Method="urn:oasis:names:tc:SAML:2.0:cm:bearer">
         <saml:SubjectConfirmationData
            InResponseTo="identifier_1" NotOnOrAfter="2009-08-22T05:11:08.739Z"
            Recipient="https://a-suisseid-sp.ch/saml/acs" />
      </saml:SubjectConfirmation>
   </saml:Subject>
   <saml:Conditions
      NotBefore="2009-08-22T14:01:08.739Z"
      NotOnOrAfter="2009-08-22T14:11:08.739Z">
      <saml:AudienceRestriction>
         <saml:Audience>https://a-suisseid-sp.ch/</saml:Audience>
      </saml:AudienceRestriction>
   </saml:Conditions>
   <saml:AuthnStatement
      AuthnInstant="2009-08-21T14:11:08.739Z"
      SessionIndex="_f7d9b353-3457-4c55-acb8-bdasga4f8a944b4570">
      <saml:AuthnContext>
         <saml:AuthnContextClassRef>
            urn:oasis:names:tc:SAML:2.0:ac:classes:SmartcardPKI
         </saml:AuthnContextClassRef>
      </saml:AuthnContext>
   </saml:AuthnStatement>
   <saml:AttributeStatement>
      <saml:Attribute
         NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri"
         Name="http://www.ech.ch/xmlns/eCH-0113/1/identificationNumber"
         FriendlyName="Identification Number">
         <saml:AttributeValue
            xmlns:xs="http://www.w3.org/2001/XMLSchema"
            xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
            xsi:type="eCH-0113:stringMaxLength9Type">
            C01745261
         </saml:AttributeValue>
      </saml:Attribute>
   </saml:AttributeStatement>
</saml:Assertion>
```

Element `Subject`	The SuisseID number is part of the subject (see 4.6.3.4)
Attributes NotBefore and NotOnOrAfter	The values combine to a 10 minutes timespan.
Element AudienceRestriction	The assertion is valid only for a particular Service Provider, the one who initiated the request using HTTP POST.

Element AuthnStatement	`urn:oasis:names:tc:SAML:2.0:ac:classes:SmartcardPKI` is the class to be referenced as SuisseID is the only authentication method.
Element AttributeStatement	AttributeStatment is used to return the requested SuisseID core attributes.

4.10.1.6 Step 9 – SAML Attribute Request to the CAS

The following is a sample attribute request for obtaining the SAML attribute `eCH-0113:identificationValidUntil`:

```
<samlp:AttributeQuery
   ID="acf23494-1743-2155-4334a-fc453464ab56"
   Version="2.0"
   IssueInstant="2009-11-26T20:31:40Z"
   Destination="https://a-suisseid-idp.ch/samlprovider/query">
   <saml:Issuer>https://a-suisseid-sp.ch/saml/acs</saml:Issuer>
      <samlp:Extensions>
         <ic:PrivacyNotice Version="1">http://a-suisseid-sp.ch/privacy_policy.html
         </ic:PrivacyNotice>
        <eCH-0113:assertionConsumerServiceUrl>
            https://www.a-suisseid-sp.ch/saml/acs
         </eCH-0113:assertionConsumerServiceUrl>
      </samlp:Extensions>
      <saml:Subject>
         <saml:NameID Format="urn:oasis:names:tc:SAML:1.1:nameid-↳format:unspecified">1234-
5678-9012-3456
         </saml:NameID>
      </saml:Subject>
      <saml:Attribute NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri"
         Name="http://www.ech.ch/xmlns/eCH-0113/1/identificationValidUntil"
         FriendlyName="Identification Valid Until"
         eCH-0113:required="true" />
</samlp:AttributeQuery>
```

Element eCH-0113:assertionConsumerServiceUrl	In the SAML authentication request protocol, <AuthnRequest> contains an attribute called AssertionConsumerServiceURL to specify the receiver of the data in <Response>. The SAML assertion query and request protocol does not have such an attribute, mostly because it is normally used for synchronous bindings. However, user centricity requires that asynchronous front channel binding can be done. For that purpose eCH-0113:assertionConsumerServiceUrl has been introduced as a SuisseID-specific assertion query/request profile within the Extensions element of <AttributeQuery> to specify the location to which <Response> must be returned. If the element is not specified in a request, the IdP SHOULD take the location from either the metadata or from the issuer element.

If no SAML attributes are specified in the SAML attribute request then this is interpreted as a request for all attributes allowed by policy.

4.10.1.7 Step 12 – SAML Attribute Response

The following is an attribute response to the above request. Note that `<samlp:Response>` and signatures are skipped for better reading.

