

Draft – FOR INFORMATION ONLY

SAML 2.0 Profile for SSO in Danish Public Sector

*Author: Thomas Gundel,
IBM Crypto Competence Center Copenhagen*

Contents

1. Introduction.....	4
1.1 Prerequisites.....	4
1.2 Acknowledgements.....	4
2. Architectural Overview.....	5
3. SAML Profile Description.....	7
3.1 Profile Information.....	7
3.1.1 SAML References.....	7
3.2 Profile Overview.....	7
3.2.1 SSO Use Case Description.....	8
3.3 Profile Description.....	9
3.3.1 Identity Provider Sends Artifact.....	10
3.3.2 Service Provider Resolves Artifact.....	10
3.4 SAML Assertion Contents.....	11
3.4.1 Main Assertion Element.....	12
3.4.2 Issuer Element.....	12
3.4.3 Signature Element.....	13
3.4.4 Subject Element.....	13
3.4.5 Conditions Element.....	13
3.4.6 Advice Element.....	13
3.4.7 AuthnStatement Element.....	13
3.4.8 AttributeStatement Element.....	14
3.4.9 An Assertion Example.....	17
4. Profile Discussion.....	19
4.1 Identifiers for Service- and Identity Providers.....	19
4.2 Meta Data.....	19
4.2.1 Meta Data about services and service providers.....	20
4.2.2 Meta Data about authentication services and providers.....	20
4.2.3 Other meta data.....	21
4.3 Protocol and Binding Selection.....	21
4.3.1 Artifact Binding.....	21
4.4 Security Considerations.....	21
4.4.1 Transport Level Security.....	21
4.4.2 Signing and Encryption of SAML elements.....	21
4.4.3 Other Security Mechanisms.....	22
4.4.4 Securing Session Cookies.....	22

4.5 Error Handling.....	23
4.6 Governance and Management of the Profile.....	23
4.7 Privacy.....	24
4.7.1 Registration and use of private or personal data.....	24
4.7.2 Information and choices that must be given to the user.....	25
4.7.3 Transfer of private or personal data between Identity Provider and Service Provider.....	25
4.8 Single Logout.....	25
5. Implementation Considerations.....	27
5.1 Logical Security Architecture.....	27
5.2 SSO Scenario.....	28
5.3 Service Provider Tasks.....	29
5.3.1 SSO Handler Tasks.....	29
5.3.2 SAML Engine Tasks.....	29
5.3.3 Authentication Engine Tasks.....	29
5.3.4 Authorization Engine Tasks.....	30

1.Introduction

IT- og Telestyrelsen in Denmark has launched an initiative aiming for a common approach to authentication for E-Government in Denmark. The initiative is based on the E-Authentication Initiative from USA (<http://www.cio.gov/eauthentication>) which is part of the President's Management Agenda established to create trust and confidence in E-Government transactions.

The goal of the Danish initiative is to enable government organizations to use external authentication services instead of developing their own, simplify Single Sign-On (SSO) across disparate systems and establish a foundation for federated identity management. This will hopefully result in cost-reductions through re-use of authentication services, faster development cycles for E-Government applications, consistent application of security technology, and improved user experiences (via Single Sign-On).

To start the initiative, IT- og Telestyrelsen has produced a set of documents and published them for public hearing (which ended September 22. 2005). The base document [ITTArch] defines the overall architecture and the scenarios for Single Sign-On (SSO) to be supported. The architecture is based on the concept of federation and is technology-agnostic such that it can be implemented using different underlying technologies. This is a huge advantage since several different federation technologies currently exist in parallel and still undergo continuous development.

Separate technical specifications are thus needed to define how the architecture should be implemented with current federation technology. These specifications should ensure that implementations from different vendors will interoperate in real life. A key area will be definitions of the data formats and mechanisms used to exchange information about an end user. IT- og Telestyrelsen has decided to base the initial version on a SAML 2.0 profile tailored to Danish E-Government needs. The purpose of this document is to describe exactly such a profile. It further needs to be supplemented by interface specifications defining the hand-off mechanisms (e.g. cookies and HTTP parameters) between the entities in the architecture not covered by the SAML profile itself. These interface specifications are however not in scope for the present document.

This initial version of the SAML profile was created without supporting pilot implementations. It is however expected that subsequent pilot implementations could identify areas which need subsequent clarification or modification in order to achieve true interoperability by different commercial SAML implementations.

1.1Prerequisites

The reader is assumed to be familiar with SAML 2.0 and with the material from E-Authentication and IT- & Telestyrelsen regarding SSO architecture.

Good starting points for reading are [SAMLCore], [SAMLProf], [SAMLBind], [ITTArch], [EgovTechApp] and [EgovSAMLProf].

1.2Acknowledgements

Thanks to my colleague Hanne Søndergaard who contributed with the section on privacy and provided several helpful comments to the main text.

2. Architectural Overview

This chapter briefly describes the overall architecture for Single Sign-On and illustrates where the SAML profile fits in the overall picture.

