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

Towards pan-European recognition of electronic IDs (eIDs)

ICT PSP call identifier: ICT-PSP/2007/1
ICT PSP Theme/objective identifier: 1.2

Project acronym: STORK

Project full title: Secure Identity Across Borders Linked
Grant agreement no.: 224993

D5.8.3b Interface Specification

Deliverable Id :	**D5.8.3**
Deliverable Name :	**D5.8.3 Technical Design**
Status :	**Final**
Dissemination Level :	**Public**
Due date of deliverable :	**December 31st 2011**
Actual submission date :	**November 11th 2010**
Work Package :	**5.1**
Organisation name of lead contractor for this deliverable :	**ES-MPT**
Author(s):	**Joaquín Alcalde-Moraño, Jorge López Hernández-Ardieta, Adrian Johnston, Daniel Martinez, Bernd Zwattendorfer, Marc Stern, John Heppe**
Partner(s) contributing :	**AT, BE, DE, ES, UK**

Abstract: This document specifies the interfaces between PEPS and V-IDP implementations, the interfaces between the Common functionalities of the PEPS and Member State specific functionalities, and the corresponding VIdP interfaces for middleware countries. Thus it specifies the data (definition and messages) and communication protocols.

Project co-funded by the European Community under the ICT Policy Support Programme

History

Version	*Date*	*Modification reason*	*Modified by*
0.1	26/10/2010	Initial version (D5.8.2b), amended attributeValue type=anyType, and added the schema's chapter.	John Heppe
0.5.3	15/10/2011	Updated version	Ricardo Ferreira, John Heppe, Joaquín Alcalde-Moraño, Marc Stern, Bernd Zwattendorfer, Thomas Kopp
Final 1.0	11/11/2011	Quality review	A v Overeem, R Wannee, S Koppius

Intermediate internal versions, e.g. for quality reviews, have been omitted

Table of contents

List of figures

List of tables

List of abbreviations

The glossary can be accessed at the corporate STORK Website, clicking the following link: http://www.eid-stork.eu/index.php?option=com_smf&Itemid=33&topic=42.0.

For readability, we briefly enumerate to most common acronyms.

<Abbreviation>	<Explanation>
C-PEPS	Citizen Country PEPS: PEPS in the citizen's origin country
C-VIdP	VIdP (in SP-PEPS country) that verifies the citizen's identity. A S-PEPS may forward an authentication request to a C-VIdP to request user authentication.
IdP	ID Provider. An institution, which verifies the citizen's identity, and issues an electronic ID.
MW	MiddleWare. Architecture of the integration of eIDs in Services, with a direct communication between SP and user's PC, without any central server. The term also refers to the piece of software of this architecture that executes on the user's PC.
MS	STORK Member State
PEPS	Pan European Proxy Service or Server
QAA	Quality Authentication Assurance Level
S-PEPS	Service Provider PEPS: PEPS in the Service Provider's country
S-VIdP	VIdP (in Citizen's Country) acting on behalf of the Service Provider. SP-MW calls S-VIdP to request user authentication. S-VIdP will forward request to a C-PEPS if necessary.
SP	Service Provider
SP-MW	Service Provider in a MiddleWare country
SP-PEPS	Service Provider in the PEPS country
SPWare	Piece of software installed at the Service Provider, that complements the MW
VIdP	Virtual IDP. A system component helping to abstract Pan-European eID-interoperability. It either serves as a delegation component between the SP-MW or S-PEPS and the needed SPware (appropriate MW server component) or enables an SP-MW to communicate with other C-PEPS.
WP	Work Package

Executive summary

This document is the interface specification for the STORK platform, the aim of which is to achieve the interoperability of electronic identifiers all over the 14 (+3) participating states.

SAML 2.0 is the chosen messaging standard to be used between the STORK components in each member state. The STORK authentication request and response formats are defined. The STORK protocols (bindings and profiles) used by the STORK components to inter-communicate are also defined.

Communicating information between states requires a shared understanding about what identity attributes are available and what each attribute means. A list of STORK attributes that each country should understand [but not necessarily provide] are defined.

This document is part of the D5.8.3 Technical design, where a more complete summary and introduction are included. Both are based on their D5.8.2 equivalents; please refer to the introduction for a description of changes.

1 Introduction

1.1 Scope and objectives

The scope of this document is to describe the common interfaces to or used by a Pan-European Proxy Servers (PEPS) or Virtual Identity Provider (VIdP) – it does not define internal PEPS or VIdP interfaces.

The objective is to define the interfaces between PEPS and between a PEPS and a Virtual Identity Providers (VIdP). This is to enable any software implementer to implement a PEPS or VIdP that is conformant to the inter-PEPS and PEPS-VIdP protocols, respectively.

The architectures of the reference PEPS and the VIdP are described in another document - D5.8a Software Class Design.

The interfaces are defined from a number of view points:

- Service Provider PEPS (S-PEPS)
- Service Provider's VIdP (S-VIdP).
- Citizen Country Authentication

For completeness the country-specific interfaces to the Reference PEPS as well as the interface between the Reference VIdP and the national SPWares are also documented.

1.2 Methodology

This document was produced by taking the relevant SAML specifications and using the Functional Requirements document as a standards compliant interface that met those Functional Requirements.

Note: As this document is a part of the D5.8.3 Technical Design most of the introductory chapters are specified in that master document.

1.3 Changes in this version

This document – as explained in D5.8.3, the master document – is based on the D5.8.2 equivalent. The main changes in this version are the following ones:

- Several corrections, updated the check on length of spApplication, spSector and SPID
- Amended with <minor> in the signedDoc attribute
- Allowed EU as a countrycode in spCountry
- Corrected namespace of citizenCoutryCode
- Included a paragraph in the introduction on version numbering
- Eliminated chapter 4, the interface of the common software with specific modules. This chapter didn't specify the interoperability layer.
- Included the chapter 7 for version control
- Corrected global schema
- Set AttributeValue type to "xs:anyType"

- Included error codes

1.4 Version numbering

The version number in the document is also the version number of the corresponding software which implements these specs. Please note that some versions of the document may not result in new versions of the software.

This version number is fixed in the software and will be extracted when producing the version control file. Thus all participants of STORK may know the compatibility of installed software with projected future versions.

Thus the protocol number is: **0.5.3.**

2 Process Flows

This chapter contains a high-level description showing the external interfaces between STORK components and how they fit into the process flows.

2.1 Service Provider PEPS (S-PEPS)

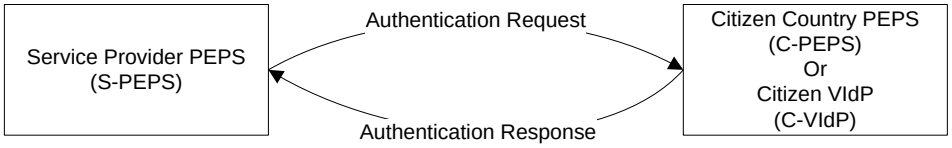

Figure 1: Service Provider PEPS authentication request/response

2.2 Service Provider Virtual IdP (S-VIdP)

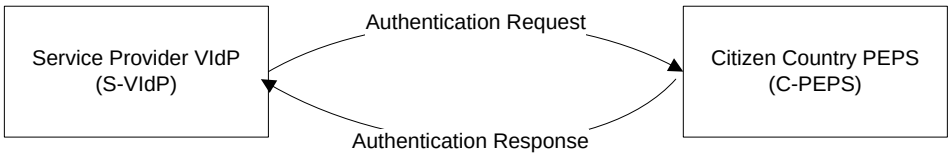

Figure 2: Service Provider VIdP authentication request/response

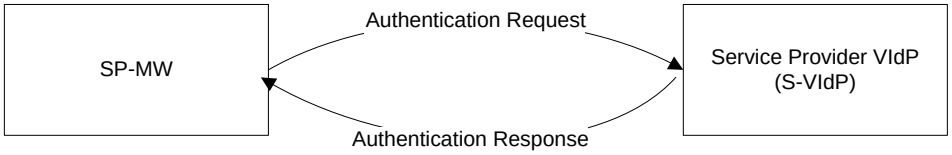

Figure 3: SP-MW authentication request/response

2.3 Digital Certificate Validation

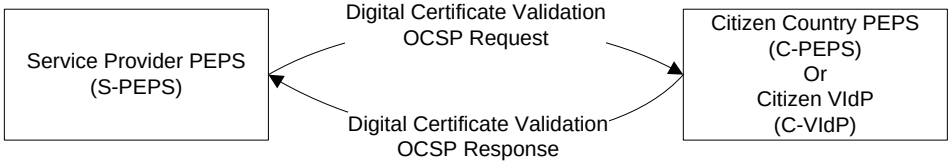

Figure 4: S-PEPS Digital Certificate Validation

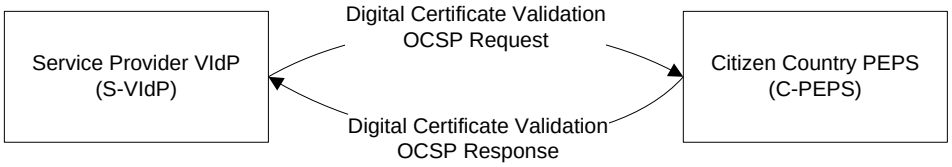

Figure 5: S-VIdP Digital Certificate Validation

3 Inter PEPS Interfaces

This chapter defines the interfaces between Pan-European Proxy Servers (PEPS) and between PEPS and Virtual Identity Providers (VIdP).

The interfaces use standard STORK SAML 2.0 authentication request and response formats[2] & [3]. These are defined in a subsequent chapter. This chapter concentrates on the SAML 2.0 bindings and profiles used in the interfaces.

This chapter also defines the digital certificate validation interfaces.

3.1 Service Provider PEPS (S-PEPS) Authentication Request

3.1.1 Overview

The Service Provider PEPS (S-PEPS) will allow the user to choose which citizen country PEPS (or VIdP) the user wishes to authenticate via. The S-PEPS will construct an Authentication Request for the selected Citizen Country PEPS (or VIdP) in order to obtain the authentication of the user.

The S-PEPS will generate a SAML 2.0 Authentication Request conforming to the SAML 2.0 specification [3] see 5.1. Communication between the citizen's browser and the S-PEPS must be via SSL V3+ or TLS 1.0+.

3.1.2 C-PEPS Request Invocation Method

The only SAML 2.0 profiles and bindings accepted between STORK PEPS are:

- HTTP Post Binding[4]
- Web Browser SSO Profile[5] (N.B. STORK only supports a limited sub-set, see below)
- Holder of Key Web Browser SSO Profile[6] (as a supplement to the Web Browser SSO Profile)

The SAML 2.0 Authentication Request is sent from the S-PEPS to the C-PEPS using the HTTP POST binding e.g.

```
<form action="https://authenticate.C-PEPS.gov.xx/SP-request.aspx "
method="post">
  <input type="hidden" name="SAMLRequest value="[Base64 encoded
Authentication Request]" />
  <input type="hidden" name="RelayState" value="State information to be
persisted across operation" />
</form>.
```

Note: RelayState may be used by the requestor to store an opaque reference to the state of the S-PEPS. It must not exceed 80 characters in length and should have in-built integrity checking. If present, must be persisted by the C-PEPS and returned to the S-PEPS in the Response.

3.1.3 C-VIdP Request Invocation Method

The only SAML 2.0 profiles and bindings accepted between a STORK PEPS and its Virtual IdP are:

- HTTP Post Binding[4]
- Web Browser SSO Profile[5] (N.B. STORK only supports a limited sub-set, see below)

- Holder of Key Web Browser SSO Profile[6] (as a supplement to the Web Browser SSO Profile)

The SAML 2.0 Authentication Request is sent from the S-PEPS to the C-VIdP using the HTTP POST binding e.g.

```
<form action="https://authenticate.VIdP.gov.xx/SP-request.aspx "
method="post">
  <input type="hidden" name="SAMLRequest value="[Base64 encoded
Authentication Request]" />
  <input type="hidden" name="RelayState" value="State information to be
persisted across operation" />
</form>.
```

Note: RelayState may be used by the requestor to store an opaque reference to the state of the S-PEPS. It must not exceed 80 characters in length and should have in-built integrity checking. If present, must be persisted by the C-VIdP and returned to the S-PEPS in the Response.

3.2 Service Provider VIdP (S-VIdP) Authentication Request

3.2.1 Overview

To verify a citizen's identity in middleware countries, the service provider does not interact directly with the Citizen Country PEPS. Instead it will redirect the user to a VIdP (usually in the citizen's country) which forwards the request to its PEPS connector (see deliverable D5.8.2a for details). This PEPS connector interacts with the Citizen Country PEPS.

For scalability and trust reasons, it is intended to route such S-VIdP requests to the corresponding C-VIdP in the citizen's country. In this case, the C-VIdP communicates with the C-PEPS and thus direct communication between S-VIdP and C-PEPS is avoided.

3.2.2 C-PEPS Request Invocation Method

The only SAML 2.0 profiles and bindings accepted between a VIdP and a Citizen Country PEPS are:

- HTTP Post Binding[4]
- Web Browser SSO Profile[5] (N.B. STORK only supports a limited sub-set, see below)
- Holder of Key Web Browser SSO Profile[6] (as a supplement to the Web Browser SSO Profile)

The SAML 2.0 Authentication Request is sent from the S-VIdP to the C-PEPS using the HTTP POST binding e.g.

```
<form action="https://authenticate.C-PEPS.gov.xx/SP-request.aspx "
method="post">
  <input type="hidden" name="SAMLRequest value="[Base64 encoded
Authentication Request]" />
  <input type="hidden" name="RelayState" value="State information to be
persisted across operation" />
</form>.
```

Note: RelayState may be used by the requestor to store an opaque reference to the state of the S-VIdP. It must not exceed 80 characters in length and should have in-built integrity checking. If present, must be persisted by the C-PEPS and returned to the S-VIdP in the Response.

3.3 SP-MW Authentication Request

3.3.1 Overview

Using middleware, a service provider directly communicates with a S-VIdP. Depending on the citizen's nationality the request is forwarded to the appropriate national SPWare. In case the citizen is not using middleware for authentication, the request will be forwarded to the PEPS connector and the desired C-PEPS (This case is covered in section 3.2.2).

3.3.2 V-IDP Request Invocation Method

The only SAML 2.0 profiles and bindings accepted between a SP-MW and a V-IDP are:

- HTTP Post Binding [4]
- SOAP Binding [4]
- Web Browser SSO Profile (N.B. STORK only supports a limited sub-set, see below)
- Holder of Key Web Browser SSO Profile [6] (as a supplement to the Web Browser SSO Profile)

The SAML 2.0 Authentication Request is sent from the SP-MW to the S-VIdP using either the HTTP POST binding or the SOAP Binding. In case of the HTTP POST Binding, the HTML page includes a form field like:

```
<form action="https://authenticate.VIdP .gov.xx/SP-request.aspx "
method="post">
  <input type="hidden" name="SAMLRequest value="[Base64 encoded
Authentication Request]" />
  <input type="hidden" name="RelayState" value="State information to be
persisted across operation" />
</form>.
```

Note: RelayState may be used by the requestor to store an opaque reference to the state of the SP-MW. It must not exceed 80 characters in length and should have in-built integrity checking. If present, must be persisted by the VIdP and returned to the SP-MW in the Response.

3.4 Citizen Country Authentication Response

3.4.1 Overview

The Citizen Country PEPS (C-PEPS) will receive a S-PEPS or S-VIdP Authentication Request. A S-VIdP will receive a SP-MW Authentication Request, whereas a C-VIdP will receive a S-PEPS Authentication Request.

Where a citizen country has no PEPS (and the citizen's identity is verified via middleware) the S-PEPS does not interact directly with the citizen to verify their identity. Instead it will redirect the user to a VIdP (usually in the service provider's country) which acts as a pseudo-PEPS (for the citizen country) and interacts with the citizen via middleware. In the case where the request comes from a Middleware Service Provider (SP-MW) directly, the SP-MW will communicate with the S-VIdP which may forward the request to a C-PEPS if necessary.

The C-PEPS will interface with the selected Identity Provider using an Identity Provider specific protocol. How the C-PEPS allows the user to select an IdP and how the C-PEPS interfaces with its local Identity Providers is MS specific. Additionally the C-PEPS may obtain additional attributes from in-country Attribute Providers and/or derive attribute values based on attributes received.

The C-VIdP/S-VIdP will interface with the citizen's authentication token (smart card, digital certificate, etc.) using token specific and national protocols.

The C-PEPS (or C-VIdP/S-VIdP) will construct an Authentication Response based on the Authentication Request and the information received from the user's selected Identity Provider / Authentication Token and optional Attribute Providers. If required the C-PEPS will obtain the user's consent to request and/or forward the information requested by the S-PEPS (or S-VidP). The same process flow can occur between a requesting SP-MW and a S-VIdP.

The C-PEPS (or C-VIdP) in the PEPS scenario and the S-VIdP in the middleware scenario will generate a SAML 2.0 Authentication Response conforming to the SAML 2.0 specification [3] (see 5.2). Communication between the citizen's browser and the C-PEPS/C-VIdP must be via SSL V3+ or TLS 1.0+.

3.4.2 Citizen Country PEPS (C-PEPS) Response Invocation Method

The STORK Profile for a PEPS Authentication Response is as shown below.

The only SAML 2.0 profiles and bindings accepted between STORK PEPSs and between a PEPS and a VIdP are:

- HTTP Post Binding[4]
- Web Browser SSO Profile[5] (N.B. STORK only supports a limited sub-set, see below)
- Holder of Key Web Browser SSO Profile[6] (as a supplement to the Web Browser SSO Profile)

The SAML 2.0 Authentication Response is sent from the C-PEPS to the S-PEPS or to the S-VIdP using the HTTP POST binding e.g.

```
<form action=" https://www.S-PEPS.gov.xx/consume_assertion "
method="post">
  <input type="hidden" name="SAMLResponse value="[Base64 encoded
Authentication Response]" />
  <input type="hidden" name="RelayState" value="State information
persisted across operation" />
</form>.
```

Note: If RelayState was present in the initial request then it must be persisted by the C-PEPS and returned unaltered to the S-PEPS (or S-VIdP) in the Response. It is an opaque reference to the relay state of the S-PEPS (or S-VIdP). It must not exceed 80 characters in length and should have in-built integrity checking.