```
<saml:Assertion ID="aaf23196-1773-2113-474a-fe114412ab72"
   IssueInstant="2009-11-21T14:11:08.739Z" Version="2.0"
   <saml:Issuer>https://a-suisseid-idp.ch/saml-idp</saml:Issuer>
   <saml:Subject>
      <saml:NameID Format="urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified">
         1234-5678-9012-3456</saml:NameID>
      <saml:SubjectConfirmation Method="urn:oasis:names:tc:SAML:2.0:cm:bearer">
         <saml:SubjectConfirmationData InResponseTo="identifier_1"
            NotOnOrAfter="2009-08-22T05:11:08.739Z"
            Recipient=" https://a-suisseid-sp.ch/saml-sp/acs" />
      </saml:SubjectConfirmation>
   </saml:Subject>
   <saml:Conditions NotBefore="2009-08-21T14:01:08.739Z"
      NotOnOrAfter="2009-08-22T05:11:08.739Z">
      <saml:AudienceRestriction>
         <saml:Audience>https://a-suisseid-sp.ch/</saml:Audience>
      </saml:AudienceRestriction>
   </saml:Conditions>
   <saml:AttributeStatement>
      <saml:Attribute NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri"
         Name="http://www.ech.ch/xmlns/eCH-0113/1/identificationValidUntil"
         FriendlyName="Identification Valid Until">
         <saml:AttributeValue xsi:type="xs:date">2009-11-10</saml:AttributeValue>
      </saml:Attribute>
   </saml:AttributeStatement>
</saml:Assertion>
```

4.10.2 Attribute Request to an STS with WS-Trust using Information Cards

This section introduces the usage of STS with Information Cards.

Core IdP/CAS providers **MAY** support Information Card technology.

There is no obligation for a IdP/CAS provider to support Information Cards.

4.10.2.1 Sequence Chart

Information Cards are based on mechanisms described in [18] and [21]. They allow for a digital identity to be integrated in a user-centric identity framework that promotes interoperability between identity providers and relying parties with the user in control.

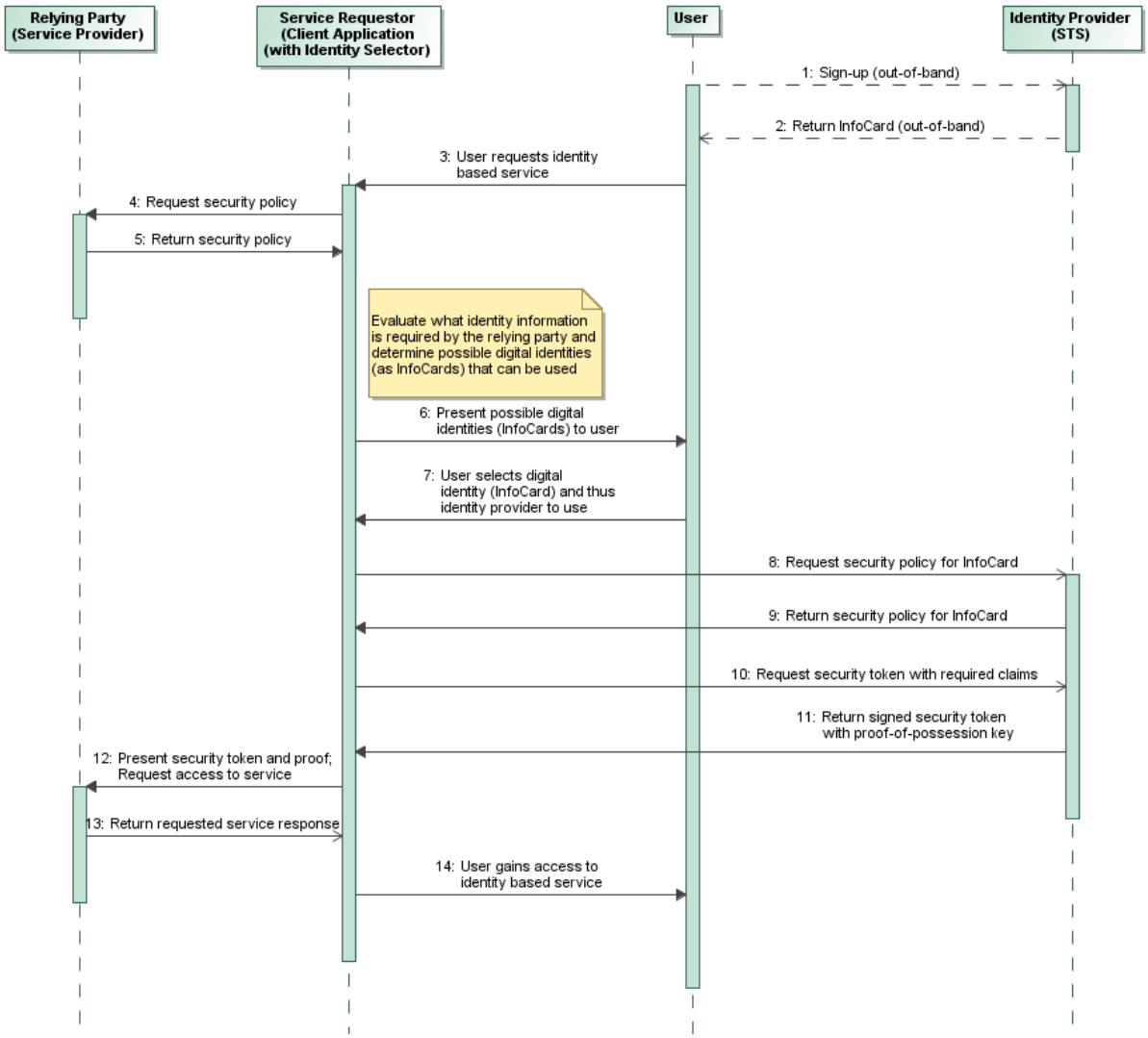

Chart: STS/WS-Trust Confirmation Card Dialog

4.10.2.2 Identity Provider Information Card

An Information Card represents a digital entity that can be issued by an IdP. Technically speaking, Information Cards are signed XML documents issued by the IdP. The following is sample Information Card XML code:

```
<ic:InformationCard>
   <ic:InformationCardReference>
      <ic:CardId>eCH-0113:csp-0001</ic:CardId>
      <ic:CardVersion>1</ic:CardVersion>
   </ic:InformationCardReference>
   <ic:CardName>CSP-0001 SuisseID Information Card</ic:CardName>
   <ic:CardImage MimeType="image/png">___CSP-0001 SuisseID Logo___</ic:CardImage>
   <ic:Issuer>https://a-suisseid-idp.ch:7443/card</ic:Issuer>
   <ic:TimeIssued>2010-05-29T22:32:17Z</ic:TimeIssued>
   <ic:TimeExpires>9999-12-31T23:59:59.9999999Z</ic:TimeExpires>
   <ic:TokenServiceList>
      <ic:TokenService>
         <wsa:EndpointReference xmlns:wsa="http://www.w3.org/2005/08/addressing">
            <wsa:Address>https://a-suisseid-idp.ch:7443/sts</wsa:Address>
            <wsa:Metadata>
               <mex:Metadata xmlns:mex="http://schemas.xmlsoap.org/ws/2004/09/mex">
                  <mex:MetadataSection>
                     <mex:MetadataReference>
                        <wsa:Address>https://a-suisseid-idp.ch:7443/mex
                        </wsa:Address>
                     </mex:MetadataReference>
                  </mex:MetadataSection>
               </mex:Metadata>
            </wsa:Metadata>
         </wsa:EndpointReference>
         <ic:UserCredential>
            <ic:DisplayCredentialHint>CSP-0001 SuisseID</ic:DisplayCredentialHint>
            <ic:X509V3Credential>
               <ds:X509Data>
                  <wsse:KeyIdentifier ValueType="http://docs.oasis-open.org/
↳wss/2004/xx/oasis-2004xx-wss-soap-message-security-1.1#ThumbprintSHA1"
                     EncodingType="http://docs.oasis-open.org/wss/2004/01/oasis-↳200401-
wss-soap-message-security-1.0#Base64Binary"
                     xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-↳200401-wss-
wssecurity-secext1.0.xsd">IdigssraDay3mptsjsfieU1Ud=
                  </wsse:KeyIdentifier>
               </ds:X509Data>
            </ic:X509V3Credential>
         </ic:UserCredential>
      </ic:TokenService>
   </ic:TokenServiceList>
   <ic:SupportedTokenTypeList>
      <wst:TokenType xmlns:wst="http://schemas.xmlsoap.org/ws/2005/02/trust">
            urn:oasis:names:tc:SAML:2.0:assertion
      </wst:TokenType>
   </ic:SupportedTokenTypeList>
   <ic:SupportedClaimTypeList>
      <ic:SupportedClaimType
         Uri="http://www.ech.ch/xmlns/eCH-0113/1/identificationValidUntil">
         <ic:DisplayTag>Identification Valid Until</ic:DisplayTag>
         <ic:Description>Date until the identification document is valid.
         </ic:Description>
      </ic:SupportedClaimType>
      …
   </ic:SupportedClaimTypeList>
   <ic:PrivacyNotice Version="1">http://www.a-suisseid-idp.ch/privacy_policy.html
   </ic:PrivacyNotice>
</ic:InformationCard>
```

Element `UserCredential`	The SuisseID certificate and keys are stored on smart card, thus X509V3Credential is set as the user credential.
Element `TokenType`	The only `TokenType` that **MUST** be supported is SAML 2.0.

Element `SupportedClaimType-List`	Contains the SuisseID attributes (see 4.6.3). The attributes `DisplayTag` and `Description` are localized in the SuisseID standard languages.
Element `PrivacyNotice`	Provides the location of the privacy statement of the Identity Provider.

The SuisseID Identity Card is localized in the following languages: German, French, Italian, and English.

5 Best Practices for Certificate Validation

5.1 Validation of SuisseID IAC

5.1.1 Precondition for Validation:

The SP MUST configure at least the root CA of the SuisseID IAC issuing CA of its supported SuisseID providers. Additionally the entire certificate chain(s) MAY be configured for better interoperability.

5.1.2 Validation Algorithm:

1) Basic Path Validation according to RFC5280 (see Certificate Path Validation) with the following particular input:
   - **user-initial-policy-set** (CertPolicyId to check):

   OID=2.16.756.5.26.1.1.2
   - **initial-any-policy-inhibit** (inhibit anyPolicy policy): true
2) CRL check according to RFC5280 (CRL Validation, see Certificate Path Validation) or OSCP check according to RFC2560.

5.2 Validation of SuisseID QC

5.2.1 Precondition for Validation:

The SP MUST configure at least the root CA of the SuisseID QC issuing CA of its supported SuisseID providers. Additionally the entire certificate chain(s) MAY be configured for better interoperability.

5.2.2 Validation Algorithm:

1) Basic Path Validation according to RFC5280 (see Certificate Path Validation) with the following particular input:
   - **user-initial-policy-set** (CertPolicyId to check):

   OID=2.16.756.5.26.1.1.1
   - **initial-any-policy-inhibit** (inhibit anyPolicy policy): true
2) CRL check according to RFC5280 (CRL Validation, see Certificate Path Validation) or OSCP check according to RFC2560.

5.3 Validation of a Claim Assertion

SAML assertions MUST be signed by the core IdP/CAS using an advanced signature.

Until version 1.3 of SuisseID specification

- no specific policy is defined to be contained in the advanced certificate used for assertion signing.
- only the signer certificate is contained in the assertion signature section but not the corresponding certificate chain.

5.3.1 Precondition for Validation:

For each SuisseID provider, the SP MUST either configure the entire certificate chain(s) of the issuing CA of the assertion signer certificate or it MUST configure all possible assertion signer certificates as peer trusts

Cofiguring the assertion signer certificates as peer trusts is not recommended because of operational issues with certificate renewal procedures.

5.3.2 Validation Algorithm (until version 1.3 of SuisseID specification):

1) Basic Path Validation according to RFC5280 (see Certificate Path Validation).
   Hint: see hint section above.
2) CRL check according to RFC5280 (CRL Validation, see Certificate Path Validation) or OSCP check according to RFC2560.
3) **DN check (in case of non-peer-trust)**: limit accepted certificates by certificate DN

5.3.3 Validation Algorithm (after version 1.5 of SuisseID specification):

1) Basic Path Validation according to RFC5280 (see Certificate Path Validation) with the following particular input:
   - **user-initial-policy-set** (CertPolicyId to check):

     OID=2.16.756.5.26.1.1.4
   - **initial-any-policy-inhibit** (inhibit anyPolicy policy): true
2) CRL check according to RFC5280 (CRL Validation, see Certificate Path Validation) or OSCP check according to RFC2560.

5.4 Validation of a QC Signed Attribute

Each QC signed core assertion attribute MUST contain a qualified signature of the CA to increase traceability and trust.

5.4.1 Preconditions for Validation:

The SP MUST configure several trust anchors:

- QC attribute signer issuing CA: For each SuisseID provider, the SP MUST either configure the entire certificate chain(s) of the issuing CA of the QC attribute signer certificate or it MUST configure all possible QC attribute signer certificates as peer trusts.
- QC timestamp token signer issuing CA according to RFC 3161: For each SuisseID provider, the SP MUST either configure the entire certificate chain(s) of the issuing CA of the QC timestamp token signer certificate or it MUST configure all possible QC timestamp token signer certificates as peer trusts.

Cofiguring the QC attibute signer or the QC timestamp token signer certificates as peer trusts is not recommended because of operational issues with certificate renewal procedures.

5.4.2 Validation Algorithm: see also chapter 3.6.3.4.4 Validation Checks

[1] Validate the QC attribute signer certificate

1) Basic Path Validation according to RFC5280 (see Certificate Path Validation) with the following particular input:
   - **user-initial-policy-set** (CertPolicyId to check):

   OID=2.16.756.5.26.1.1.3

   - **initial-any-policy-inhibit** (inhibit anyPolicy policy): true
2) CRL check according to RFC5280 (CRL Validation, see Certificate Path Validation) or OSCP check according to RFC2560.

[2] Verify the XML signature on the attribute using embedded X509Certificate

[3] Validate the QC timestamp token signer certificate

Hint: see preconditions for validation

[4] Validate the XAdES signature timestamp:

1) Check that QualifyingProperties Target references Signature Id
2) Check that HashDataInfo uri references SignatureValue Id
3) Check if message digest of base-64 decoded time stamp token matches the base-64 decoded signature value string.