3.4.3 VIdP Response Invocation Method

The STORK Profile for a VIdP Authentication Response is as shown below.

The only SAML 2.0 profiles and bindings accepted between a VIdP and a STORK PEPS are:

- HTTP Post Binding[4]

- Web Browser SSO Profile[5] (N.B. STORK only supports a limited sub-set, see below)
- Holder of Key Web Browser SSO Profile[6] (as a supplement to the Web Browser SSO Profile)

If the authentication functions of a VIdP are being invoked by a SP-MW then the SOAP Binding[4] can be used as well.

The SAML 2.0 Authentication Response is sent from the C-VIdP to the S-PEPS using the HTTP POST binding

```
<form action=" https://www.S-PEPS.gov.EU/consume_assertion "
method="post">
  <input type="hidden" name="SAMLResponse value="[Base64 encoded
Authentication Response]" />
  <input type="hidden" name="RelayState" value="State information
persisted across operation" />
</form>.
```

Note: If RelayState was present in the initial request then it must be persisted by the C-VIdP and returned unaltered to the S-PEPS in the Response. It is an opaque reference to the relay state of the S-PEPS. It must not exceed 80 characters in length and should have in-built integrity checking.

3.5 Digital Certificate Validation Message.

The initial implementation of STORK will use OCSP [RFC 2560] as the digital certificate validation protocol. Future implementation may look at interfaces such as W3C XML Key Management Specification [XKMS].

The standard does not restrict the transport protocol. Many options can be used (i.e. HTTP, SMTP, LDAP, etc.). In this version of STORK we will support HTTP only.

Next subsections give the details respecting the OCSP requests and responses to be exchanged between the PEPSs and VIdPs. Subsection 3.5.1 defines the OCSP request message format, while subsection 3.6 focuses on the OCSP response message format.

It is important to remark that requirements, constraints and information herein given are intended for the OCSP messages exchanged between the S-PEPS and the C-PEPS, and in any case restricts the protocol to be used with national IdPs.

Currently, certificate validation using OCSP will not be supported by VIdP.

3.5.1 OCSP Request

A S-PEPS may issue a validation request to a C-PEPS or C-VIdP to verify that the certificate used to authenticate and/or sign an object is still valid and has not been revoked. A S-VIdP may issue a validation request to a C-PEPS to verify that the certificate used to authenticate and/or sign an object is still valid and has not been revoked. Although the OCSP protocol allows the requestor to include validation requests for several certificates in the same OCSP request, S-PEPSs and S-VIdPs are expected to require just one certificate validation at the same time (same request).

The particularities of the validation process itself are country specific.

According to RFC 2560, the OCSP request is ASN.1definition is the next:

```
OCSPRequest     ::=     SEQUENCE {
       tbsRequest                  TBSRequest,
       optionalSignature   [0]     EXPLICIT Signature OPTIONAL }
```

```
TBSRequest      ::=     SEQUENCE {
    version             [0]     EXPLICIT Version DEFAULT v1,
    requestorName       [1]     EXPLICIT GeneralName OPTIONAL,
    requestList                 SEQUENCE OF Request,
    requestExtensions   [2]     EXPLICIT Extensions OPTIONAL }

Signature       ::=     SEQUENCE {
    signatureAlgorithm      AlgorithmIdentifier,
    signature               BIT STRING,
    certs               [0] EXPLICIT SEQUENCE OF Certificate
OPTIONAL}

Version         ::=             INTEGER  {  v1(0) }

Request         ::=     SEQUENCE {
    reqCert                     CertID,
    singleRequestExtensions     [0] EXPLICIT Extensions OPTIONAL }

CertID          ::=     SEQUENCE {
    hashAlgorithm       AlgorithmIdentifier,
    issuerNameHash      OCTET STRING, -- Hash of Issuer's DN
    issuerKeyHash       OCTET STRING, -- Hash of Issuers public key
    serialNumber        CertificateSerialNumber }
```

Most relevant fields are commented next:

- *optionalSignature*

  This field will be MANDATORY in order to protect the request from unauthorised modifications and to authenticate the S-PEPS or S-VIdP. An SSL/TLS channel has to be established between the S-PEPS (or S-VIdP) and the C-PEPS. However, once the request reaches the C-PEPS, it could be modified if not signed. Moreover, audits may require the source of the data stored in the logs be authenticated and its integrity verified. It can be assured by providing a message level protection.

- *signatureAlgoritthm*

  Signature algorithm sha256withrsaencryption (OID 1.2.840.113549.1.1.11) must be used.

  RFC 2560 indicates that OCSP responders shall support the SHA1 hashing algorithm. However, using a stronger algorithm provides higher assurance for a long term. Besides, every PEPS implementation will include the cryptographic provider needed to support such algorithm (please refer to[1]), so no problem is expected in this sense.

- *requestorName*

  The requestor name must correspond to the subject distinguished name included in the certificate of the S-PEPS or S-VIdP.

- *requestExtensions*

  Next extension, according to RFC 2560 – 4.4 Extensions, should be added.

```
id-pkix-ocsp-nonce      OBJECT IDENTIFIER ::= { id-pkix-ocsp 2 }
critical ::= FALSE
extnValue ::= OCTECT STRING (value of the nonce)
```

The nonce prevents reply attacks by cryptographically binding a request and a response.

- *certs*

  The certificate of the S-PEPS or S-VIdP may be included in this field. In that case, the C-PEPS or C-VIdP must check that the certificate corresponds to the one included in the trusted keystore (see [1] for further details).

- *hashAlgorithm*

  Hash algorithm sha256 (OID 2.16.840.1.101.3.4.2.1) must be used for calculating the issuer name hash and the issuer public key hash fields of the *reqCert* field,

3.6 OCSP Response

The ASN.1 definition of an OCSP response message, in accordance with the standard RFC 2560, is the following:

```
OCSPResponse ::= SEQUENCE {
      responseStatus         OCSPResponseStatus,
      responseBytes          [0] EXPLICIT ResponseBytes OPTIONAL }

   OCSPResponseStatus ::= ENUMERATED {
       successful             (0),  --Response has valid confirmations
       malformedRequest       (1),  --Illegal confirmation request
       internalError          (2),  --Internal error in issuer
       tryLater               (3),  --Try again later
                                    --(4) is not used
       sigRequired            (5),  --Must sign the request
       unauthorized           (6)   --Request unauthorized
   }

   ResponseBytes ::=       SEQUENCE {
       responseType   OBJECT IDENTIFIER,
       response       OCTET STRING }
```

The responseBytes field must correspond to a BasicOCSPResponse response type. Therefore, the OID to include in the responseType field is the next:

```
   id-pkix-ocsp           OBJECT IDENTIFIER ::= { id-ad-ocsp }
   id-pkix-ocsp-basic     OBJECT IDENTIFIER ::= { id-pkix-ocsp 1 }
```

The basic response ASN.1 definition is the following:

```
   BasicOCSPResponse       ::= SEQUENCE {
      tbsResponseData      ResponseData,
      signatureAlgorithm   AlgorithmIdentifier,
      signature            BIT STRING,
      certs                [0] EXPLICIT SEQUENCE OF Certificate OPTIONAL
}

   ResponseData ::= SEQUENCE {
```

```
    version              [0] EXPLICIT Version DEFAULT v1,
    responderID              ResponderID,
    producedAt               GeneralizedTime,
    responses                SEQUENCE OF SingleResponse,
    responseExtensions   [1] EXPLICIT Extensions OPTIONAL }

ResponderID ::= CHOICE {
    byName               [1] Name,
    byKey                [2] KeyHash }

SingleResponse ::= SEQUENCE {
    certID                       CertID,
    certStatus                   CertStatus,
    thisUpdate                   GeneralizedTime,
    nextUpdate         [0]       EXPLICIT GeneralizedTime OPTIONAL,
    singleExtensions   [1]       EXPLICIT Extensions OPTIONAL }

CertStatus ::= CHOICE {
     good        [0]     IMPLICIT NULL,
     revoked     [1]     IMPLICIT RevokedInfo,
     unknown     [2]     IMPLICIT UnknownInfo }

RevokedInfo ::= SEQUENCE {
     revocationTime              GeneralizedTime,
     revocationReason    [0]     EXPLICIT CRLReason OPTIONAL }

UnknownInfo ::= NULL -- this can be replaced with an enumeration
```

The OCSP response will be generated either by the C-VIdP/C-PEPS or the national IdP. In any case, the C-PEPS (or C-VIdP) must sign the response before sending it to the S-PEPS (or S-VIdP). As a consequence, if the OCSP response is generated outside the C-PEPS/C-VIdP, the signature of the national IdP must be replaced by the C-PEPS/C-VIdP signature. If not, the S-PEPS/S-VIdP will not be able to trust the response (see [1] for further details on the Circle of Trust). However, it is important to remark that the S-PEPS/S-VIdP must not modify any data included in the response by the national IdP, except the *signature*, *signatureAlgorithm*, *certs*, *byName* and *singleExtensions* fields.

Most relevant fields are commented next:

- *signatureAlgorithm*

  Signature algorithm sha256withrsaencryption (OID 1.2.840.113549.1.1.11) must be used.

- *certs*

  The certificate of the C-PEPS/C-VIdP may be included in this field. In that case, the S-PEPS/S-VIdP must check that the certificate corresponds to the one included in the trusted keystore (see [1] for further details).

- *byName*

  byName field of *ResponderID* type definition must be chosen for specifying the identifier of the responder. This field must correspond to the subject distinguished name included in the C-PEPS OCSP Responder certificate (and possibly included in the *certs* field).

- *responseExtensions*

  The nonce extension should be added here. Its value must correspond to the nonce received in the request.

```
id-pkix-ocsp-nonce      OBJECT IDENTIFIER ::= { id-pkix-ocsp 2 }
critical ::= FALSE
extnValue ::= OCTECT STRING (value of the nonce)
```

4 VIdP – National SPWare specific interfaces

A VIdP encapsulates national SPWares by providing a common interface. This interface can be Web service based and could be implemented by the SPWare. All messages exchanged via Web Service between VIdP and SPWare are defined in this section.

4.1 Overview

A VIdP has received an authentication request from a SP-MW or S-PEPS. The VIdP has to tell the appropriate SPWare (depending on the citizen's country) to start the authentication process. How the VIdP selects the according SPWare is out of scope of this interface specification. To start the process, the VIdP forwards the authentication request message received from a SP-MW or S-PEPS and sends it to the SPWare. The SPWare processes the authentication request and handles necessary user interactions. Afterwards the SPWare returns an appropriate response message including an identifier and additional attribute information back to the VIdP. The VIdP forwards this information to the requesting SP-MW or S-PEPS (In case of a PEPS scenario, the S-PEPS in turn hands the information over to the SP). Details of this process flow can be found in Deliverable 5.8a.

The interface between a VIdP and a SPWare is based on SAML 2.0[3]. The following SAML components are supported in case of Web Service invocation:

- SAML Authentication Request Protocol [2]
- SAML SOAP Binding [4]

Figure 7 illustrates the interface on a high level view. The SAML protocol messages between those two entities are exchanged via a SOAP Web service interface.

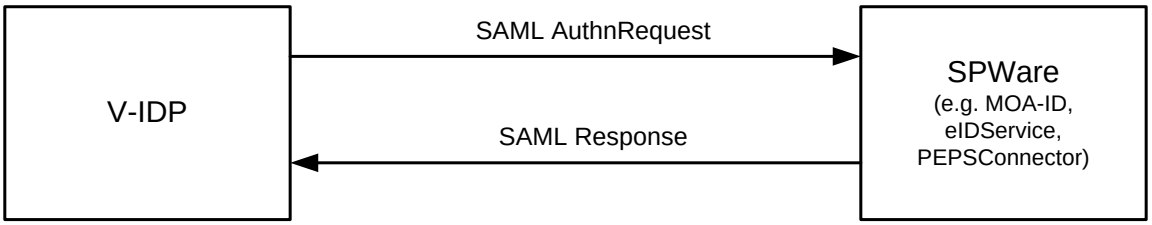

Figure 6: SAML via SOAP between VIdP and SPWare

The detailed authentication process requires additional interactions between the VIdP and the SPWare, hence the interface actually consists of two Web service interfaces. Figure 7 outlines these two interfaces and the exchanged messages. These messages are described in more detail in the next sub-sections.

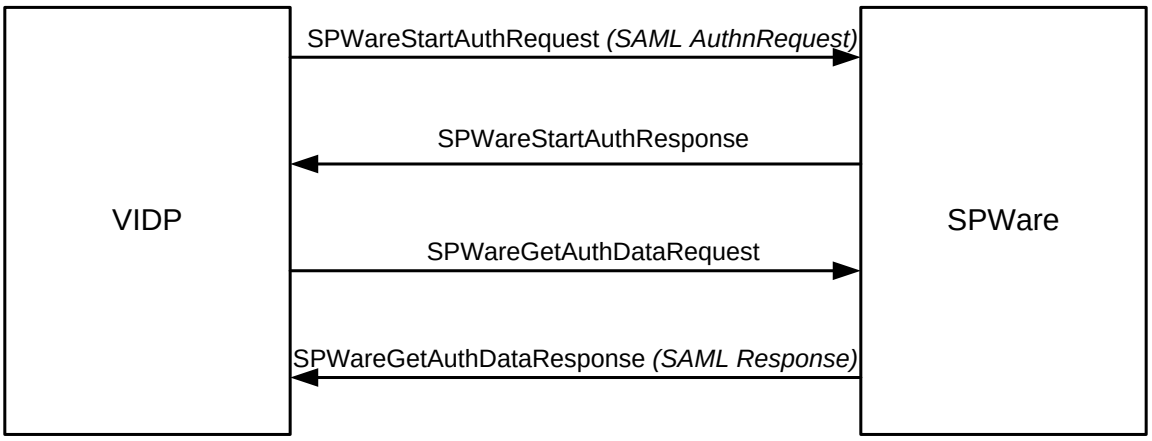

Figure 7: VIdP- SPWare Web service interfaces

4.2 SPWareStartAuthRequest (SAML AuthnRequest)

A VIdP is intended to be an entity operating at a low level of logical complexity. Hence, the VIdP should only delegate/forward the SAML authentication request to the underlying SPWare depending on the provided citizen country code. Due to the fact that the <saml:AuthnRequest> sent from a VIdP to a SPWare is the same request as sent from a SP-MW/S-PEPS to a VIdP, these messages are not described in this section. For details on this request we refer to section 5.1.

4.2.1 Sample SPWareStartAuthRequest (SAML AuthnRequest)

See section 5.1 for a sample <saml:AuthnRequest>.

4.3 SPWareStartAuthResponse

This message defines the return message of the first Web service interface. It is sent from the SPWare – wrapped in a SOAP message - back to the requesting VIdP. This response message is necessary for the total authentication process and is beyond the scope of SAML. It is defined by a separate XML schema. The next sub-sections describe the including elements and attributes in more detail.

Attribute	Required	Type	Allowable Values	Notes
Version	Mandatory	String	1.0	Defines the version of this VIdP message.

Table 1: Attributes of <vidp:VIDPStartAuthResponse>

4.3.1 <vidp:HTTPStatusCode>

Required: Mandatory

Base Type: Integer

This element contains the HTTP status code of the HTTP Response message returned from the SPWare.

4.3.2 <vidp:Base64Content>

Required: Mandatory

Base Type: Base64 encoded byte[]

This element includes the HTML page returned from the SPWare. The HTML page will be transformed into a byte array which will be base64 encoded for transmission.

4.3.3 <vidp:HTTPHeaders>

Required: Optional

This element contains several HTTPHeader elements (HTTP Headers mapped to XML elements), which have been set by the SPWare.

4.3.4 <vidp:HTTPHeader>

Required: Optional

Base Type: String

Attribute	Required	Type	Allowable Values	Notes
Name	Mandatory	String	All HTTP Headers	Defines the name of the HTTP Header to set.

Table 2: Attributes of <vidp: HTTPHeader>

The contents of this element covers the value of the HTTP Header.

4.3.5 <vidp:Extension>

Required: Optional

Base Type: any

This element is intended as extension point in case any additional information has to be included in this message.

4.3.6 Sample SPWareStartAuthResponse

The following code sample is informative and intended to clarify the use of the definitions above. In case of any inconsistencies between the sample code presented below and the defining sections above, those have priority.

```
<vidp:SPWareStartAuthResponse xmlns:vidp="urn:STORK:V-IDP:1.0"
Version="1.0">
<vidp:HTTPStatusCode>200</vidp:HTTPStatusCode >
<vidp:Base64Content>SFRNTCBDb250ZW50…</vidp:Base64Content>
<vidp:HTTPHeaders>
      <vidp:HTTPHeader Name=”Cache-Control”>no-cache</vidp:HTTPHeader>
</vidp: HTTPHeaders >
</vidp:SPWareStartAuthResponse>
```

4.4 SPWareGetAuthDataRequest

This message is sent from the VIdP to the appropriate SPWare to retrieve the final authentication data. To address the correct citizen, the citizen’s session id is included in this request.

Attribute	Required	Type	Allowable Values	Notes
Version	Mandatory	String	1.0	Defines the version of this VIdPmessage.

Table 3: Attributes of <vidp:SPWareGetAuthDataRequest>

4.4.1 <vidp:SessionID>

Required: Mandatory

Base Type: String

This element contains the session id previously established between the SPWare and the citizen.

4.4.2 <vidp:Extension>

See section 4.3.5 for details.