[5] Optional: verify timestamp on attribute:

Ensure that difference between the timestamp and the embedded signer certificate.notBefore time differ by less than a SP-specific (configurable) maxTimeStampDeltaInSecs.

[6] Verify that the attribute corresponds to the given user:

Check the SuisseID number.

5.5 Certification Path Validation

Certification path processing verifies the binding between the subject distinguished name and/or subject alternative name and subject public key.

RFC5280 describes an algorithm for validating certification paths:

- Basic Path Validation
- CRL Validation

6 Haftungsausschluss/Hinweise auf Rechte Dritter

eCH-Standards, welche der Verein **eCH** dem Benutzer zur unentgeltlichen Nutzung zur Verfügung stellt, oder welche **eCH** referenziert, haben nur den Status von Empfehlungen. Der Verein **eCH** haftet in keinem Fall für Entscheidungen oder Massnahmen, welche der Benutzer auf Grund dieser Dokumente trifft und / oder ergreift. Der Benutzer ist verpflichtet, die Dokumente vor deren Nutzung selbst zu überprüfen und sich gegebenenfalls beraten zu lassen. **eCH**-Standards können und sollen die technische, organisatorische oder juristische Beratung im konkreten Einzelfall nicht ersetzen.

In **eCH**-Standards referenzierte Dokumente, Verfahren, Methoden, Produkte und Standards sind unter Umständen markenrechtlich, urheberrechtlich oder patentrechtlich geschützt. Es liegt in der ausschliesslichen Verantwortlichkeit des Benutzers, sich die allenfalls erforderlichen Rechte bei den jeweils berechtigten Personen und/oder Organisationen zu beschaffen.

Obwohl der Verein **eCH** all seine Sorgfalt darauf verwendet, die **eCH**-Standards sorgfältig auszuarbeiten, kann keine Zusicherung oder Garantie auf Aktualität, Vollständigkeit, Richtigkeit bzw. Fehlerfreiheit der zur Verfügung gestellten Informationen und Dokumente gegeben werden. Der Inhalt von **eCH**-Standards kann jederzeit und ohne Ankündigung geändert werden.

Jede Haftung für Schäden, welche dem Benutzer aus dem Gebrauch der **eCH**-Standards entstehen ist, soweit gesetzlich zulässig, wegbedungen.

7 Urheberrechte

Wer **eCH**-Standards erarbeitet, behält das geistige Eigentum an diesen. Allerdings verpflichtet sich der Erarbeitende sein betreffendes geistiges Eigentum oder seine Rechte an geistigem Eigentum anderer, sofern möglich, den jeweiligen Fachgruppen und dem Verein **eCH** kostenlos zur uneingeschränkten Nutzung und Weiterentwicklung im Rahmen des Vereinszweckes zur Verfügung zu stellen.

Die von den Fachgruppen erarbeiteten Standards können unter Nennung der jeweiligen Urheber von **eCH** unentgeltlich und uneingeschränkt genutzt, weiterverbreitet und weiterentwickelt werden.

eCH-Standards sind vollständig dokumentiert und frei von lizenz- und/oder patentrechtlichen Einschränkungen. Die dazugehörige Dokumentation kann unentgeltlich bezogen werden.

Diese Bestimmungen gelten ausschliesslich für die von **eCH** erarbeiteten Standards, nicht jedoch für Standards oder Produkte Dritter, auf welche in den **eCH**-Standards Bezug genommen wird. Die Standards enthalten die entsprechenden Hinweise auf die Rechte Dritter.

Annex A – XML Schema eCH-0113

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- eCH-0113 Specification SuisseID -->
<xs:schema xmlns:eCH-0113="http://www.ech.ch/xmlns/eCH-0113/1"
xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
xmlns:icc="http://schemas.xmlsoap.org/ws/2005/05/identity/claims" target-
Namespace="http://www.ech.ch/xmlns/eCH-0113/1" elementFormDefault="qualified"
blockDefault="#all" version="2">
   <xs:import namespace="http://www.w3.org/2000/09/xmldsig#" schemaLocation="xmldsig-core-
schema.xsd"/>
   <xs:import namespace="http://schemas.xmlsoap.org/ws/2005/05/identity/claims" schemaLoca-
tion="claims-1.0-os.xsd"/>
   <xs:annotation>
      <xs:documentation xml:lang="en">Issue date: 18.03.2010
      </xs:documentation>
   </xs:annotation>
   <xs:simpleType name="stringMaxLength255MinLength1Type">
      <xs:restriction base="xs:string">
         <xs:maxLength value="255"/>
         <xs:minLength value="1"/>
      </xs:restriction>
   </xs:simpleType>
   <xs:simpleType name="stringMaxLength255Type">
      <xs:restriction base="xs:string">
         <xs:maxLength value="255"/>
      </xs:restriction>
   </xs:simpleType>
   <xs:simpleType name="stringMaxLength9Type">
      <xs:restriction base="xs:string">
         <xs:maxLength value="9"/>
      </xs:restriction>
   </xs:simpleType>
   <xs:simpleType name="stringMaxLength24Type">
      <xs:restriction base="xs:string">
         <xs:maxLength value="24"/>
      </xs:restriction>
   </xs:simpleType>
   <xs:complexType name="givenNamesType">
      <xs:sequence>
         <xs:element name="givenName" type="eCH-0113:stringMaxLength255MinLength1Type"
            minOccurs="1" maxOccurs="20"/>
      </xs:sequence>
   </xs:complexType>
   <xs:simpleType name="countryIdISO3Type">
      <xs:restriction base="xs:token">
         <xs:length value="3"/>
      </xs:restriction>
   </xs:simpleType>
   <xs:complexType name="datePartiallyKnownType">
      <xs:choice>
         <xs:element name="yearMonthDay" type="xs:date"/>
         <xs:element name="yearMonth" type="xs:gYearMonth"/>
         <xs:element name="year" type="xs:gYear"/>
      </xs:choice>
   </xs:complexType>
   <!-- identificationKind claims are serialized as follows: 0-Passport, 1-ID, 2-Stateless
-->
   <xs:simpleType name="identificationKindType">
      <xs:restriction base="xs:token">
         <xs:enumeration value="0"/>
         <xs:enumeration value="1"/>
         <xs:enumeration value="2"/>
      </xs:restriction>
   </xs:simpleType>
   <xs:complexType name="signedAttributeType">
      <xs:sequence>
         <xs:element name="attribute" type="eCH-0113:attributeType"/>
         <xs:element ref="ds:Signature" minOccurs="0"/>
      </xs:sequence>
   </xs:complexType>
   <xs:complexType name="attributeType">
      <xs:annotation>
```

```
         <xs:documentation>
         The actual content model cannot be expressed by means of
         XML Schema 1.0 without violating the
         'Unique Particle Attribution' (UPA) rule, hence wild-card.

         The following shows the intended content model:

         &lt;xs:choice maxOccurs="1" minOccurs="1">
            &lt;xs:element name="givenNames" type="eCH-0113:givenNamesType" />
            &lt;xs:element name="dateOfBirthPartiallyKnown" type="eCH-
0113:datePartiallyKnownType" />
            &lt;xs:element name="placeOfBirth" type="eCH-0113:stringMaxLength255Type" />
            &lt;xs:element name="origin" type="eCH-0113:stringMaxLength255Type" />
            &lt;xs:element name="nationality" type="eCH-0113:countryIdISO3Type" />
            &lt;xs:element name="identificationNumber" type="eCH-0113:stringMaxLength9Type"
/>
            &lt;xs:element name="identificationNumberFull" type="eCH-
0113:stringMaxLength24Type" />
            &lt;xs:element name="identificationKind" type="eCH-0113:identificationKindType"
/>
            &lt;xs:element name="issuingCountry" type="eCH-0113:countryIdISO3Type" />
            &lt;xs:element name="issuingOffice" type="eCH-0113:stringMaxLength255Type" />
            &lt;xs:element name="identificationIssuedOn" type="xs:date" />
            &lt;xs:element name="identificationValidUntil" type="xs:date" />
            &lt;xs:element ref="icc:givenname" />
            &lt;xs:element ref="icc:surname" />
            &lt;xs:element ref="icc:dateofbirth" />
            &lt;xs:element ref="icc:gender" />
            &lt;xs:any namespace='##other' />
         &lt;/xs:choice>

         </xs:documentation>
      </xs:annotation>
      <xs:complexContent>
         <xs:extension base="xs:anyType">
            <xs:attribute name="name" type="xs:string"/>
            <xs:attribute name="suisseIdNo" type="xs:string"/>
            <xs:attribute name="certIssuerDnQc" type="xs:string"/>
            <xs:attribute name="certSerialNoQc" type="xs:integer"/>
            <xs:attribute name="certIssuerDnIac" type="xs:string"/>
            <xs:attribute name="certSerialNoIac" type="xs:integer"/>
            <xs:attribute name="Id" type="xs:ID" use="optional"/>
         </xs:extension>
      </xs:complexContent>
   </xs:complexType>
   <xs:complexType name="suisseIdRecordType">