4.4.3 Sample SPWareGetAuthDataRequest

The following code sample is informative and intended to clarify the use of the definitions above. In case of any inconsistencies between the sample code presented below and the defining sections above, those have priority.

```
<vidp:SPWareGetAuthDataRequest xmlns:vidp="urn:STORK:V-IDP:1.0"
Version="1.0">
<vidp:SessionID>edb0e8665db4e9042fe0176a89aade16</vidp:SessionID>
</vidp:SPWareGetAuthDataRequest>
```

4.5 SPWareGetAuthDataResponse (SAML Response)

The VIdP receives this response message from the appropriate SPWare. Since the VIdP usually does not need to modify or alter this message, the VIdP directly forwards this message to the requesting SP-MW or S-PEPS. Because of this, the <saml:Response> message sent from the SPWare to the V-IDP is identical with the message sent from the VIdP to the authentication requesting SP-MW/S-PEPS. For details on this response we refer to section 5.2.

4.5.1 Sample SPWareGetAuthDataResponse (SAML Response)

See section 5.2 for a sample <saml:Response>.

5 SAML 2.0 Authentication Request and Response

The interactions between the major STORK components (i.e. PEPS and VIdP) will be based on the SAML2.0 specifications. The SAML 2.0 Authentication Request and Authentication Response definitions will be used.

Note: The XML Schema definitions in the following sub-sections are taken from the SAML 2.0 specifications. To meet the STORK requirements, some of those definitions need to be adapted which is illustrated in the corresponding tables or sub-sections of the described XML element.

Additional namespaces to be used and their corresponding prefix:

xmlns:stork="urn:eu:stork:names:tc:STORK:1.0:assertion"

xmlns:storkp="urn:eu:stork:names:tc:STORK:1.0:protocol"

5.1 Authentication Request

The SAML 2.0 authentication request specification will be used to request the authentication of the citizen from either a PEPS or a VIdP. The extensions element of the <samlp:AuthnRequest> has been used to allow additional STORK attributes to be requested at authentication time. The proposed use of a child <storkp:RequestedAttributes> element with child <stork:RequestedAttribute> similar to the <md:RequestedAttribute> from metadata has been used. The extension element also allows the transport of additional information required for completing the authentication process (necessary for VIdP authentication). An authentication request may not be larger than 128k bytes

5.1.1 <samlp:AuthnRequest>

```
<sequence>
            <element ref="saml:Issuer" minOccurs="0"/>
            <element ref="ds:Signature" minOccurs="0"/>
            <element ref="samlp:Extensions" minOccurs="0"/>
             <element ref="saml:Subject" minOccurs="0"/>
             <element ref="samlp:NameIDPolicy" minOccurs="0"/>
             <element ref="saml:Conditions" minOccurs="0"/>
             <element ref="samlp:RequestedAuthnContext" minOccurs="0"/>
             <element ref="samlp:Scoping" minOccurs="0"/>
</sequence>
<attribute name="ID" type="ID" use="required"/>
<attribute name="Version" type="string" use="required"/>
<attribute name="IssueInstant" type="dateTime" use="required"/>
<attribute name="Destination" type="anyURI" use="optional"/>
<attribute name="Consent" type="anyURI" use="optional"/>
<attribute name="ForceAuthn" type="boolean" use="optional"/>
<attribute name="IsPassive" type="boolean" use="optional"/>
<attribute name="ProtocolBinding" type="anyURI" use="optional"/>
<attribute name="AssertionConsumerServiceIndex" type="unsignedShort"
use="optional"/>
 <attribute name="AssertionConsumerServiceURL" type="anyURI"
use="optional"/>
<attribute name="AttributeConsumingServiceIndex" type="unsignedShort"
use="optional"/>
<attribute name="ProviderName" type="string" use="optional"/>
```

Attribute	Required	Allowable Values	Notes
ID	Mandatory		Encoded value between 128-160 bits long.
Version	Mandatory	2.0	Version of SAML
IssueInstant	Mandatory		UTC Date & time request was issued.
Destination	Optional		URI reference of the SAML Request processor this request is being sent to. This is mandatory if the HTTP-Post Binding is used.
Consent	Optional	urn:oasis:names:tc:SAML:2.0:consent:unspecified	
ForceAuthn	Mandatory	true	The user must be actively authenticated by the IdP.
IsPassive	Mandatory	false	A passive authentication is not permitted.
ProtocolBinding	Mandatory	urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST urn:oasis:names:tc:SAML:2.0:bindings:SOAP	Currently only HTTP-Post binding is supported for inter PEPS and between PEPS and VIdP. The SOAP binding is only supported for direct communication between SP-MW and VIdP.
AssertionConsumerServiceIndex	Not Used		This is unsupported and its use will result in an urn:oasis:names:tc:SAML:2.0:status:RequestUnsupported
AssertionConsumerServiceURL	Mandatory		URL to which a Authentication Response must be sent. This must be via a secure SSL connection i.e. Https
AttributeConsumingServiceIndex	Not Used		This is unsupported and its use will result in an urn:oasis:names:tc:SAML:2.0:status:RequestUnsupported
ProviderName	Mandatory	UTF-8, maximum 128 characters	Human readable name of the original service provider requesting the authentication.

Table 4: SAML Attributes of an Authentication Request

5.1.2 <saml:Issuer>

```
<element name="Issuer" type="saml:NameIDType"/>
<complexType name="NameIDType">
    <simpleContent>
        <extension base="string">
            <attributeGroup ref="saml:IDNameQualifiers"/>
            <attribute name="Format" type="anyURI" use="optional"/>
            <attribute name="SPProvidedID" type="string" use="optional"/
>
        </extension>
```

```
    </simpleContent>
</complexType>
```

Required: Mandatory

The <Issuer> element contains a URI that identifies the issuing PEPS and must be mutually agreed with the receiving PEPS.

Attribute	Required	Allowable Values	Notes
NameQualifier	Not used		The security domain that qualifies that name.
SPNameQualifier	Not used		Qualifying the name with a name of a service provider.
Format	Optional	urn:oasis:names:tc:SAML:2.0:nameid-format:entity	URI representing the classification of the identifier. Default is urn:oasis:names:tc:SAML:2.0:nameid-format:entity.
SPProvidedID	Not used		Name identifier if different from the name in the contents of the element.

Table 5: SAML:Issuer Attributes

5.1.3 <ds:Signature>

Required: Optional

If the HTTP POST Binding is used, the authentication request must be signed. In case of the SOAP binding similar techniques like SSL/TLS can be used instead. The SOAP binding is only supported for the communication between SP-MW and S-VIdP.

An XML Signature authenticates the requestor (S-PEPS, S-VIdP or SP-MW) and ensures message integrity (signature over complete request).

The signature must be an enveloped signature and applied to the <samlp:AuthnRequest> element and all its children. The signature must contain a single <ds:Reference> containing the ID attribute value of the <samlp:AuthnRequest> element.

<ds:Signature> is defined in http://www.w3.org/TR/xmldsig-core/.

5.1.4 <samlp:Extensions>

Required: Mandatory

This element contains extension to the standard SAML 2.0 Authentication Request. In STORK these extensions include:

- A mandatory QAA Level – The quality of authentication required for the subject. Please see deliverable D2.3 for the authentication quality scheme.
- An optional eIDSectorShare, by which the SP can indicate whether an eID can be shared within the Service Provider's sector or not.
- An optional eIDCrossSectorShare, by which the SP can indicate whether an eID can be shared outside of the Service Provider's sector or not.
- An optional eIDCrossBorderShare, by which the SP can indicate whether an eID can be shared outside of the Service Provider's member state or not.
- An optional <RequestedAttributes> element to allow additional STORK attributes to be requested

- Additional attributes necessary for processing the authentication. Any additional attributes which are not specified in the following paragraphs are ignored.

5.1.4.1 <stork:QualityAuthenticationAssuranceLevel>

Required: Mandatory

The minimum STORK QAA level that the subject's authentication method must meet.

5.1.4.2 <storkp:eIDSectorShare>

Required: Optional

Base Type: Boolean

Default: false

This element permits a SP to indicate whether an eID can be shared within the Service Provider's sector. This element is mandatory for requests sent to some countries which derive the eIdentifier, depending on the context in which it is to be used, e.g. BE.

5.1.4.3 <storkp:eIDCrossSectorShare>

Required: Optional

Base Type: Boolean

Default: false

This element permits a SP to indicate whether an eID can be shared outside of the Service Provider's sector. This element is mandatory for requests sent to some countries which derive the eIdentifier, depending on the context in which it is to be used, e.g. BE.

5.1.4.4 <storkp:eIDCrossBorderShare>

Required: Optional

Base Type: Boolean

Default: false

This element permits a SP to indicate whether an eID can be shared outside of the Service Provider's Member State. This element is mandatory for requests sent to some countries which derive the eIdentifier, depending on the context in which it is to be used, e.g. BE.

5.1.4.5 <stork:spSector>

Required: Optional

Base Type: UTF-8, maximum 20 characters.

This element defines a SP Sector. This element is mandatory for requests sent to some countries which derive the eIdentifier, depending on the context in which it is to be used, e.g. BE.

5.1.4.6 <stork:spApplication>

Required: Optional

Base Type: UTF-8,maximum 100 characters.

This element defines a SP Application. This element is mandatory for requests sent to some countries which derive the eIdentifier, depending on the context in which it is to be used, e.g. BE.

5.1.4.7 <stork:spCountry>

Required: Optional

Base Type: Alpha string ISO3166-1 or “EU”

This element defines a SP Country. This element is mandatory for requests sent to some countries which derive the eIdentifier, depending on the context in which it is to be used, e.g. BE.

5.1.4.8 <storkp:RequestedAttributes>

Required: Optional

This element contains zero or more <stork:RequestedAttribute>. This allows additional STORK attributes to be requested to be added to the Authentication Response.

5.1.4.8.1 < stork:RequestedAttribute>

```
<complexType name="RequestedAttributeType">
<sequence>
<element ref="stork:AttributeValue" type="anyType" minOccurs="0"
maxOccurs="unbounded"/>
</sequence>
<attribute name="Name" type="string" use="required"/>
<attribute name="NameFormat" type="anyURI" use="required"/>
<attribute name="FriendlyName" type="string" use="optional"/>
<anyAttribute namespace="##other" processContents="lax"/>
<attribute name="isRequired" type="boolean" use="optional"/>
</complexType>
```

Required: Optional

An <stork:RequestedAttribute> element is necessary for each STORK Attribute required with an authentication. The list of available STORK Attributes including their Names and Formats is in chapter 6.3 below. If any unknown attribute is requested, it will be ignored. Nevertheless, if such unknown attribute is marked as mandatory the request is rejected

Attribute	Required	Allowable Values	Notes
Name	Mandatory		Agreed Name of Attribute Required
NameFormat	Mandatory		Agreed Format of the Attribute Name
FriendlyName	Optional		A friendly name for the attribute that can be displayed to a user e.g. when requesting confirmation to send to SP. As all initial attribute will be STORK Attributes a UI can display the local name for that attribute. If used then this should either be in the SP language or English.
isRequired	Optional	“true” or “false”	A Boolean to specify if the SP requires this attribute or if it is optional

Table 6: Attributes of Person’s Attributes in an Authentication Request

5.1.4.8.1.1 <stork:AttributeValue>

Required: Optional

The <stork:AttributeValue> element allows the requestor to specify that the attribute requested must have one of the specified values i.e. only return this attribute if the value of this attribute value (for the subject) is one of the requested values.

This is used to request derived attributes e.g. IsAgeOver. For example if the requestor wants to know if the subject is over 16 then they can request the IsAgeOver attribute with the requested <stork:AttributeValue> of 16. If the subject was 18 the attribute would be returned in the SAML response assertion with a value of 16. If the subject was 15 then the resultant attribute in the assertion would have no value.

Note this does not cater for complex conditions such as Is Subject aged between 15 and 18. Nevertheless, if the attribute isAgeOver is requested twice, e.g. one isAgeOver(15) and the other one isAgeOver(18), the reply gives information on whether or not the subject is in this range of ages.

5.1.4.9 <storkp:AuthenticationAttributes>

Required: Optional

This element covers all necessary attributes required for completing the authentication process

5.1.4.9.1 <storkp:VIDPAuthenticationAttributes>

Required: Optional, Mandatory for VIdP implementations

Requests to VIdP‘s must include this element and its child elements since the contents of these elements are required for finishing the authentication process.

5.1.4.9.1.1 <storkp:CitizenCountryCode>

Required: Mandatory for citizens of V-IDP countries, absent for others

Base Type: Alpha string ISO3166-1

This element specifies the citizen country code and is used for delegating the authentication request to the appropriate SPWare.

5.1.4.9.1.2 <storkp:SPInformation>

Required: Mandatory

This element encapsulates additional service provider information required for VIdP -based authentications.

5.1.4.9.1.2.1 <storkp:SPID>

Required: Mandatory

Base Type: UTF-8, maximum 40 characters

The value of this element uniquely identifies a service provider. It need not have human readable information; it associates the service provider with data registered about him at the V-IDP

5.1.4.9.1.2.2 <storkp:SPCertSig>

Required: Optional

This element contains the digital certificate the service provider has signed the authentication request with. The certificate is placed in a <ds:KeyInfo> element. For details see the XML Signature Syntax [7].

5.1.4.9.1.2.3 <storkp:SPCertEnc>

Required: Optional

This element contains the encryption certificate of the service provider. A SAML issuing entity can use this certificate to encrypt an eID and/or personal attributes. Due to this fact, end to end encryption can be achieved. The certificate is placed in a <ds:KeyInfo> element. For details see the XML Signature Syntax [7].

5.1.4.9.1.2.4 <storkp:SPAuthRequest>

Required: Optional

Base Type: any

This element contains the original authentication request of a service provider if the S-PEPS acts as intermediary between SP-PEPS and VIdP.

5.1.5 <saml:Subject>

Required: Not Used

This element specifies the subject whom the assertion is requested for. This element is optional in the SAML specification and will not be used in STORK.

5.1.6 <saml:NameIdPolicy>

```
<element name="NameIDPolicy" type="samlp:NameIDPolicyType"/>
  <complexType name="NameIDPolicyType">
      <attribute name="Format" type="anyURI" use="optional"/>
      <attribute name="SPNameQualifier" type="string" use="optional"/>
      <attribute name="AllowCreate" type="boolean" use="optional"/>
  </complexType>
```

Required: Optional

Requests specific formats and qualifiers for the Identifier that represents the Subject – Note: the <NameIdPolicy> in the Response may not have the requested specific formats and qualifiers.

Attribute	Required	Allowable Values	Notes
Format	Not Used		A URI defining the requested format of the NameId in the Response. STORK will not use NameId to contain the citizen's eId, it will be a separate attribute.
SPNameQualifier	Not Used		Requests that the assertion's subject identifier be returned in the namespace other than the requestor's.
AllowCreate	Optional Mandatory for all Formats except transient when it is "Not Used"	True	Allows the SAML responder to create a new identifier for the subject. Recommended by OASIS to be set to true for maximum interoperability; default is "false".

Table 7: NameId Attributes of an Authentication Request

5.1.7 <saml:Conditions>

Required: Not Used

This defines SAML conditions the requestor expects to limit the validity and/or use of the resultant assertions in the Authentication Response.

5.1.8 <samlp:RequestedAuthnContext>

Required: Not Used

The <RequestedAuthnContext> element is used to restrict the authentication methods that the IdP will use to authenticate the user. The only restriction in the method of authentication should be the STORK QAA level. As this is not an authentication method it has been placed in the extension.

5.1.9 <samlp:Scoping>

Required: Not Used

Scoping defines the Identity Providers acceptable to the requestor and can place limits and context on proxying.

Attribute	Required	Allowable Values	Notes
Count	Not Used	0	Limits the number of proxy indirections between the receiver and the ultimate identity provider – must be unspecified or non-zero for STORK. Default is no restrictions.

Table 8: SAML: Attributes of Scoping element

5.1.9.1 <samlp:IDPList>

Required: Not Used

Request to limit the choice of Identity Providers [e.g. C-PEPSs or C-VIdPs] that the receiver [S-PEPS or S-VIdP] may use.

5.1.9.2 <samlp:RequesterID>

Required: Not Used

Zero or more showing the Ids of the requestors this requestors is acting on behalf of.