      <xs:sequence minOccurs="0" maxOccurs="unbounded">
         <xs:element name="signedAttribute" type="eCH-0113:signedAttributeType"/>
      </xs:sequence>
   </xs:complexType>
   <!-- SuisseID record for exchange of all QC signed core claim attributes defined by eCH-
0113 -->
   <xs:element name="suisseIdRecord" type="eCH-0113:suisseIdRecordType"/>
   <!-- SuisseID QC signed core claim elements defined by eCH-0113 -->
   <xs:element name="givenNamesQc" type="eCH-0113:signedAttributeType"/>
   <xs:element name="dateOfBirthPartiallyKnownQc" type="eCH-0113:signedAttributeType"/>
   <xs:element name="placeOfBirthQc" type="eCH-0113:signedAttributeType"/>
   <xs:element name="originQc" type="eCH-0113:signedAttributeType"/>
   <xs:element name="nationalityQc" type="eCH-0113:signedAttributeType"/>
   <xs:element name="identificationNumberQc" type="eCH-0113:signedAttributeType"/>
   <xs:element name="identificationNumberFullQc" type="eCH-0113:signedAttributeType"/>
   <xs:element name="identificationKindQc" type="eCH-0113:signedAttributeType"/>
   <xs:element name="issuingCountryQc" type="eCH-0113:signedAttributeType"/>
   <xs:element name="issuingOfficeQc" type="eCH-0113:signedAttributeType"/>
   <xs:element name="identificationIssuedOnQc" type="eCH-0113:signedAttributeType"/>
   <xs:element name="identificationValidUntilQc" type="eCH-0113:signedAttributeType"/>
   <xs:element name="givenNameQc" type="eCH-0113:signedAttributeType"/>
   <xs:element name="surnameQc" type="eCH-0113:signedAttributeType"/>
   <xs:element name="dateOfBirthQc" type="eCH-0113:signedAttributeType"/>
   <xs:element name="genderQc" type="eCH-0113:signedAttributeType"/>
   <!-- SuisseID plain core claim elements defined by eCH-0113 -->
   <xs:element name="givenNames" type="eCH-0113:givenNamesType"/>
   <xs:element name="dateOfBirthPartiallyKnown" type="eCH-0113:datePartiallyKnownType"/>
   <xs:element name="placeOfBirth" type="eCH-0113:stringMaxLength255Type"/>
```

```
   <xs:element name="origin" type="eCH-0113:stringMaxLength255Type"/>
   <xs:element name="nationality" type="eCH-0113:countryIdISO3Type"/>
   <xs:element name="identificationNumber" type="eCH-0113:stringMaxLength9Type"/>
   <xs:element name="identificationNumberFull" type="eCH-0113:stringMaxLength24Type"/>
   <xs:element name="identificationKind" type="eCH-0113:identificationKindType"/>
   <xs:element name="issuingCountry" type="eCH-0113:countryIdISO3Type"/>
   <xs:element name="issuingOffice" type="eCH-0113:stringMaxLength255Type"/>
   <xs:element name="identificationIssuedOn" type="xs:date"/>
   <xs:element name="identificationValidUntil" type="xs:date"/>
   <xs:element name="organizationName" type="eCH-0113:stringMaxLength255Type"/>
   <xs:element name="title" type="eCH-0113:stringMaxLength255Type"/>
   <!-- SuisseID core claim elements defined by
http://schemas.xmlsoap.org/ws/2005/05/identity/claims -->
   <!--
      <xs:element name="givenname" type="icc:StringMaxLength255MinLength1" />
      <xs:element name="surname" type="icc:StringMaxLength255MinLength1" />
      <xs:element name="dateofbirth" type="xs:date" />
      <xs:element name="gender" type="icc:GenderType" />
      <xs:element name="emailaddress" type="icc:StringMaxLength255MinLength1" />
   -->
   <!-- SuisseID derived core claim elements defined by eCH-0113 -->
   <xs:element name="age" type="xs:unsignedInt"/>
   <xs:element name="isOver16" type="xs:boolean"/>
   <xs:element name="isOver18" type="xs:boolean"/>
   <xs:element name="isSwissCitizen" type="xs:boolean"/>
   <!-- SuisseID derived core claim elements defined by
http://schemas.informationcard.net/@ics -->
   <!--  <xs:element name="age-18-or-over/2008-11" type="xs:token" /> -->
   <!-- SuisseID SAML request extensions -->
   <xs:element name="suisseidRessourceUrl" type="xs:anyURI"/>
   <xs:attribute name="required" type="xs:boolean"/>
   <xs:element name="assertionConsumerServiceUrl" type="xs:anyURI" />
   <!--
      SAML Core 2.0: System entities are free to define more
      specific status codes by defining appropriate URI
      references
   -->
   <!-- http://www.ech.ch/xmlns/eCH-0113/1/abortedByUser -->
</xs:schema>
```