5.1.10 Sample Authentication Request

```
<saml2p:AuthnRequest
  xmlns:saml2p="urn:oasis:names:tc:SAML:2.0:protocol"
  xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
  xmlns:saml2="urn:oasis:names:tc:SAML:2.0:assertion"
  xmlns:stork="urn:eu:stork:names:tc:STORK:1.0:assertion"
  xmlns:storkp="urn:eu:stork:names:tc:STORK:1.0:protocol"
  AssertionConsumerServiceURL="http://S-
PEPS.gov.xx/PEPS/ColleagueResponse"
  Consent="urn:oasis:names:tc:SAML:2.0:consent:unspecified"
  Destination="http://C-PEPS.gov.xx/PEPS/ColleagueRequest"
  ForceAuthn="true"
```

```
  ID="390205d2-ea52-4aaa-966c-61f312131ddc"
  IsPassive="false"
  IssueInstant="2010-02-03T17:06:18.521Z"
  ProtocolBinding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"
  ProviderName="University Oxford"
  Version="2.0">
  <saml2:Issuer Format="urn:oasis:names:tc:SAML:2.0:nameid-
format:entity">http://S-PEPS.gov.xx</saml2:Issuer>
   <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
      <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="#390205d2-ea52-4aaa-966c-61f312131ddc">
            <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#"><ec:InclusiveNamespaces
xmlns:ec="http://www.w3.org/2001/10/xml-exc-c14n#" PrefixList="ds saml2
saml2p stork storkp xs"/>
              </ds:Transform>
           </ds:Transforms>
          <ds:DigestMethod
Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
          <ds:DigestValue>tm8x2Yk/YEJMiiHjC2f3poypLXg=</ds:DigestValue>
       </ds:Reference>
    </ds:SignedInfo>
<ds:SignatureValue>JXo0KtPOFmA3aixdlzqUH1gXY/NmBqSC7V+lQrxefwxglBfI4Z1xb
TqL32uZXJ+ZEsXhw7vFxuw9WRjRTtwFkYkk+ob61fyB2YeFTVbli0ZHFVzXxccPGgogvDmZr
uXbunaRlG3s/6ZumJAPAvkAU6fBxYVC9AqXdJVrFWezHCI=</ds:SignatureValue>
<ds:KeyInfo><ds:X509Data><ds:X509Certificate>MIIBkTCB+wIESpT8bTANBgkqhki
G9w0BAQUFADAQMQ4wDAYDVQQDEwVzdG9yazAeFw0wOTA4MjYwOTEyMTNaFw0wOTExMjQwOTE
yMTNaMBAxDjAMBgNVBAMTBXN0b3JrMIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQDK/QL
8NuMd41I1lObObeRA6DaM8bjeYqIz5mg5WnnZv4jlcK7Gq89Lk6htXRFllAXpDYhI3zolMIM
HEMZ3zQQPc7lgTV6Bbz9uD2YTJ9Kx55e8Y6Y49DO+TiiejGJxTzTFUcuBJHaKipuvLVd1a8N
3RAnaGSUOozhrTqxba82mEwIDAQABMA0GCSqGSIb3DQEBBQUAA4GBAFJgeS18nhUlr7WnvSn
9FlkI94U//Hk3iZLc3/cScTu7D7Y/J0eUq4TF8PsSzWX5khGuqrTkswNgfEtl2IpACQ2wkB8
+RxeRNdddQlGHlO4ZqnpvxXBwSouiy2yUeAo0y++vMFm6DO4sxfk8eTtimPDo5SzBtjtGtbq
S3cyl/wz8</ds:X509Certificate></ds:X509Data></ds:KeyInfo>
   </ds:Signature>
   <saml2p:Extensions
xmlns:saml2p="urn:oasis:names:tc:SAML:2.0:protocol">

<stork:QualityAuthenticationAssuranceLevel>3</stork:QualityAuthenticatio
nAssuranceLevel>
      <storkp:eIDSectorShare>false</stork:eIDSectorShare>
      <storkp:eIDCrossSectorShare>false</stork:eIDCrossSectorShare>
      <storkp:eIDCrossBorderShare>false</stork:eIDCrossBorderShare>
      <storkp:RequestedAttributes>
         <stork:RequestedAttribute
Name="http://www.stork.gov.eu/1.0/isAgeOver"
```

```
NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri"
isRequired="true">
            <stork:AttributeValue
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="xs:anyType">16</saml2:AttributeValue>
            <stork:AttributeValue
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="xs:anyType">18</saml2:AttributeValue>
         </stork:RequestedAttribute>
         <stork:RequestedAttribute
Name="http://www.stork.gov.eu/1.0/dateOfBirth"
NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri"
isRequired="false"/>
         <stork:RequestedAttribute
Name="http://www.stork.gov.eu/1.0/eIdentifier"
NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri"
isRequired="true"/>
      </storkp:RequestedAttributes>
   </stork:Extensions>
</saml2p:AuthnRequest>
```

5.2 Authentication Response

An authentication response may not be larger than 128k bytes

5.2.1 <samlp:AuthnResponse>

```
      <sequence>
            <element ref="saml:Issuer" minOccurs="0"/>
<element ref="ds:Signature" minOccurs="0"/>
<element ref="samlp:Extensions" minOccurs="0"/>
      <element ref="samlp:Status"/>
<choice minOccurs="0" maxOccurs="unbounded">
<element ref="saml:Assertion"/>
<element ref="saml:EncryptedAssertion"/>
</choice>
      </sequence>
      <attribute name="ID" type="ID" use="required"/>
      <attribute name="InResponseTo" type="NCName" use="optional"/>
      <attribute name="Version" type="string" use="required"/>
      <attribute name="IssueInstant" type="dateTime" use="required"/>
      <attribute name="Destination" type="anyURI" use="optional"/>
      <attribute name="Consent" type="anyURI" use="optional"/>
```

Required: Mandatory

Attribute	Required	Allowable Values	Notes
ID	Mandatory		Random value between 128-160 bits long
InResponseTo	Mandatory		The identifier (ID) of the request this response refers to.
Version	Mandatory	2.0	
IssueInstant	Mandatory		UTC Date & time response was issued.
Destination	Mandatory		URI reference of the S-PEPS, VIdP or SP-MW SAML Response processor this response is being sent to. (Should be the same as AssertionConsumerServiceURL in the associated Authentication Request.)
Consent	Optional	urn:oasis:names:tc:SAML:2.0:consent:obtained urn:oasis:names:tc:SAML:2.0:consent:prior urn:oasis:names:tc:SAML:2.0:consent:curent-implicit urn:oasis:names:tc:SAML:2.0:consent:curent-explicit urn:oasis:names:tc:SAML:2.0:consent:unspecified	Defines the type of user consent obtained from the user for this authentication and data transfer.

Table 9: SAML Attributes of an Authentication Response

5.2.2 <saml:Issuer>

```
<element name="Issuer" type="saml:NameIDType"/>
<complexType name="NameIDType">
    <simpleContent>
        <extension base="string">
            <attributeGroup ref="saml:IDNameQualifiers"/>
            <attribute name="Format" type="anyURI" use="optional"/>
            <attribute name="SPProvidedID" type="string" use="optional"/
>
        </extension>
    </simpleContent>
</complexType>
```

Required: Mandatory

The <Issuer> element must contain a URI that identifies the issuing PEPS and must be mutually agreed with the receiving PEPS organisation.

Attribute	Required	Allowable Values	Notes
NameQualifier	Not used		The security domain that qualifies that name.
SPNameQualifier	Not used		Qualifying the name with a name of a service provider.
Format	Optional	urn:oasis:names:tc:SAML:2.0:nameid-format:entity	URI representing the classification of the identifier. Default is urn:oasis:names:tc:SAML:2.0:nameid-format:entity.
SPProvidedID	Not used		Name identifier if different from the name in the contents of the element.

Table 10: SAML:Issuer Attributes

5.2.3 <ds:Signature>

Required: Mandatory

If the HTTP POST Binding is used, the SAML response must be signed. In case of the SOAP binding similar techniques like SSL/TL can be used instead. The SOAP binding is only supported for the communication between SP-MW and V-IDP.

An XML Signature authenticates the sender (C-PEPS or C-VIdP) and ensures message integrity (signature over complete assertion). The signature must be an enveloped signature and applied to the <samlp:Response> element and all its children. The signature must contain a single <ds:Reference> containing the ID attribute value of the <samlp:Response> element.

<ds:Signature> is defined in http://www.w3.org/TR/xmldsig-core/.

5.2.4 <samlp:Extensions>

Required: Not Used

This element provides the possibility to optionally extend the SAML response message.

5.2.5 <samlp:Status>

```
<element name="Status" type="samlp:StatusType"/>
```

```
    <complexType name="StatusType">
        <sequence>
            <element ref="samlp:StatusCode"/>
            <element ref="samlp:StatusMessage" minOccurs="0"/>
            <element ref="samlp:StatusDetail" minOccurs="0"/>
        </sequence>
    </complexType>
    <element name="StatusCode" type="samlp:StatusCodeType"/>
    <complexType name="StatusCodeType">
        <sequence>
            <element ref="samlp:StatusCode" minOccurs="0"/>
        </sequence>
        <attribute name="Value" type="anyURI" use="required"/>
    </complexType>
    <element name="StatusMessage" type="string"/>
    <element name="StatusDetail" type="samlp:StatusDetailType"/>
    <complexType name="StatusDetailType">
        <sequence>
            <any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
        </sequence>
    </complexType>
```

Required: Mandatory

<samlp:StatusCode>

Required: Mandatory

Attribute	Required	Allowable Values	Notes
Value	Mandatory	Values of section 3.2.2.2 in [3]	A URI reference representing the status code value.

Table 11: Attributes of element StatusCode

Specifies an optional set of nested status codes and a value attribute representing the status of the Authentication Request.

A list of status codes are defined by OASIS in the SAML 2.0 specification, and these will be adopted wherever relevant. Additional STORK specific status codes are used for STORK specific situations.

The status code will consist of two elements:

- Top-level status code, indicating the status of the authentication operation. This status is included as an URI in the "Value" attribute of the <samlp:StatusCode> element:
- A subordinate status code that provides more specific information on an error condition. This status is included as a nested element in the <samlp:StatusCode> child element.

The Top-level status is MANDATORY. The subordinate status code is also MANDATORY, if the error produced during the STORK operation is covered by one of the subordinate status codes defined below. Otherwise, it is OPTIONAL.

Values for both levels of status codes are listed below. For further information, please refer to OASIS SAML 2.0 specification [2].

Top-level status codes for STORK

- urn:oasis:names:tc:SAML:2.0:status:Success

  The request succeeded. No additional information needs to be returned in the <StatusMessage> nor <StatusDetail> elements.

- urn:oasis:names:tc:SAML:2.0:status:Requester
- The request could not be performed due to an error on the part of the requester.
- urn:oasis:names:tc:SAML:2.0:status:Responder

The request could not be performed due to an error on the part of the SAML responder or SAML authority.

Subordinate status codes for STORK

- urn:oasis:names:tc:SAML:2.0:status:AuthnFailed

The responding provider was unable to successfully authenticate the principal.

- urn:oasis:names:tc:SAML:2.0:status:InvalidAttrNameOrValue

Unexpected or invalid content was encountered within a <saml:Attribute> or <saml:AttributeValue> element.

- urn:oasis:names:tc:SAML:2.0:status:InvalidNameIDPolicy

The responding provider cannot or will not support the requested name identifier policy.

- urn:oasis:names:tc:SAML:2.0:status:RequestDenied

The SAML responder or SAML authority is able to process the request but has chosen not to respond.

This status code MAY be used when there is concern about the security context of the request message or the sequence of request messages received from a particular requester.

The SAML 2.0 specificataion allows additional status codes (agreed by both the requestor and responder) to be used. The current list of STORK specific status code are:

Subordinate STORK specific status codes:

- http://www.stork.gov.eu/saml20/statusCodes/QAANotSupported

The requested QAA is not supported by the Citizen country. There is no national QAA that can be mapped to the requested STORK QAA or a higher one.

<samlp:Status-Message>

Required: Optional

It is a string explaining the status value in human readable terms. The table below defines the status messages in English. If a country chooses to provide these messages in their local language then they must be semantically equivalent to the English message given below.

If the subordinate status code is included in the response, then the status message must be the one corresponding to the subordinate status code, not the top-level status code.

Status Code	Status Message
urn:oasis:names:tc:SAML:2.0:status:Success	-
urn:oasis:names:tc:SAML:2.0:status:Requester	The request could not be performed due to an error on the SAML requester side (S-PEPS, S-VIdP or SP-MW) identified by its URI.
urn:oasis:names:tc:SAML:2.0:status:Responder	The request could not be performed due to an error on the SAML responder side (C-PEPS, V-IDP) identified by its URI.
urn:oasis:names:tc:SAML:2.0:status:AuthnFailed	It was unable to successfully authenticate the citizen
urn:oasis:names:tc:SAML:2.0:status:InvalidAttrNameOrValue	Unexpected or invalid content was encountered within a <saml:Attribute> or <saml:AttributeValue> element
urn:oasis:names:tc:SAML:2.0:status:InvalidNameIDPolicy	The requested name identifier policy is not supported.
urn:oasis:names:tc:SAML:2.0:status:RequestDenied	The request has not been processed.
http://www.stork.gov.eu/saml20/statusCodes/QAANotSupported	The requested QAA cannot be reached by the citizen's country.

Table 12: Status messages for each status code

<samlp:Status-Detail>

Required: Not used

This element can be used to specify additional information concerning the status represented of zero ore more elements from any namespace.

5.2.6 <saml:Assertion>

Required: Mandatory if authentication was successful and <samlp:StatusCode> is "urn:oasis:names:tc:SAML:2.0:status:Success". If the <samlp:Response> contains an error message and another <samlp:StatusCode>, a SAML assertion must not be present.

The Authentication Response must contain a SAML <saml:Assertion> element if the authentication was successful. The <saml:Assertion> will contain a single <saml:Subject> indicating the user which the <saml:Assertion> relates to. It will also contain a single <saml:AuthnStatement> containing the results of the user authentication and a single <saml:AttributeStatement> containing zero or more <saml:Attribute>s asserted about the user. A detailed description of the <saml:Assertion> is given in section 5.3.

5.2.7 <saml:EncryptedAssertion>

Required: Not Used

It has been agreed that encrypted assertions will not be used within STORK.

5.2.8 Sample Authentication Response

```
<saml2p:Response
  xmlns:saml2p="urn:oasis:names:tc:SAML:2.0:protocol"
  xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
  xmlns:saml2="urn:oasis:names:tc:SAML:2.0:assertion"
  xmlns:stork="urn:eu:stork:names:tc:STORK:1.0:assertion"
  xmlns:storkp="urn:eu:stork:names:tc:STORK:1.0:protocol"
  Consent="urn:oasis:names:tc:SAML:2.0:consent:obtained"
  Destination="http://S-PEPS.gov.xx/PEPS/ColleagueRequest"
  ID="698491a7-94c0-41e3-ab2a-3ada2f3e231c"
  InResponseTo="390205d2-ea52-4aaa-966c-61f312131ddc"
  IssueInstant="2010-02-03T17:06:17.864Z"
  Version="2.0">
   <saml2:Issuer Format="urn:oasis:names:tc:SAML:2.0:nameid-
format:entity">http://C-PEPS.gov.xx</saml2:Issuer>
   <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
     <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="#698491a7-94c0-41e3-ab2a-3ada2f3e231c">
           <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#"><ec:InclusiveNamespaces
xmlns:ec="http://www.w3.org/2001/10/xml-exc-c14n#" PrefixList="ds saml2
saml2p stork storkp xs"/></ds:Transform>
           </ds:Transforms>
           <ds:DigestMethod
Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
           <ds:DigestValue>8IO1IBY0VqHQKlkcIy6qn9Hb++I=</ds:DigestValue>
        </ds:Reference>
     </ds:SignedInfo>

<ds:SignatureValue>rxvIFFUJ9jFVOR7DcwFJmv6NjQomVEBAQ5V26vap+LqzkhooBWlRL
CTvPYurNxgmMJp2Rkq9iV0rmITjbCWV2MrNal7aCFi0u2VHWvy4Oc64LBKPSEyJWdnt1VT2w
0xWyj0hYw4WsFVnqV3CKWDjdWtdZPa+GUnyl0mXePCFmB4=</ds:SignatureValue>
<ds:KeyInfo><ds:X509Data><ds:X509Certificate>MIIBkTCB+wIESpT8bTANBgkqhki
G9w0BAQUFADAQMQ4wDAYDVQQDEwVzdG9yazAeFw0wOTA4MjYwOTEyMTNaFw0wOTExMjQwOTE
yMTNaMBAxDjAMBgNVBAMTBXN0b3JrMIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQDK/QL
8NuMd41I1lObObeRA6DaM8bjeYqIz5mg5WnnZv4jlcK7Gq89Lk6htXRFllAXpDYhI3zolMIM
HEMZ3zQQPc7lgTV6Bbz9uD2YTJ9Kx55e8Y6Y49DO+TiiejGJxTzTFUcuBJHaKipuvLVd1a8N
3RAnaGSUOozhrTqxba82mEwIDAQABMA0GCSqGSIb3DQEBBQUAA4GBAFJgeS18nhUlr7WnvSn
9FlkI94U//Hk3iZLc3/cScTu7D7Y/J0eUq4TF8PsSzWX5khGuqrTkswNgfEtl2IpACQ2wkB8
```

```
+RxeRNdddQlGHlO4ZqnpvxXBwSouiy2yUeAo0y++vMFm6DO4sxfk8eTtimPDo5SzBtjtGtbq
S3cyl/wz8</ds:X509Certificate></ds:X509Data></ds:KeyInfo>
   </ds:Signature>
   <saml2p:Status>
      <saml2p:StatusCode
Value="urn:oasis:names:tc:SAML:2.0:status:Success"/>

<saml2p:StatusMessage>urn:oasis:names:tc:SAML:2.0:status:Success</saml2p
:StatusMessage>
   </saml2p:Status>
   <saml2:Assertion ID="848aa4c8-0745-4576-b8b6-8bd5a748cbc8"
   …
   </saml2:Assertion>
</saml2p:Response>
```

5.3 SAML Assertion

A SAML assertion is a packet of security information. It specifies that this assertion was issued by an entity at a certain time about a subject and attests the entity's identity provided that specified conditions are met.

5.3.1 <saml:Assertion>

```
<element name="Assertion" type="saml:AssertionType"/>
<complexType name="AssertionType">
    <sequence>
        <element ref="saml:Issuer"/>
        <element ref="ds:Signature" minOccurs="0"/>
        <element ref="saml:Subject" minOccurs="0"/>
        <element ref="saml:Conditions" minOccurs="0"/>
        <element ref="saml:Advice" minOccurs="0"/>
        <choice minOccurs="0" maxOccurs="unbounded">
            <element ref="saml:Statement"/>
            <element ref="saml:AuthnStatement"/>
            <element ref="saml:AuthzDecisionStatement"/>
            <element ref="saml:AttributeStatement"/>
        </choice>
    </sequence>
    <attribute name="Version" type="string" use="required"/>
    <attribute name="ID" type="ID" use="required"/>
    <attribute name="IssueInstant" type="dateTime" use="required"/>
</complexType>
```

Required: Mandatory if authentication was successful and <samlp:StatusCode> is "urn:oasis:names:tc:SAML:2.0:status:Success".

Attribute	Required	Allowable Values	Notes
ID	Mandatory		Random value between 128-160 bits long
Version	Mandatory	2.0	
IssueInstant	Mandatory		UTC Date & time Assertion was issued.

Table 13: SAML Attributes of an Authentication Responset

Conforming to the Authentication Request Protocol the Assertion contains one AuthnStatement and zero or one AttributeStatements.

5.3.2 <saml:Issuer>

```
<element name="Issuer" type="saml:NameIDType"/>
<complexType name="NameIDType">
    <simpleContent>
        <extension base="string">
            <attributeGroup ref="saml:IDNameQualifiers"/>
            <attribute name="Format" type="anyURI" use="optional"/>
            <attribute name="SPProvidedID" type="string" use="optional"/
>
        </extension>
    </simpleContent>
</complexType>
```

Required: Mandatory

This element identifies the entity that generated the <saml:Assertion>. The <saml:Issuer> element is mandatory within an <saml:Assertion> and contains a string value (URI) referring to the issuing entity.

The <saml:Issuer> element must contain a URI that identifies the issuing PEPS or VIdP. This URI must be mutually agreed with the receiving PEPS or receiving SP-MW.

Attribute	Required	Allowable Values	Notes
NameQualifier	Not used		The security domain that qualifies that name.
SPNameQualifier	Not used		Qualifying the name with a name of a service provider.
Format	Optional	urn:oasis:names:tc:SAML:2.0:nameid-format:entity	URI representing the classification of the identifier. Default is urn:oasis:names:tc:SAML:2.0:nameid-format:entity.
SPProvidedID	Not used		Name identifier if different from the name in the contents of the element.

Table 14: SAML:Issuer Attributes

5.3.3 <ds:Signature>

Required: Optional

Since the complete <samlp:Response> must be signed if the HTTP-Post Binding is used the signing of the SAML assertion is optionally.

5.3.4 <saml:Subject>

```
<complexType name="SubjectType">
    <choice>
        <sequence>
            <choice>
                <element ref="saml:BaseID"/>
                <element ref="saml:NameID"/>
                <element ref="saml:EncryptedID"/>
            </choice>
            <element ref="saml:SubjectConfirmation" minOccurs="0" maxOcc
urs="unbounded"/>
```

```
        </sequence>
        <element ref="saml:SubjectConfirmation" maxOccurs="unbounded"/>
    </choice>
</complexType>
```

Required: Mandatory

This element specifies the principal to whom the SAML statements are asserted to. The subject can be identified by three different elements:

- <saml:BaseID>
- <saml:NameID>
- <saml:EncryptedID>

Within the STORK context, only the element <saml:NameID> or <saml:EncryptedID> are used. Inter PEPS or PEPS-VIdP assertions will only support NameId. Encrypted IDs will only be supported by issuing VIdPs whose SPWares support encryption as well.

5.3.4.1 <saml:NameId>

Required: Mandatory

Identifier that represents the Subject – Note: the <NameIdPolicy> in the Request may have requested specific formats and qualifiers.

Attribute	Required	Allowable Values	Notes
NameQualifier	Mandatory		Security or Admin Domain that qualifies the name. This should be the namespace of the C-PEPS/VIdP.
SPNameQualifier	Optional		Further qualifies the name with a [group of] Service Provider. This should be the namespace of the original Service Provider
Format	Mandatory	urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified	A URI defining the format of the NameId. The Citizen's eID is provided in a separate attribute. NameId should not be used to assert the subject's identity but may be used to assert return visits from a user using the same authentication.
SPProvidedID	Not Used		Name identifier if different from the name in the contents of the element.

Table 15: NameId Attributes of an Authentication Response

5.3.4.2 <saml:EncryptedID>

Required: Optional (Only supported by VIdPs if the SPWare also supports encryption)

This element can be optionally used in the <saml:Subject> context instead of the <saml:NameID> element and carries the eID in encrypted format.

5.3.4.2.1 <xenc:EncryptedData>

Required: Mandatory, if <saml:EncryptedID> is used

This element is mandatory and contains the encrypted contents. For details to this element we refer to XML Encryption Syntax and Processing [7].

5.3.4.2.2 <xenc:EncryptedKey>

Required: Optional

This element contains zero or more wrapped encryption keys as defined in [7].

5.3.4.3 <saml:SubjectConfirmation>

```
<complexType name="SubjectConfirmationType">
    <sequence>
        <choice minOccurs="0">
            <element ref="saml:BaseID"/>
            <element ref="saml:NameID"/>
            <element ref="saml:EncryptedID"/>
        </choice>
        <element ref="saml:SubjectConfirmationData" minOccurs="0"/>
    </sequence>
    <attribute name="Method" type="anyURI" use="required"/>
</complexType>
```

Required: Mandatory

The <saml:SubjectConfirmation> element provides means for verification of the correspondence between the SAML subject with the party whom the relying party is communicating with.

Attribute	Required	Allowable Values	Notes
Method	Mandatory	urn:oasis:names:tc:SAML:2.0:cm:bearer urn:oasis:names:tc:SAML:2.0:cm:holder-of-key	Bearer is mandatory to allow in one SubjejctConfirmation to support the Browser SSO profile. Holder-of-key may be used in another SubjectConfirmation (where supported by the Issuer) to allow Holder-of-Key Web Browser Profile.

Table 16: SubjectConfirmation Attributes of an Authentication Response

The SAML 2.0 specification allows multiple SubjectConfirmations and permits the recipient of a SAML assertion to use any of the methods within those SubjectConfirmation to confirm the attesting entity. If the Holder-of-Key Web Browser Profile is supported by the recipient and the holder-of-key method is present in a SubjectConfirmation then it should be used in preference to the SubjectConfirmation containing the bearer method to eliminate the possibility of man-in-the middle attacks.

5.3.4.4 <saml:BaseId>, <saml:NameId>, <saml:EncryptedID>

Within STORK the subject entity is deemed to be the same as the attesting entity and therefore <saml:BaseID>, <saml:NameID> and <saml:EncryptedID> elements must not be used.

5.3.4.5 <saml:SubjectConfirmationData>

```
<complexType name="SubjectConfirmationDataType" mixed="true">
    <complexContent>
        <restriction base="anyType">
            <sequence>
                <any namespace="##any" processContents="lax" minOccurs="
0" maxOccurs="unbounded"/>
<element ref="ds:KeyInfo" minOccurs="0">
            </sequence>
            <attribute name="NotBefore" type="dateTime" use="optional"/>
            <attribute name="NotOnOrAfter" type="dateTime" use="optional
"/>
            <attribute name="Recipient" type="anyURI" use="optional"/>
            <attribute name="InResponseTo" type="NCName" use="optional"/
>
            <attribute name="Address" type="string" use="optional"/>
            <anyAttribute namespace="##other" processContents="lax"/>
        </restriction>
    </complexContent>
</complexType>
```

Required: Mandatory

This element specifies additional data allowing the SAML subject to be confirmed.

Attribute	Required	Allowable Values	Notes
NotBefore	Optional		Not allowed under Browser SSO Profile i.e. where SubjectConfirmation method is bearer. If SubjectConfirmation method is holder-of-key then this value, if present, must be greater than or equal to the NotBefore attribute in the X.509 certificate.
NotOnOrAfter	Mandatory		Subject cannot be confirmed on or after this time. If SubjectConfirmation method is holder-of-key then this value must be less than or equal to the NotBefore attribute in the X.509 certificate.
Recipient	Mandatory		URI reference of the S-PEPS this assertion is being sent to. This should be the same value as the AssertionConsumerServiceURL attribute in the Authentication Request
InResponseTo	Mandatory		Id of the Request that requested this assertion
Address	Optional		IP address of user that this assertion was issued to. Mandatory for bearer SubjectConfirmation method as it allows Relying Parties to mitigate against a Man-In-The-Middle.

Table 17: SubjectConfirmationData Attributes of an Authentication Response

5.3.4.5.1.1 <ds:KeyInfo>

```
<element name="KeyInfo" type="ds:KeyInfoType"/>
 <complexType name="KeyInfoType" mixed="true">
<choice maxOccurs="unbounded">
<element ref="ds:KeyName"/>
```

```
<element ref="ds:KeyValue"/>
<element ref="ds:RetrievalMethod"/>
<element ref="ds:X509Data"/>
<element ref="ds:PGPData"/>
<element ref="ds:SPKIData"/>
<element ref="ds:MgmtData"/>
            <any processContents="lax" namespace="##other"/>
      </choice>
<attribute name="Id" type="ID" use="optional"/>
</complexType>
```

<ds:KeyInfo> is fully defined in second edition of the XML Signature specification (http://www.w3.org/TR/xmldsig-core/).

Required: Optional

At least one <ds:KeyInfo> element is required if the SubjectConfirmation method is holder-of-key. Where more than one is present then the presenter may use any one to confirm that they are the attesting entity.

It usage here is further constrained to contain exactly one <ds:X509Data>.

Attribute	Required	Allowable Values	Notes
Id	Not Used		

Table 18: KeyInfo Attributes of an SubjectConfirmation in Authentication Response

5.3.4.5.1.2 <ds:X509Data>

```
<element name="X509Data" type="ds:X509DataType"/>
 <complexType name="X509DataType" mixed="false">
<sequence maxOccurs="unbounded">
<choice>
<element ref="ds:X509IssuerSerial" type="ds: :X509IssuerSerialType"/>
<element ref="ds:X509SKI" type="base64Binary"/>
<element ref="ds:X509SubjectName" type="string"/>
<element ref="ds:X509Certificate" type="base64Binary"/>
<element ref="ds:X509CRL" type="base64Binary"/>
<element ref="ds:SPKIData"/>
<element ref="ds:MgmtData"/>
 <any processContents="lax" namespace="##other" minoccurs="1"
maxoccurs="1"/>
</choice>
</sequence>
</complexType>
```

<ds:X509Data> is fully defined in second edition of the XML Signature specification (http://www.w3.org/TR/xmldsig-core/).

Required: Mandatory

There must be exactly one <ds:X509Data> element.

The <ds:X509Data> element must contain a <ds:X509Certificate> element with the base-64 encoding of the X.509 certificate (i.e. a copy of the certificate being used in the current TLS session). If the <ds:X509Data> element contains other permitted child elements (more granular X.509 data) then the recipient must use the <ds:X509Certificate> element to confirm the attesting entity.

The <ds:X509CRL> element must not be used.

5.3.5 <saml:Conditions>

```
<complexType name="ConditionsType">
    <choice minOccurs="0" maxOccurs="unbounded">
        <element ref="saml:Condition"/>
        <element ref="saml:AudienceRestriction"/>
        <element ref="saml:OneTimeUse"/>
        <element ref="saml:ProxyRestriction"/>
    </choice>
    <attribute name="NotBefore" type="dateTime" use="optional"/>
    <attribute name="NotOnOrAfter" type="dateTime" use="optional"/>
</complexType>
```

Required: Mandatory

This element specifies conditions that must be evaluated when using the <Assertion>. These conditions must be the same as the requested conditions specified in the <AuthnRequest>

Attribute	Required	Allowable Values	Notes
NotBefore	Mandatory		Assertion not valid before this time
NotOnOrAfter	Mandatory		Assertion not valid on or after this time

Table 19: Conditions Attributes of an Authentication Response

5.3.5.1 <saml:Condition>

Required: Not Used

This element serves as an extension point for new conditions. New conditions must be derived from an abstract type.

5.3.5.2 <saml:AudienceRestriction>

```
<complexType name="AudienceRestrictionType">
    <complexContent>
        <extension base="saml:ConditionAbstractType">
            <sequence>
                <element ref="saml:Audience" maxOccurs="unbounded"/>
            </sequence>
        </extension>
    </complexContent>
</complexType>
```

Required: Mandatory

This should restrict the Audience of this Assertion to the S-PEPS or SP-MW and contains the URI reference of the S-PEPS or SP-MW this assertion is being sent to.

5.3.5.2.1 <saml:Audience>

```
<element name="Audience" type="anyURI"/>
```

Required: Mandatory

Specifies the audiences to whom new assertions are allowed to be issued (one or more elements).

5.3.5.3 <saml:OneTimeUse>

Required: Mandatory

Defines that this STORK Assertion has to be used immediately and cannot be retained for future use.

5.3.5.4 <saml:ProxyRestrictions>

Required: Not Used

This element specifies limitations for the case that a requesting party wants to issue itself assertions containing information extracted from the originally requested assertion.

5.3.6 <saml:Advice>

Required: Not Used

Can specify additional information for processing the assertion..

5.3.7 <saml:AuthnStatement>

```
<complexType name="AuthnStatementType">
    <complexContent>
        <extension base="saml:StatementAbstractType">
            <sequence>
                <element ref="saml:SubjectLocality" minOccurs="0"/>
                <element ref="saml:AuthnContext"/>
            </sequence>
            <attribute name="AuthnInstant" type="dateTime" use="required
"/>
            <attribute name="SessionIndex" type="string" use="optional"/
>
            <attribute name="SessionNotOnOrAfter" type="dateTime" use="o
ptional"/>
        </extension>
    </complexContent>
</complexType>
```

Required: Mandatory

Attribute	Required	Allowable Values	Notes
AuthnInstant	Mandatory		Date & Time User was actually authenticated
SessionIndex	Optional		Index of the User's IdP session. Allow for increased interoperability with other profiles.
SessionNotOnOrAfter	Not Used		When the User's IdP session is deemed to have expired.

Table 20: AuthnStatement Attributes of an Authentication Response

5.3.7.1 <saml:SubjectLocality>

```
<complexType name="SubjectLocalityType">
    <attribute name="Address" type="string" use="optional"/>
    <attribute name="DNSName" type="string" use="optional"/>
</complexType>
```

Required: Mandatory

This element should contain the DNS domain name and the IP address of the system from which the SAML subject was authenticated.

Attribute	Required	Allowable Values	Notes
Address	Mandatory		IP address of authenticating user's client system
DNSName	Optional		DNS Name of authenticating user's client system

Table 21: SubjectLocality Attributes of an Authentication Request

5.3.7.2 <saml:AuthnContext>

Required: Not Used

This element specifies the context of an authentication event. Instead STORK will rely on its own QAA levels.

5.3.8 <saml:AttributeStatement>

Required: Optional

This element contains several <Attribute> or <EncryptedAttribute> (Only supported by VIdPs and encryption supporting SPWares) elements containing attribute information associated with the SAML subject. For each requested STORK attribute in the <AuthnRequest> the <AttributeStatement> contains a single <Attribute> or <EncryptedAttribute> element if available.

5.3.8.1 <saml:Attribute>

```
<complexType name="AttributeType">
    <sequence>
        <element ref="saml:AttributeValue" minOccurs="0" maxOccurs="unbo
unded"/>
    </sequence>
    <attribute name="Name" type="string" use="required"/>
    <attribute name="NameFormat" type="anyURI" use="optional"/>
    <attribute name="FriendlyName" type="string" use="optional"/>
    <anyAttribute namespace="##other" processContents="lax"/>
</complexType>
```

Required: Optional

An <Attribute> element is required for each STORK Attribute requested in the original request. The list of available STORK Attributes including their Names and Formats is in chapter 6.3 below. If any unknown attribute was requested, it will be ignored. Nevertheless, if such unknown attribute was marked as mandatory the request is rejected.

Attribute	Required	Allowable Values	Notes
Name	Mandatory		Agreed Name of Attribute Required
NameFormat	Mandatory		Agreed Format of the Attribute Name
FriendlyName	Optional		A friendly name for the attribute that can be displayed to a user. The C-PEPS responsible for user consent so probably not required by S-PEPS or SP.
stork:AttributeStatus	Optional	"Available", "NotAvailable" or "Withheld"	Used to specify whether or not the <Attribute> requested was "Available", "NotAvailable" or "Withheld". The default value is "Available" i.e. attribute value has been returned.

Table 22: Attribute Attributes of an Authentication Request

5.3.8.1.1 <saml:AttributeValue>

Required: Optional

Value of the attribute, if available. Note: For derived attributes this should only have a value if the value related to the Subject is one of the requested values.

5.3.8.2 <saml:EncryptedAttribute>

Required: Optional

This element represents a <saml:Attribute> element in encrypted fashion. Attribute encryption is only supported by V-IdP implementations.

5.3.8.2.1 <xenc:EncryptedData>

Required: Mandatory, if <saml:EncryptedAttribute> is used.

When using encrypted attributes, this element is required and contains an encrypted <saml:Attribute>. See the SAML specification [3] and the XML Encryption Syntax and Processing specification [7] for details.

5.3.8.2.2 <xenc:EncryptedKey>

Required: Optional

This element can occur zero or more times and contains wrapped decryption keys as defined in [7].