Annex B – References

[1] SR 943.03, ZertES, *Bundesgesetz vom 19. Dezember 2003 über Zertifizierungsdienste im Bereich der elektronischen Signatur*

[2] SR 943.032, VZertES, *Verordnung vom 3. Dezember 2004 über Zertifizierungsdienste im Bereich der elektronischen Signatur*

[3] SR 943.032.1 TAV-ZertES, *Verordnung vom 6. Dezember 2004 des BAKOM über Zertifizierungsdienste im Bereich der elektronischen Signatur, Ausgabe 3 vom 13. November 2006*

[4] SR 837.0, *Bundesgesetz über befristete konjunkturelle Stabilisierungsmassnahmen im Bereich des Arbeitsmarkts und der Informations- und Kommunikationstechnologien vom 25. September 2009*

[5] Common PKI Specification V2.0

[6] RFC 5280 (May 2008), *Internet X.509 Public Key Infrastructure – Certificate and CRL Profile*

[7] RFC 3279 (April 2002), *Algorithms and Identifiers for the Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile*

[8] ITU-T Recommendation X.509 (2000) – ISO 9594-8:2001 (4. Ausgabe), *Information technology – Open systems interconnection – The Directory: Public key and attribute certificate frameworks*

[9] RFC 3739 (März 2004), *Internet X.509 Public Key Infrastructure – Qualified Certificates Profile*

[10] FIPS 140-1 (11.1.94), *Security Requirements for Cryptographic Modules*

[11] FIPS 140-2 (25.5.01), *Security Requirements for Cryptographic Modules*

[12] ISO/IEC 15408: 2005, *Information technology – Security techniques. Evaluation criteria for IT security*

[13] ITSEC Version 1.2 (28. Juni 1991), *Information Technology Security Evaluation Criteria*

[14] RFC 4055 (Juni 2005), *Additional Algorithms and Identifiers for RSA Cryptography for use in the Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile*

[15] RFC 3161 (August 2001), *Internet X.509 Public Key Infrastructure Time-Stamp Protocol (TSP)*

[16] RFC 2119 (März 1997), *Key words for use in RFCs to Indicate Requirement Levels*

[17] Assertions and Protocol for the OASIS Security Assertion Markup Language (SAML) V2.0 (March 2005), Organization for the Advancement of Structured Information Standards.

[18] WS-Trust 1.3 (19.3.2007), Organization for the Advancement of Structured Information Standards.

[19] Identity Metasystem Interoperability 1.0 (1.7.2009), Organization for the Advancement of Structured Information Standards.

[20] ETSI TS 101 903, V1.3.2 (2006-03), *XML Advanced Electronic Signatures (XAdES)*

[21] WS-SecurityPolicy 1.2 (July 2007), *Web Services Security Policy Language, Organization for the Advancement of Structured Information Standards*

[22] Doc 9303, Part 3, Machine Readable Official Travel Documents, Volume 1 (Third Edition – 2008), *MRtds with Machine Readable Data stored in Optical Character Recognition Format', International Civil Aviation Organization*.

[23] OASIS Web Services Security X.509 Certificate Token Profile 1.1 (1 February 2006), Organization for the Advancement of Structured Information Standards.

Annex C – Abbreviations

Abbr.	Full name
ASN.1	Abstract Syntax Notation 1
CA	Certificate Authority
CAI	Claim Assertion Infrastructure
CAS	Claim Assertion Service
CN	Common Name
CSP	Certification Service Provider
DN	Distinguished Name
eCH	Verein für E-Government- und E-Health-Standards für die Schweiz
EIdP	Extended Identity Provider
FIPS	Federal Information Processing Standards
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
IAC	Identification and Authetication Certificate
IAM	Identity and Access Management
IAS	Identity and Authentication Service
ICAO	International Civil Aviation Organization
ID	Identity
IdP	Identity Provider
IKT	Informations- und Kommunikationstechnologien
ITSEC	Information Technology Security Evaluation Criteria
LDAP	Lightweight Directory Access Protocol
OCSP	Online Certificate Status Protocol
OID	Object Identifier
PKI	Public Key Infrastructure
QC	Qualified Certificate
RA	Registration Authority
RDN	Relative Distinguished Name
RFC	Request for Comments
RST	Request Security Token
RSTR	Request Security Token Response
SAML	Security Assertion Markup Language
SECO	Staatssekretariat für Wirtschaft
SP	Service Provider

Abbr.	Full name
SR	Systematische Rechtssammlung
SSCD	Secure Signature Creation Device
SOAP	Simple Object Access Protocol
SSL/TLS	Secure Socket Layer / Transport Layer Security
SSO	Single Sign On
STS	Security Token Service
TAV	Technische und administrative Vorschriften
TSA	Time Stamping Authority
UPN	User Principal Name
URL	Uniform Resource Locator
UTF-8	Unicode Transformation Format-8
WSDL	Web Service Definition Language
WSS	Web Services Security
XAdES	XML Advanced Electronic Signatures
XHTML	Extensible Hypertext Markup Language
XML	Extended Markup Language
ZertES	Bundesgesetz über Zertifizierungsdienste im Bereich der elektronischen Signatur

Annex D – Changes from Version 1.3

The following changes have been made to this document:

- The document has been formatted as an eCH standard
- The attributes organization, e-mail and title are being provided in the IdP as attributes. To support backwards compatibility, these attributes are provided as core assertion attributes only. Implementers **MAY** choose to read the attributes from the authentication certificate provided to the IdP instead of storing the attributes in the IdP database.
- The attributes identificationNumberFull and identificationNumberFullQc are now being provided by the IdP. This attribute is to be implemented both as a plain core assertion attributes and as a QC signed core assertion attributes.
- The issuing procedure of SAML assertions was modified. The advanced certificate used by the IdP/CAS to sign the SAML assertions **MUST** contain a well-defined object identifier in its PolicyInformation and the IdP/CAS has to include the complete chain of the advanced signature certificate into the signed SAML Assertion for certificate path validation.
- Best practices for the validation of a SuisseID have been added