5.3.9 Sample SAML Assertion

```
<saml2:Assertion
   xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
   xmlns:stork="urn:eu:stork:names:tc:PEPS:1.0:assertion"
   ID="848aa4c8-0745-4576-b8b6-8bd5a748cbc8"
   IssueInstant="2010-02-03T17:06:18.099Z" Version="2.0">
   <saml2:Issuer Format="urn:oasis:names:tc:SAML:2.0:nameid-
format:entity">http://C-PEPS.gov.xx</saml2:Issuer>
        <saml2:Subject>
          <saml2:NameID Format="urn:oasis:names:tc:SAML:1.1:nameid-
format:unspecified" NameQualifier="http://C-
PEPS.gov.xx">urn:oasis:names:tc:SAML:1.1:nameid-
format:unspecified</saml2:NameID>
          <saml2:SubjectConfirmation
Method="urn:oasis:names:tc:SAML:2.0:cm:bearer">
            <saml2:SubjectConfirmationData
              Address="111.222.333.444"
              InResponseTo="390205d2-ea52-4aaa-966c-61f312131ddc"
              NotOnOrAfter="2010-02-03T17:11:18.099Z"
              Recipient="http://S-PEPS.gov.xx/PEPS/ColleagueRequest"/>
          </saml2:SubjectConfirmation>
        </saml2:Subject>
        <saml2:Conditions
         NotBefore="2010-02-03T17:06:18.099Z"
```

```
          NotOnOrAfter="2010-02-03T17:06:18.099Z">
           <saml2:AudienceRestriction>
             <saml2:Audience>http://S-PEPS.gov.xx</saml2:Audience>
           </saml2:AudienceRestriction>
           <saml2:OneTimeUse/>
         </saml2:Conditions>
         <saml2:AuthnStatement AuthnInstant="2010-02-03T17:06:18.114Z">
           <saml2:SubjectLocality Address="111.222.333.444"/>
           <saml2:AuthnContext/>
         </saml2:AuthnStatement>
         <saml2:AttributeStatement>
           <saml2:Attribute Name="http://www.stork.gov.eu/1.0/isAgeOver"
NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri"
stork:AttributeStatus="Available">
             <saml2:AttributeValue
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="xs:string">16</saml2:AttributeValue>
         </saml2:Attribute>
         <saml2:Attribute Name="http://www.stork.gov.eu/1.0/dateOfBirth"
NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri"
stork:AttributeStatus="Available">
           <saml2:AttributeValue
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="xs:string">20081216</saml2:AttributeValue>
          </saml2:Attribute><saml2:Attribute
Name="http://www.stork.gov.eu/1.0/eIdentifier"
NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri"/
stork:AttributeStatus="NotAvailable">
      </saml2:AttributeStatement>
</saml2:Assertion>
```

5.4 Digitally Signing SAML.

```
<element name="Signature" type="ds:SignatureType"/>
   <complexType name="SignatureType">
     <sequence>
       <element ref="ds:SignedInfo"/>
       <element ref="ds:SignatureValue"/>
       <element ref="ds:KeyInfo" minOccurs="0"/>
       <element ref="ds:Object" minOccurs="0" maxOccurs="unbounded"/>
     </sequence>
     <attribute name="Id" type="ID" use="optional"/>
   </complexType>
```

<ds:Signature> is fully defined in http://www.w3.org/TR/xmldsig-core/.

SAML requests and responses must be signed using XML signature[7] if the HTTP-Post Binding is used. The signature must be based on xs:ID-typed attributes present in the root element,

specifically the ID attribute in <samlp:AuthnRequest> and <samlp:Response>. The signature must be an enveloped signature.

Signatures must be enveloped and should use RSA signing and verification. The current W3C XML Signature standard recommends http://www.w3.org/2009/xmldsig#rsa-sha1.

http://www.w3.org/2001/04/xmldsig-more#rsa-sha256 is under consideration by W3C as mandatory for a future version of XML Signature.

Exclusive Canonicalization [Excl-C14N] both with or without comments should be used in the <ds:CanonicalizationMethod> element in <ds:SignedInfo> and as a <ds:Transform> algorithm.

Signatures should not contain transforms other than the enveloped signature transform http://www.w3c.org/2000/09/xmldsig#enveloped-signature or the exclusive canonicalization transforms.

The use of <ds:KeyInfo> is optional in SAML. It is mandatory in STORK request, response and assertion signatures. It should contain the <ds:X509Data> element, with the key information specified in the <ds:X509Certificate> element (i.e. a copy of the signing certificate).

6 Stork Data Definitions

6.1 Data value Formats

This section defines some formats used in the data definitions described here after.

6.1.1 Dates

All dates are encoded in ISO 8601 format. It must be YYYYMMDD, YYYYMM or YYYY.

[YYYY] indicates a four-digit year, 0000 through 9999.

[MM] indicates a two-digit month of the year, 01 through 12.

[DD] indicates a two-digit day of that month, 01 through 31..

Warning, the day of month is optional. If the day of month is not provided, the month is optional also.

6.1.2 Country code

All country codes are encoded in alpha-2 string ISO3166-1 (uppercase), except for countryCodeOfBirth, which is defined on the ISO3166-3 (uppercase) domain. The reason for this difference is that someone may have been born in a country which doesn't exist anymore.

6.1.3 E-mail

All e-mail addresses are encoded following IETF RFC 822.

6.1.4 Strings

Strings can only contain UTF-8 readable characters. They cannot contain the <tab> character, for example.

Unless explicitly specified, strings cannot contain a <new line> character (<carriage return> or <line feed>).

In case multi-line is allowed for a string, <new lines> characters must be encoded with <line feed> only.

Multiple consecutive spaces are not allowed.

Definitions:

Alpha: only the [a-zA-Z] characters

Alphanumeric: only the [a-zA-Z0-9] characters

UTF-8: All readable UTF-8 characters

6.2 Authentication Request Data Definitions

This section defines the attributes additionally required for completing an authentication process. Below are Attributes that may be used in an authentiction request <extension> to

- define the required Qualified Authentication Assurance level of an authentication

Friendly Name	Element Name	Value Format	Description
QAA	stork:QualityAuthenticationAssuranceLevel	Numeric {1:4}	The minimum QAA level for this authentication

Table 23: Stork Authentication Request QAA Data definitions

- to supply parameters to allow the C-PEPS to issue a unique privacy-aware NationalId i.e. create a NationalId that is unique to the user but application specific.

Friendly Name	Element Name	Value Format	Description
Service Provider Sector	stork:spSector	Alphanumeric string	The Service Provider's sector – national agreement on standard values
Service Provider Application	stork:spApplication	UTF-8	A name for the Service Provider's application - SP specific
Service Provider Country	stork:spCountry	Alpha (Uppercase) ISO3166-1	The country of origin of the Service Provider (may be overridden by C-PEPS and substituted with the S-PEPS country

Table 24: Stork Authentication Request eID Data definitions

- Required by VIdP to complete its authentication process.

Element Name	Description
SPID	The unique identifier of the service provider requesting the authentication
CCC	Citizen country code of the citizen who wishes to authenticate
SPCertSig	Signing certificate of the service provider
SPCertEnc	Encryption certificate of the service provider
SPAuthRequest	The original authentication request of the service provider if the authentication request is transferred from a S-PEPS to a VIdP.

Table 25: Stork Authentication Request VIdP Data definitions

6.3 Subject Attribute Definitions

These are all attributes that can be queried about the subject of an authentication. They may be requested several times in one query, e.g. with isAgeOver this makes sense. Unknown attributes (attributes not listed in the following table) in a request are ignored.

Friendly Name	Name	Name Format	Description
eIdentifier	http://www.stork.gov.eu/1.0/eIdentifier	CC/CC/Base64	see [9]
Given Name	http://www.stork.gov.eu/1.0/givenName	UTF-8	see [9]

Surname	http://www.stork.gov.eu/1.0/surname	UTF-8	see [9]
Inherited Family Name	http://www.stork.gov.eu/1.0/inheritedFamilyName	UTF-8	see [9]
Adopted Family Name	http://www.stork.gov.eu/1.0/adoptedFamilyName	UTF-8	see [9]
Gender	http://www.stork.gov.eu/1.0/gender	"M" or "F"	see [9]
Date of Birth	http://www.stork.gov.eu/1.0/dateOfBirth	Date	see [9]
Country of Birth	http://www.stork.gov.eu/1.0/countryCodeOfBirth	ISO-3166-3	see [9]
Nationality	http://www.stork.gov.eu/1.0/nationalityCode	Country code	see [9]
Marital Status	http://www.stork.gov.eu/1.0/maritalStatus	S = Single M = Married P = Separated D = Divorced W = Widowed	see [9]
Text Residence Address	http://www.stork.gov.eu/1.0/textResidenceAddress	UTF-8 (with new lines)	see [9]
Canonical Residence Address	http://www.stork.gov.eu/1.0/canonicalResidenceAddress	XML	see 6.4.1
eMail Address	http://www.stork.gov.eu/1.0/eMail	e-mail	see [9]
Title	http://www.stork.gov.eu/1.0/title	UTF-8	see [9]
Residence Permit	http://www.stork.gov.eu/1.0/residencePermit	UTF-8	see [9]
Pseudonym	http://www.stork.gov.eu/1.0/pseudonym	UTF-8	see [9]
Age	http://www.stork.gov.eu/1.0/age	Numeric	see [9]
Is AgedOver	http://www.stork.gov.eu/1.0/isAgeOver	Requested age boundary if true, empty if false	see 5.1.4.8.1.1
Signed Document	http://www.stork.gov.eu/1.0/signedDoc	see below	see 6.5
Citizen QAA Level	http://www.stork.gov.eu/1.0/citizenQAALevel	Numeric {1:4}	Level between 1 and 4
Fiscal Number	http://www.stork.gov.eu/1.0/fiscalNumber	UTF-8	

Table 26: Stork Data definitions

All attribute name formats are "urn:oasis:names:tc:SAML:2.0:attrname-format:uri"

6.4 Additional Attribute Definitions

6.4.1 CanonicalResidenceAddress

```
<element                                    name="canonicalResidenceAddress"
type="stork:canonicalResidenceAddressType"/>
    <complexType name="canonicalResidenseAddressType">
    <sequence>
    <element ref="stork:countryCodeAddress"/>
      <element name="state" type="string" minOccurs="0"/>

      <element name="municipalityCode" type="string" minOccurs="0"/>
      <element name="town" type="string"/>
      <element name="postalCode" type="string"/>
      <element name="streetName" type="string"/>
      <element name="streetNumber" type="string" minOccurs="0"/>
      <element name="apartmentNumber" type="string" minOccurs="0"/>
      </sequence>
    </complexType>

      <element                                    name="countryCodeAddress"
type="stork:countryCodeAddressType"/>
      <simpleType name="countryCodeAddressType">
      <restriction base="string">
            <maxLength value="2"/>
            <minLength value="2"/>
      </restriction>
      </simpleType>
```

6.5 Create-Signature Request/Response

Required: Optional

The STORK-Interface provides an optional Create-Signature-method for signing arbitrary data using the citizens' certificate based eID token (or signature device in general). Thanks to this method a SP (through S-PEPS or V-IDP or directly) is able to call this method provided by the citizen's C-PEPS or MW.

In order to not introduce a new class of request/response dialogues at the STORK interface, this Create-Signature method is wrapped in a STORK attribute request. The returned signed document is the attribute-value. The method call is indicated by requesting the STORK attribute named `http://www.stork.gov.eu/1.0/signedDoc`.

If the Create-Signature method is not provided by C-PEPS, MW etc., for example because the citizen's eID token is not based on qualified certificates, the C-PEPS or MW will prompt this attribute request with stork:AttributeStatus "NotAvailable". On the other hand, the positive case could be as follows (in the case of a PEPS-PEPS scenario – see Figure 8):

1. The SP in MS-A asks its own S-PEPS to handle the Create-Signature-request by sending the according STORK attribute request.

2. As the citizen comes from MS B and MS B is a PEPS-country, the Create-Signature – request will be forwarded to the C-PEPS of MS B.

3. Assuming that MS B supports electronic signatures, the citizen is asked to sign the document. This can be technically done by, for example, presenting an ActiveX®-component, using a Java-Applet™ or communicating with a piece of proprietary software (this is finally Member State specific). However, these details are out of scope and are Member State specific.

4. The created signature will be conveyed back to the requesting service (through the C-PEPS or VIDP via the S-PEPS to the requesting SP) in form of an attribute value.

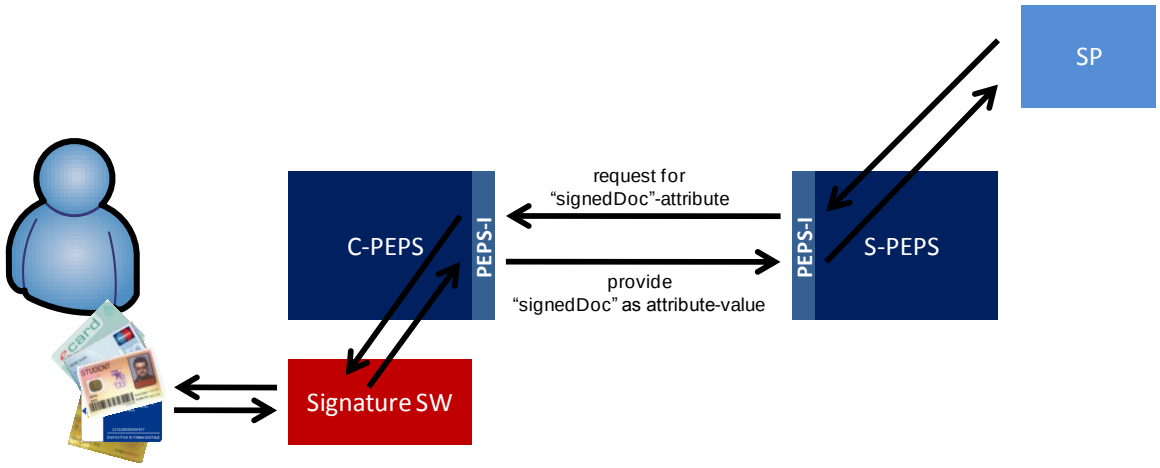

Figure 8: Simple flow of a Create-Signature request/response in a PEPS-PEPS scenario.

6.5.1 Request for Creating a Signature

The request for creating a signature is wrapped inside the <stork:RequestedAttribute> Element. The Create-Signature request itself follows the OASIS Digital Signature Service (DSS) Core Protocol [8] and is given within the <stork:AttributeValue>-Element.

Example of a DSS based Sign Request inside a STORK attribute request (relevant parts only):

```
...
<stork:RequestedAttribute
Name="http://www.stork.gov.eu/1.0/signedDoc"
NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri"
isRequired="false"
 <saml:AttributeValue>
 <dss:SignRequest>...</dss:SignRequest>
 </saml:AttributeValue>
</stork:RequestedAttribute>
...
```

The DSS SignRequest SHOULD NOT require the signatory to follow a special DSS profile.

The only variable required within a SignRequest is the content to be signed. The content to be signed SHOULD be given as base-64 encoded data object. In order to enable the signatory to view the content to be signed before signing, the MIME-type[1] of the content MUST be given along with the content.

If the content to be signed is given as base 64 encoded binary data as recommended above, the signature creation device (i.e. the consumer of the DSS request) is not required to process the

[1] Note that MIME-types other than text/plain may allow for insecure content, like scripting. Please make sure that the viewer of the content doesn't support such insecure facilities

content to be signed before signing. According to the DSS standard, the given base-64 content has to be signed as it is (i.e. according to 3.3.4 of DSS [8]: "*No transforms or other changes are made to the octet string before hashing*"). In this case, the preparation of the content to be signed MUST be done by the requesting service provider, i.e. by the service which created the SignRequest. Also the language selection MUST be done before the creation of the content to be signed. Based on the reference process flow of STORK, the country-selection—which implies the language as well—has to be happened before any attribute is requested. Thus the service requesting a signature is already aware of the preferred language of the user and is able to provide the according content to be signed.

Figure 9 gives the reduced scheme of a DSS SignRequest as it shows the selected required elements of a DSS SignRequest only. The DSS SignRequest given in Figure 9 SHOULD be used within the STORK AttributeValue element. The specification of the elements is given in [8].

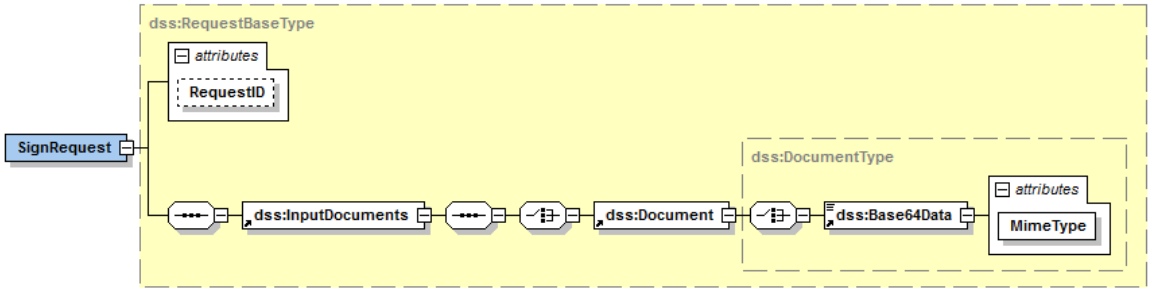

Figure 9: Schema of the DSS SignRequest to be used in STORK

The following listing is an example of a STORK Create-Signature request. In this example, the requestor asks the signatory to sign the plain text "Hello World".

```
<stork:RequestedAttribute
Name="http://www.stork.gov.eu/1.0/signedDoc"
NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri"
isRequired="false"
<saml:AttributeValue>
<dss:SignRequest      xmlns:dss="urn:oasis:names:tc:dss:1.0:core:schema"
RequestID="123456">
      <dss:InputDocuments>
            <dss:Document>
<dss:Base64Data MimeType="text/plain">SGVsbG8gV29ybGQ=</dss:Base64Data>
            </dss:Document>
      </dss:InputDocuments>
</dss:SignRequest>
</saml:AttributeValue>
</stork:RequestedAttribute>
```

6.5.2 Response

The created signature is an attribute-value within a SAML AttributeStatement as it is used at the STORK interfaces. The created signature MUST be put into a SAML AttributeValue element.

Example (relevant parts only):

. . .

```
<saml2:Attribute Name=" http://www.stork.gov.eu/1.0/signedDoc"
NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri">
<saml2:AttributeValue xsi:type="xs:string">
<dss:SignResponse>...</dss:SignResponse>
</saml2:AttributeValue>
</saml2:Attribute>
...
```

Note that although the SignResponse element is of a complex XML type (i.e. DSS SignResponse type) the AttributeValue may be indicated as a xs:string. The xs:string type is kept in order to ensure backwards compatibility.

The created signature MUST be given within the AttributeValue element in form of a DSS SignResponse element. Similar to the DSS SignRequest, the created signature SHOULD be given in base-64 encoded form within the dss:Base64Signature element. In addition to this, the type-attribute of the dss:Base64Signature element MUST indicate the type of the created signature as defined in section 7.1 of [8].

The format and type of the created signature is a decision of the Member State and depends on the common national infrastructure.

Requirements for the signature to be created:

The created signature MUST be an XML Digital Signature (XML-DSig [7]). This corresponds to DSS type-URI `urn:ietf:rfc:3275` (see 7.1 in [8]).

The created signature SHOULD be a XAdES signature (preferably XAdES BES).

The created signature MUST be an Enveloping Signature according to the definition given in [7].

The created signature MUST be an advanced electronic signature based on a qualified certificate according to the EU Signature Directive 93/1999/EC.

Figure 10 gives the reduced schema of a DSS SignResponse as it shows selected required elements only. The scheme given in Figure 10 SHOULD be used for providing signatures within the AttributeValue element of a STORK AttributeStatement. The elements and attributes are defined in [8] and should contain values according the DSS specification.

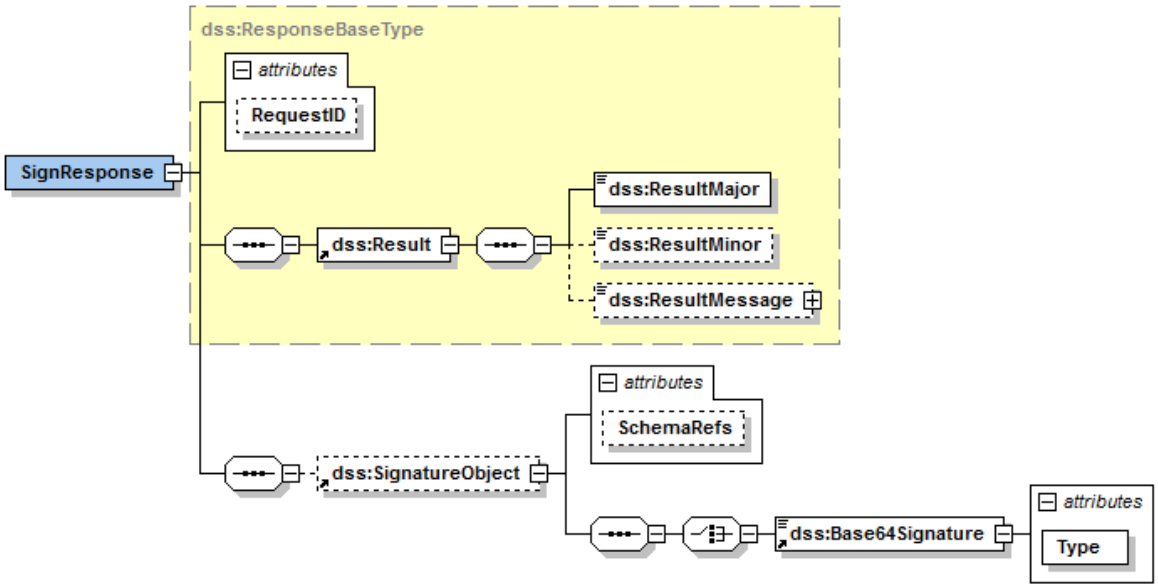

Figure 10: Schema of the DSS SignResponse to be used in STORK

The following listing is a fully fletched example of a STORK signature response. In this example, the requestor asked the signatory to sign the plain text "Hello World". In exchange, this AttributeStatement contains an enveloping XML Signature (XAdES BES).

```
<saml2:Attribute Name=" http://www.stork.gov.eu/1.0/signedDoc"
NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri">
<saml2:AttributeValue xsi:type="xs:string">
<dss:SignResponse xmlns:dss="urn:oasis:names:tc:dss:1.0:core:schema"
RequestID="123456">
        <dss:Result>
        <dss:ResultMajor>urn:oasis:names:tc:dss:1.0:resultmajor:Success<
/dss:ResultMajor><dss:ResultMinor>urn:oasis:names:tc:dss:1.0:resultminor
:valid:signature:OnAllDocuments</dss:ResultMinor>

      </dss:Result>
      <dss:SignatureObject>
            <dss:Base64Signature
Type="urn:ietf:rfc:3275">lcz48L3hhZGVzOlF1YWxpZnlpbmdQcm9wZXJ0aWVzPjwvZH
.....
NpZzpPYmplY3Q+PC9kc2lnOlNpZ25hdHVyZT48L3NsOkNyZWF0ZVhNTFNpZ25hdHVyZVJlc3
BvbnNlPg==</dss:Base64Signature>
      </dss:SignatureObject>
</dss:SignResponse>
</saml2:AttributeValue>
</saml2:Attribute>
```

7 STORK errors

7.1 Introduction

Most error codes are generated by the C-PEPS, and (also) shown to the user by the Service Provider. As normally both websites use different languages, for the end user it's weird experience to see different languages on one page.

This chapter provides a means of semantic error handling which allows service providers to show the messages in national language, translated by their S-PEPS. Thus the error code must be exchanged for which the first 7 characters of the error message (see 5.2.5, "StatusMessage") are used, 6 for the error code, followed by a hyphen (-). For cross border usage, it's recommended to add after this code the text of the message of the sender, in national language or English.

This functionality is implemented by the common functionalities, as specified in D5.8.3a and c.

The error code formats defined in the *D.5.8.3a Software Architecture Design* deliverable, were specific to the PEPS model to allow the easy identification of the given error in the Sequence AU diagrams (please see 2.3.1.1.3 and 2.3.1.1.4 sub-sections for S-PEPS related errors and 2.3.2.1.1 and 2.3.2.1.2 sub-sections for C-PEPS related errors). These errors were meant to be internal to the system, whereas the errors described in this chapter are designed to be used by all system in the STORK platform

7.2 Structure of codes

For cross border usage a new error code format was created, in order to uniformise the error code format to both PEPS and VIdP components. Three error lists were defined and the following 3 main error code formats were created:

- 0xxyyy
  - "0" identifies a specific PEPS error;
  - "xx" it's a two-digit number between 0 and 99 to identify the type of error – e.g. "01" to S-PEPS related error codes;
  - "yyy" it's a three-digit number between 0 and 999 to identify the error;
- 1xxyyyy
  - "1" identifies a specific VIdP error;
  - "x" it's a two-digit number between 0 and 99 to identify the type of error – e.g. "03" to SPWare related errors codes;
  - "yyy" it's a three-digit number between 0 and 999 to identify the error;
- 2xxyyy
  - "2" identifies a common error;
  - "xx" it's a two-digit number between 0 and 99 to identify the type of error – e.g. "00" to SP related errors codes;
  - "yyy" it's a three-digit number to identify the error;

The new error format allows an easier identification not only of the component that generates the error but also of the type of error (e.g. a specific VIdP error with a SPWare related error code will have the format "303y").

7.3 Specific PEPS Errors

Error Code	Message	Description	Client's reaction
	SP related error codes		
000001	QAA Level are missing or max PEPS level is lower than requested QAA level	The QAA Level is either missing from Service Provider Request (SP) or has a higher value than allowed;	3
000002	SP ID is missing or was not found in SP List	The SP ID is either missing or not found in the trusted Service Provider list.	3
000003	SP domain is not trusted or the SP Return URL is not valid	The SP domain is not trusted or the SP return URL is not valid.	3
000004	Request number is greater than or equal to the maximum number allowed.	The number of requests made by the citizen is higher than allowed.	3
000005	Selected Country is either invalid or not found.	The Citizen's country selected is either invalid or was not found.	3
000006	Attribute List is missing	The Personal Attribute list is missing on SP request.	3
000007	SP not allowed to access requested attribute.	Service Provider Requested an attribute that wasn't granted access.	3
	S-PEPS related error codes		
001001	Request number is greater than or equal to the maximum number	The number of requests made by the citizen is higher than allowed.	3
	C-PEPS related error codes		
	Generic error codes		
003001	Invalid Citizen Remote Address	The HTTP Header "remoteAddr" is either missing or has invalid value.	3
003002	Authentication Failed	Citizen's authentication failed.	3

Table 27 - Specific PEPS errors

7.4 Specific VIdP Errors

Error Code	Message	Description	Client's reaction
	SP related error codes		
100001	Invalid TransactionID	Invalid TransactionID	3
100002	TransactionID not Found	TransactionID not found	6
100003	Transaction Ended	Transaction in finished state	6
100004	Missing STORK Assertion Parameter Value	Missing STORK Assertion Parameter has value	3
100005	Missing STORK Assertion parameter	Missing STORK Assertion Parameter in response	6
	S-PEPS related error codes		
101001	S-PEPS not Found	S-PEPS not found	
101002	S-PEPS not enabled	S-PEPS not in enabled state	
	C-PEPS related error codes		
102001	Invalid C-PEPSID	No matching with adopted pattern	
102002	C-PEPS not Found	C-PEPS not found	
102003	Missing C-PEPSID Parameter	Request has not C-PEPS parameter	
102004	Missing C-PEPSID Value	C-PEPS parameter has not value	
102005	Failed connection to C-PEPS	Failed connection to C-PEPS	
102006	C-PEPS General Application Error	C-PEPS General Application Error	
102007	Failed Authentication to C-PEPS	Failed Authentication to C-PEPS	
	SPWare		
103001	Unkonw SPWare	Unkonw SPWare	
103002	Not Enabled	SPWare Not Enabled	
103003	Can't connect to SPWare	Can't connect to SPWare	
103004	SPWare Configuration Error	SPWare Configuration Error	
	SPWare-Memberstate IDP		

Error Code	Message	Description	Client's reaction
104001	Failed connection to MS-IDP	Failed connection to MS-IDP (eg. IDService)	
104002	Application error from MS-IDP	Application error from MS-IDP	
104003	SPWare-MS-IDP Connection timeout	SPWare-MS-IDP Connection timeout	
104004	MS IDP General Application Error	MS IDP General Application Error	
104005	Invalid MS IDP URL	Configured URL is Invalid	
	MS IDP-Client (BC)		
105001	MS-IDP-BC Connection timeout between IDP and BC	MS-IDP-BC Connection timeout between IDP	
105002	BC can't connect to MS IDP	BC can't connect to MS IDP	
105003	BC is not started	BC is not started	
105004	User cancelled authentication	User cancelled authentication	
105005	User denied attribute retrieval	User denied attribute retrieval	
	SP Account related error codes		
106001	Invalid SP Account	SP pattern is used else default pattern	3
106002	Invalid Password	Password invalid	3
106003	Account not Found	Account not found	6
106004	Blocked Account	Account in blocked state	6
106005	Disabled Account	Account in disabled state	6
106006	Deactivated Account	Account in deactivated state	6
106007	Suspended Account	Account in suspended state	6
106008	Deleted Account	Account in deleted state	6
106009	Missing Username Parameter	Missing Username Parameter in request	3
106010	Missing Password Parameter	Missing Password Parameter in request	3
106011	Missing Username Parameter Value	Username Parameter has no value	3
106012	Missing Password Parameter Value	Password Parameter has no value	3
106013	Unsupported Account State	Account state not defined in Spec	4

Error Code	Message	Description	Client's reaction
	General Communication		
107001	Session Ended	Session in End state	6
107002	Invalid request	Request is general invalid based on technical spec	3
107003	General API Error	General API	1
107004	Service not available	Service not reachable	1
107005	Invalid Timestamp	Timestamp not supported	4
107006	Missing Timestamp Parameter	No timestamp element in request	3
107007	Invalid TransactionId	Invalid transactionId	4
107008	Missing TransactionId Parameter	Missing TransactionId Parameter	3
107009	Invalid Policy Version	Invalid policy version. Not supported by SPAuthenticationService	1
107010	Missing Policy Version Parameter	No policy version element in request	3
107011	Missing Version Parameter Value	No policy version value in version element missing	3
107012	Invalid PolicyURL	Invalid PolicyURL	3
107013	Missing PolicyURL Parameter	Missing PolicyURL Parameter	3
107014	Missing PolicyURL Parameter Value	Missing PolicyURL Parameter Value	3
	Authentication and Authorization by SP at Middleware		
108001	SP Failed Authentication	General authentication failed	1
108002	SP Failed Authorization	General authorization failed	1
108003	SP not Found	SP not found	1
108004	SP Restriction to Service	SP not allowed to use the service called	2
108005	SP Restriction to Service Method	SP not allowed to use the service method called	2
108006	Missing SP URL for Notifications	No SP notification URL configured	1
108007	Access Denied	Access denied	1
	PEPSConnector		
109001	CCC not specified	No citizen country code specified	3
109002	PEPS country not supported	PEPS country not supported	6

Error Code	Message	Description	Client's reaction
109003	No PEPS destination found	No PEPS destination found	1
109004	No AssertionConsumerService URL found	ACS URL of PEPSConnector not defined	1
109005	No issuer name for PEPSConnector configured	PEPSConnector issuer name not configured	1
109006	Error signing request	Credentials for signing AuthnRequest not found or signing error	1
109007	Error building StartAuthResponse	StartAuthResponse cannot be built	1
109008	C-PEPS unknown	C-PEPS unknown	3
109009	C-PEPS response not valid	C-PEPS response not valid	6
109010	C-PEPS assertion not valid	C-PEPS assertion not valid	6
109011	No response stored for sessionID	No response stored for sessionID	0
	SPWare AT		
110001	Error providing BKU selection	Error providing BKU selection	1
110002	Error retrieving authentication data from MOA-ID	Error retrieving authentication data from MOA-ID	1
110003	Error building STORK response	Error building STORK response	1
	VIDP and Resources		
111001	Transaction Timeout	Transaction timeout	1
111002	Internal Error	Internal VIDP error	1
111003	VIDP Error	VIDP application error	1

Table 28 - Specific VIdP errors

7.5 Common Errors (PEPS/VIdP)

Error Code	Message	Description	Client's reaction
	SP related error codes		
200001	QAA Level is missing or max PEPS level is lower than requested QAA level	The SP QAA Level is either missing from Service Provider Request (SP) or the has a higher value than allowed QAA Level for the SP	6
200002	SP ID is missing or not found in SP List	The SP ID is either missing or not found in the trusted Service Provider list.	6
200003	SP domain is not trusted.	The SP domain is not trusted.	6
200004	Selected country is not a valid country.	The selected country is either missing or invalid.	3
200005	SP not allowed to access requested attributes.	The SP is requesting an attribute that is not authorized to.	3
200006	The SAML Request Token is missing or invalid.	The SP SAML Request is either missing or invalid.	6
200007	The server could not identify the Service Provider.	The SP was not successfully identified using the information sent by him.	6
200008	Invalid SP Return URL	The SP return URL (assertion consumer URL) is invalid.	6
200009	Invalid Relay State.	The SP's Relay State parameter size is higher than allowed.	6
200010	Error building the SAML Response	An error occurs on the SAML token response generation.	6
	S-PEPS/VIdP (as "S-PEPS") related error codes		
201001	Couldn't get the Assertion Consumer URL.	Couldn't get the Assertion Consumer URL from configuration file.	6
201002	Error building the SAML Request.	An error occurs on the SAML token request generation.	6
201002	Invalid SAML Request token.	The SAML Request sent by S-PEPS/VIdP is invalid	6
201003	Origin is not correct.	The Assertion Consumer URL on the SAML Request (sent by S-PEPS) is	6
201004	SAML Conditions are not complaint.	The SAML Request conditions are not complaint.	0
	C-PEPS/VIdP (as "C-PEPS") related error codes		
202001	Invalid Destination URL	The C-PEPS/VIdP SAML Request (from S-PEPS) has higher value than allowed.	6
202002	Invalid SAML Response token.	The SAML response sent by the C-PEPS/VIdP is invalid.	6
202003	Invalid Issuer URL	The issuer is either null or is invalid.	6
202004	Max QAA Level if lower than requested.	The C-PEPS/(VIdP) Max QAA Level is lower than requested.	3
202005	Invalid Attribute List	The SAML Request's Attribute List is either empty or size higher than allowed.	3

Error Code	Message	Description	Client's reaction
202006	Citizen consent is malformed.	The response from Citizen is malformed.	0
202007	Consent not given for a mandatory attribute	Citizen not given consent for a mandatory attribute.	0
202008	Authentication Failed.	The Authentication Failed.	0
202009	Mandatory Attribute Value not found.	A mandatory attribute value not found.	0
202010	Mandatory Attribute not found.	A mandatory attribute not found	0
202011	Error building the SAML Response.	The response from Citizen is malformed.	0
202012	Citizen consent not given.	The citizen not given his consent to send attribute values	0
	Generic error codes		
203001	Generic Error	A generic application error occurred.	3
203002	Missing Parameter	Missing parameter in request	3
203003	Invalid Parameter	Included parameter not supported	3
203004	Missing Parameter Value	Parameter has not value	3
203005	Missing Configuration	Missing configuration.	3
203006	Invalid configuration	Invalid configuration found.	3
203007	Missing Configuration Value	Configuration has not value	3
203008	Invalid Stork Attribute Value	Invalid Stork attribute value	6
203009	Missing Stork Attribute Value	Stork attribute value is missing	6
203010	Missing Session ID	The session id is missing	3
203011	Invalid Session ID	The session id is invalid	3

Table 29 - Common (VIdP/PEPS) errors

7.6 Citizen Reaction Codes

The following table presents a set of expected Citizens' reactions that should be taken by them.

Clients Reaction	Description
0	Citizen should try Again
1	Citizen should call Service (PEPS or VIDP) Support
2	Citizen should contact Service (PEPS or VIDP) Customer Service
3	Try Again with right parameters and values
4	See VIDP-WS API Specification Document
5	No reaction
6	No Retry

Table 30 - Citizens' Reaction Codes

Please note that Citizen Reaction Codes are the ones included in the three previous tables; they are not transmitted explicitly from one system to others.

8 Version Control

The STORK project has established a platform of many systems which provide the interoperability of eIDs in Europe. This platform can only work correctly if all systems use similar or compatible configurations and software. This condition explains that version control is a key factor for success: if any problem occurs this will be a first indication of the cause of the error.

Version control is a requirement for the cross border functioning of the STORK platform, and is also very useful, thus highly recommended between the national STORK system and its connected service providers. Although this last point is a responsibility of each STORK connected member state, in section 8.2 this document proposes such a version control file.

The STORK project supplies the automated tools to be personalised in order to generate the version control file, according to the following specifications.

8.1 Version control file in each PEPS or V-IDP

In order to facilitate the control of the configuration and software version as well as the evolution in functionalities of a PEPS/V-IDP, other member states need to know some information about the running version to foreigner PEPS/V-IDP:

1. Environment (prod/pre-prod/test)
2. Software version
3. SAML specification version accepted (income) and used (outcome)
4. Last modification date (including configuration)
5. Max. QAA level provided
6. Attributes provided
7. URL of the available services: C-PEPS, S-PEPS, OCSP responder
8. Sign certificate (current and future).
9. Only this file URL must be published; all other URL can be read from this file.

The schema for this version file is the following one:

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:ds="http://www.w3.org/2000/09/xmldsig#" version="1">

  <xs:import namespace="http://www.w3.org/2000/09/xmldsig#"
schemaLocation="http://www.w3.org/TR/xmlndsig-core/xmldsig-core-
schema.xsd"/>

  <xs:element name="stork-version-info" type="StorkVersionInfoType" />

  <xs:complexType name="StorkVersionInfoType">
    <xs:sequence>
      <xs:element name="GenerationDate" type="xs:date"/>
      <xs:element name="countries" type="CountriesType"/>
      <xs:element ref="ds:Signature" minOccurs="0"/>
```

```
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="CountriesType">
    <xs:sequence maxOccurs="unbounded">
      <xs:element name="country" type="CountryType"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="CountryType">
    <xs:sequence>
      <xs:element name="ID" type="CountryCodeType"/>
      <xs:element name="Name" type="xs:string" minOccurs="0"/>
      <xs:element name="environments" type="EnvironmentsType"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="EnvironmentsType">
    <xs:sequence>
      <xs:element name="prod" type="EnvironmentType"/>
      <xs:element name="pre-prod" type="EnvironmentType"/>
      <xs:element name="test" type="EnvironmentType"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="EnvironmentType">
    <xs:sequence minOccurs="0">
      <xs:element name="SoftVersion" type="xs:string"/>
      <xs:element name="SAMLSpecVersion-accept" type="xs:string"/>
      <xs:element name="SAMLSpecVersion-send" type="xs:string"/>
      <xs:element name="SAMLSpecKnownIssues" type="xs:string"
minOccurs="0"/>
      <xs:element name="LastModif" type="xs:date"/>
      <xs:element name="maxQAA" type="QAALevelType"/>
      <xs:element name="colleague-url" type="xs:anyURI" minOccurs="0"/>
      <xs:element name="SP-url" type="xs:anyURI" minOccurs="0"/>
      <xs:element name="colleague-ocsp-url" type="xs:anyURI"
minOccurs="0"/>
      <xs:element name="SP-ocsp-url" type="xs:anyURI" minOccurs="0"/>
      <xs:element name="Information" type="xs:string" minOccurs="0"/>
      <xs:element name="attributes" type="AttributesType"/>
      <xs:element name="certificates" type="CertificateType"/>
    </xs:sequence>
```

```
  </xs:complexType>

  <xs:simpleType name="CountryCodeType">
    <xs:restriction base="xs:token">
      <xs:pattern value="[A-Z]{2}"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="QAALevelType">
    <xs:restriction base="xs:integer">
      <xs:minInclusive value="1"/>
      <xs:maxInclusive value="4"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:complexType name="AttributesType">
    <xs:sequence minOccurs="0" maxOccurs="unbounded">
      <xs:element name="attribute" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="CertificateType">
    <xs:sequence>
      <xs:element ref="ds:X509Certificate/>
         <xs:element name="UpcomingCertificate"
type="UpcomingCertificateType"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="UpcomingCertificateType">
    <xs:sequence minOccurs="0">
        <xs:element name="AvailableFrom" type="xs:date"/>
        <xs:element ref="ds:X509Certificate/>
      </xs:sequence>
  </xs:complexType>

</xs:schema>
```

SAMLSpecKnownIssues contains a comma-delimited list of identifiers related to known problems identified and referenced by the Change Control Supervisory Board. These indicate that some well known deviance from the supported specification version is present in the deployed version.

Information is an optional free text field containing any specific info that may be relevant for partners.

Attribute contains the names of supported attributes (only the name, not the full name).

Dates are in ISO 8601 format "YYYY-MM-DD".

SAMLSpecVersion-send can never be higher than ***SAMLSpecVersion-accept***

An XML Signature authenticates the requestor (PEPS or VIdP) and ensures integrity.

The signature must be an enveloped signature and applied to the <stork-version-info> element and all its children.

<ds:Signature> is defined in http://www.w3.org/TR/xmldsig-core/.

Certificate used for the XML sign must be the one currently known by Colleagues (production Sign Certificate if the file contains Production-environment information, test Sign Certificate otherwise).

Here is an example of such a file:

```
<?xml version="1.0" encoding="UTF-8"?>
<stork-version-info
  <GenerationDate>2011-03-30+02:00</GenerationDate>
  <countries>
    <country>
      <ID>BE</ID>
      <Name></Name>
      <Environments>
        <test>
         <SoftVersion>0.96</SoftVersion>
         <SAMLSpecVersion-accept>0.5.1</SAMLSpecVersion-accept>
         <SAMLSpecVersion-send>0.5.1</SAMLSpecVersion-send>
         <SAMLSpecKnownIssues>KIAT25,KISP23</SAMLSpecKnownIssues>
         <LastModif>2011-03-30+02:00</LastModif>
         <maxQAA>4</maxQAA>
         <colleague-url>https://.../sp-reqcolleague</colleague-url>
         <SP-url>https://.../peps/sp-request</SP-url>
         <colleague-ocsp-url>http://ocsp.belgium.be</colleague-ocsp-url>
         <SP-ocsp-url>http://ocsp.belgium.be</SP-ocsp-url>
         <Information>
                 Real eID cards are accepted in test environment
              but personal information (national identifier,
              birth date, etc.) will be replaced by test (fake) one.
         </Information>
         <attributes>
            <attribute>givenName</attribute>
            <attribute>eIdentifier</attribute>
            [. . .]
         </attributes>
         <certificates>
           <Certificate>
             <ds:X509Certificate
                          xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
                 MIIEXTCCA0WgAwIBAgILAQAAAAABKKZaFSWEWFSDFWEFWDFSDFWEFS
                 [...]
                 ezIp1sco52HRU+R+4uMbPSDFPSDFPSDFPSDFPSDFPSDFPSDFPSHME=
             </ds:X509Certificate>
             <UpcomingCertificate>
                <AvailableFrom>2011-10-01+02:00</AvailableFrom>
                <ds:X509Certificate
                          xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
                   MCDSDSDWEFWERTHERIIEXTCCA0SDDGESERGSERGSERGSEWgAwIBA
                   [...]
                   RTWDFFACDFACADFASADFACDFADCDAFACDAFADFADCGSDFG%SGFG=
```

```
                </ds:X509Certificate>
              </UpcomingCertificate>
            </Certificate>
          </certificates>
        </test>
      </Environments>
    </country>
  </countries>
  <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
    [...]
  </ds:Signature>
</stork-version-info>
```

8.2 Service Providers

Publishing data of different countries to service providers gives them information on when to update their country selector accordingly.

When publishing toward service providers, the global file can easily be created by concatenating all files coming from all countries (without the header). However, some fields from foreign countries are usually not relevant for a Service Provider, as it only needs to know about foreign functionalities, technical problems are under the responsibility of the PEPS.

These fields should be removed:

- SoftVersion
- SAMLSpecVersion-accept
- SAMLSpecVersion-send
- SAMLSpecKnownIssues
- LastModif
- All URLs
- Certificates
- Generation date

The location of this file will be published to Service Providers. The URL must use the HTTPS scheme to protect the file.

Here is an example of such a file:

```
<?xml version="1.0" encoding="UTF-8"?>
<stork-version-info>
  <GenerationDate>2011-03-30+02:00</GenerationDate>
  <countries>
    <country>
      <ID>BE</ID>
      <Name></Name>
      <Environments>
        <prod>
          <maxQAA>4</maxQAA>
          <Information>
            Real eID cards are accepted in test environment but
            personal information (national identifier, birth date,
            etc.) will be replaced by test (fake) one
          </Information>
          <attributes>
            <attribute>age</attribute>
            <attribute>canonicalResidenceAddress</attribute>
            <attribute>citizenQAALevel</attribute>
            <attribute>countryCodeOfBirth</attribute>
            <attribute>dateOfBirth</attribute>
            <attribute>fiscalNumber</attribute>
            <attribute>gender</attribute>
            <attribute>givenName</attribute>
            <attribute>eIdentifier</attribute>
            <attribute>isAgeOver</attribute>
            <attribute>maritalStatus</attribute>
            <attribute>nationalityCode</attribute>
            <attribute>surname</attribute>
            <attribute>textResidenceAddress</attribute>
          </attributes>
        </prod>
      </Environments>
    </country>
    <country>
      <ID>ES</ID>
      <Name>España</Name>
      <Environments>
        <prod>
          <maxQAA>4</maxQAA>
          <Information></Information>
          <attributes>
            <attribute>age</attribute>
            <attribute>citizenQAALevel</attribute>
```

```
            <attribute>dateOfBirth</attribute>
            <attribute>fiscalNumber</attribute>
            <attribute>gender</attribute>
            <attribute>givenName</attribute>
            <attribute>eIdentifier</attribute>
          </attributes>
        </prod>
      </Environments>
    </country>
  </countries>
  <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
    [...]
  </ds:Signature>
</stork-version-info>
```

9 STORK Schema

9.1 Introduction

This section describes the formal schema of the STORK request and reply in such a way that they can be checked by automated tools like XMLpad (http://www.wmhelp.com/xmlpad3.htm). Thus syntactical analysis is provided easily.

Nevertheless, as this protocol is meant to be extended, the specification of the behaviour in case of unrecognised attributes in 6.1.4, 6.1.4.8.1 and 6.3.8.1 are prevalent above this xml specification.

9.2 STORK extensions

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
  elementFormDefault="qualified"
  targetNamespace="urn:eu:stork:names:tc:STORK:1.0:assertion"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:storkp="urn:eu:stork:names:tc:STORK:1.0:protocol"
  xmlns:stork="urn:eu:stork:names:tc:STORK:1.0:assertion">

  <xs:element name="QualityAuthenticationAssuranceLevel"
type="stork:QualityAuthenticationAssuranceLevelType" />
  <xs:element name="spSector" type="stork:SPSectorType" />
  <xs:element name="spApplication" type="stork:SPApplicationType"/>
  <xs:element name="spCountry" type="stork:CountryCodeType"/>
  <xs:element name="CitizenCountryCode" type="stork:CountryCodeType" />
  <xs:element name="RequestedAttribute"
type="stork:RequestedAttributeType" />
  <xs:element name="AttributeValue" type="xs:anyType" />
  <xs:element name="canonicalResidenceAddress"
type="stork:canonicalResidenceAddressType"/>
  <xs:element name="countryCodeAddress" type="stork:CountryCodeType"/>

  <xs:attribute name="AttributeStatus" type="stork:AttributeStatusType"
/>


  <xs:simpleType name="SPSectorType">
    <xs:restriction base="xs:string">
      <xs:minLength value="1" />
      <xs:maxLength value="20" />
    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="SPApplicationType">
    <xs:restriction base="xs:string">
      <xs:minLength value="1" />
      <xs:maxLength value="100" />
    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="AttributeStatusType">
    <xs:restriction base="xs:string">
      <xs:enumeration value="Available" />
      <xs:enumeration value="NotAvailable" />
      <xs:enumeration value="Withheld" />
```

```
    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="QualityAuthenticationAssuranceLevelType">
    <xs:restriction base="xs:integer">
      <xs:minInclusive value="1" />
      <xs:maxInclusive value="4" />
    </xs:restriction>
  </xs:simpleType>

  <xs:complexType name="canonicalResidenceAddressType">
    <xs:sequence>
      <xs:element name="countryCodeAddress" type="stork:CountryCodeType"
/>
      <xs:element name="state" type="xs:string" minOccurs="0"/>

      <xs:element name="municipalityCode" type="xs:string"
minOccurs="0"/>
      <xs:element name="town" type="xs:string"/>
      <xs:element name="postalCode" type="xs:string"/>
      <xs:element name="streetName" type="xs:string"/>
      <xs:element name="streetNumber" type="xs:string" minOccurs="0"/>
      <xs:element name="apartmentNumber" type="xs:string"
minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>

  <xs:simpleType name="CountryCodeType">
    <xs:restriction base="xs:token">
      <xs:pattern value="[A-Z]{2}"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:complexType name="RequestedAttributeType">
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="unbounded"
ref="stork:AttributeValue"/>
    </xs:sequence>
    <xs:attribute name="Name" use="required" type="xs:string"/>
    <xs:attribute name="NameFormat" use="required" type="xs:anyURI"/>
    <xs:attribute name="FriendlyName" use="optional" type="xs:string"/>
    <xs:attribute name="isRequired" use="optional" type="xs:boolean"/>
    <xs:anyAttribute namespace="##other" processContents="lax"/>
  </xs:complexType>
</xs:schema>
```

9.3 STORK protocol extensions

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
  elementFormDefault="qualified"
  targetNamespace="urn:eu:stork:names:tc:STORK:1.0:protocol"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:storkp="urn:eu:stork:names:tc:STORK:1.0:protocol"
  xmlns:stork="urn:eu:stork:names:tc:STORK:1.0:assertion"
  xmlns:ds="http://www.w3.org/2000/09/xmldsig#">

  <xs:import namespace="urn:eu:stork:names:tc:STORK:1.0:assertion"
schemaLocation="stork.xsd"/>
```

```
  <xs:import namespace="http://www.w3.org/2000/09/xmldsig#"
schemaLocation="http://www.w3.org/TR/2002/REC-xmldsig-core-
20020212/xmldsig-core-schema.xsd" />

  <xs:element name="eIDSectorShare" type="xs:boolean" default="false"/>
  <xs:element name="eIDCrossSectorShare" type="xs:boolean"
default="false"/>
  <xs:element name="eIDCrossBorderShare" type="xs:boolean"
default="false"/>
  <xs:element name="RequestedAttributes"
type="storkp:RequestedAttributesType" />
  <xs:element name="AuthenticationAttributes"
type="storkp:AuthenticationAttributesType" />

  <xs:complexType name="RequestedAttributesType">
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="unbounded"
ref="stork:RequestedAttribute"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="AuthenticationAttributesType">
    <xs:sequence>
      <xs:element name="VIDPAuthenticationAttributes"
type="storkp:VIDPAuthenticationAttributesType" minOccurs="0"
maxOccurs="1" />
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="VIDPAuthenticationAttributesType">
    <xs:sequence>
      <xs:element name="CitizenCountryCode" minOccurs="0" maxOccurs="1"
type="stork:CountryCodeType" />
      <xs:element name="SPInformation" minOccurs="1" maxOccurs="1"
type="storkp:SPInformationType"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="SPInformationType">
    <xs:sequence>
      <xs:element name="SPID" minOccurs="1" maxOccurs="1"
type="storkp:SPIDType" />
      <xs:element name="SPCertSig" minOccurs="0" maxOccurs="1"
type="storkp:SPCertSigType" />
      <xs:element name="SPCertEnc" minOccurs="0" maxOccurs="1"
type="storkp:SPCertEncType" />
      <xs:element name="SPAuthRequest" minOccurs="0" maxOccurs="1"
type="storkp:SPAuthRequestType"/>
    </xs:sequence>
  </xs:complexType>

  <xs:simpleType name="SPIDType">
    <xs:restriction base="xs:string">
      <xs:minLength value="1" />
      <xs:maxLength value="20" />
    </xs:restriction>
  </xs:simpleType>

  <xs:complexType name="SPCertSigType">
    <xs:sequence>
      <xs:element minOccurs="1" ref="ds:KeyInfo" />
```

```
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="SPCertEncType">
    <xs:sequence>
      <xs:element minOccurs="1" ref="ds:KeyInfo" />
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="SPAuthRequestType">
    <xs:sequence>
        <xs:any namespace="##other" processContents="lax"
maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:schema>
.
```

References

[1] *Stork Consortium. D5.8.3c Software Design.for PEPS architecture 2011.*

[2] *Assertions and Protocols for the OASIS Security Assertion Markup Languange (SAML) v2.0. OASIS Standard 15 March 2009. http://docs.oasis-open.org/security/saml/v2.0/saml-core-2.0-os.pdf*

[3] *Security Assertion Markup Language (SAML) v2.0 http://www.oasis-open.org/specs/index.php#saml*

[4] *Bindings for the OASIS Security Assertion Markup Language (SAML) V2.0.* OASIS Standard, March 2005. http://docs.oasis-open.org/security/saml/v2.0/saml-bindings-2.0-os.pdf

[5] *Profiles for the OASIS Security Assertion Markup Language (SAML) V2.0.* OASIS Standard, March 2005. http://docs.oasis-open.org/security/saml/v2.0/saml-profiles-2.0-os.pdf

[6] *SAML V2.0 Holder-of-Key Web Browser Profile Version 1.0 Committee Specification 01 29 July 2009 http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-holder-of-key-browser-sso.pdf*

[7] *XML Signature Syntax and Processing (Second Edition) World Wide Web Consortium Reccomendation, 10 June 2008, http://www.w3.org/TR/xmldsig-core/*

[8] *Digital Signature Service (DSS) Core Protocols, Elements, and Bindings Version 1.0. OASIS Standard, 11 April 2007. http://docs.oasis-open.org/dss/v1.0/oasis-dss-core-spec-v1.0-os.html*

[9] *Stork Consortium. D5.7.3 Functional Design for PEPS, MW models and interoperability, 2011*

[10] *Stork Consortium. D5.8.3e Software Design for MW architecture. 2011.*