Basically, the architecture consists of four entities [ITArch]:

- **E-Government Applications** (known as Agency Applications in E-Authentication) are offered by service providers to users online.
- **Authentication Services** (known as Credential Services in E-Authentication) are offered by identity providers. They handle user authentication and provide the user with credentials that can be used towards the applications.
- **An E-Authentication portal** which allows the end-user to locate and select E-Government applications and authentication services.
- **End Users:** These can be citizens, government employees, contractors and businesses, who authenticate to an application using a credential issued by an authentication service.

The figure below from [EGovTechApp] illustrates the basic SSO scenario where the user first goes to the authentication portal, selects application and authentication service to use, gets redirected to the authentication service (CS), and finally gets redirected to the application (AA):

□

Figure 1: Basic Use Case

ase echAppsment.ned ion of this hand-off that is specified with SAMLt the end user. rected to the application and aut



In the third step, the authentication service sends the application a credential containing authentication and identity information about the end user. It is the implementation of this hand-off that is specified using the SAML 2.0 profile defined in this document. Other federation technologies exist including Liberty Alliance, Shibboleth, and WS-Federation – but these will not (yet) be part of the Danish SSO architecture [ITTArc].

3.SAML Profile Description

This chapter defines a SAML V2.0 profile for Single Sign-On (SSO) in the Danish Public sector.

The profile is based on the SAML Web Browser SSO Profile described in [SAMLProf] and will subsequently be referred to as *DK-SAML*. It is a further specialization of the Web Browser SSO Profile designed to support the SSO architecture and scenarios described in [ITTArch] and further impose restrictions and limit options left open by the base SAML profile in order to ensure a high level of interoperability.

As described in previous chapters, this profile should be seen as part of a larger SSO architecture.

The profile text is aimed to be short and concise. Discussions of the rationale behind choices can be found in chapter 4.

3.1 Profile Information

Identification: urn:itt:xxx:yyy:zzz

Contact Information: saml-profile-comment@itst.dk

SAML Confirmation Method Identifiers: The SAML V2.0 "bearer" confirmation method identifier, urn:oasis:names:tc:SAML:2.0:cm:bearer, is used by this profile.

Description: Given below.

Updates: N/A

3.1.1 SAML References

In the following a set of references to the OASIS SAML 2.0 documents are given as a service to the reader. These references can be useful as background reading for the profile:

- The assertion structure and content is described in [SAMLCore] chapter 2.
- The SAML Artifact Resolution protocol is described in [SAMLCore] section 3.5.
- The SAML Artifact Binding is described in [SAMLBind] section 3.6.
- The SAML HTTP Redirect Binding is described in [SAMLBind] section 3.4.
- The Web Browser SSO Profile is described in [SAMLProf] section 4.1.
- The Artifact Resolution Profile is described in [SAMLProf] chapter 5.
- The X.500 / LDAP Attribute Profile is described in [SAMLProf] section 8.2. This profile is used for encoding LDAP attributes as SAML attributes.

3.2 Profile Overview

Figure 1 below illustrates the steps during SSO in the original SAML Web Browser SSO Profile. In DK-SAML only the last three steps (shown by the red rectangle) are relevant since selection of service provider and identity provider is performed by other means. In fact, the document [ITTArch] describes how the user selects the desired service- and identity providers via an authentication portal. The DK-SAML profile can be seen as the realization of the last step in each of the described scenarios in [ITTArch] where the identity provider performs a redirect to the service provider with the established identity information.

3.2.1 SSO Use Case Description

3.2.1.1 Actors

The use case covered by the profile has three actors:

- User Agent – a standard web browser.
- Service Provider – the entity providing a service to the user via a web application or portal.
- Identity Provider – the entity authenticating the end user and asserting identity information to the service provider.

3.2.1.2 Pre-Conditions

The use case is only concerned with the actual hand-over of the user from the identity provider to the service provider and therefore assumes the following pre-conditions:

- The user has selected service / service provider and identity provider.
- The identity provider is able to authenticate the user to the required assurance level as defined in [ITTArc].
- The authentication of the user by the identity provider has taken place. It is outside the scope for this profile to define how authentication is performed – it is left open for participants to decide. What matters is that the authentication method is classified according to an assurance level as described in [ITTArc] and that this level is specified in the SAML assertion (will be described in section 3.4.8).

The pre-conditions are to be satisfied by the following the scenarios in [ITTArc]. The precise steps in these scenarios should be further detailed (e.g. cookies and HTTP parameters) in a separate interface specification. The architecture implies that SAML <AuthnRequest> messages are not needed since the hand-over to the identity provider is performed by the authentication portal by other means (HTTP redirect with data identifying the service).

3.2.1.3 Steps

The actual steps in the use case are:

- The identity provider redirects the user to the service provider (step 5 in the diagram). The HTTP Artifact binding is used to transfer data to the service provider through the user agent.
- The service provider obtains the user's identity information (via a SAML assertion), performs an authorization decision regarding the requested resource and responds with the resource (if authorization allows it).

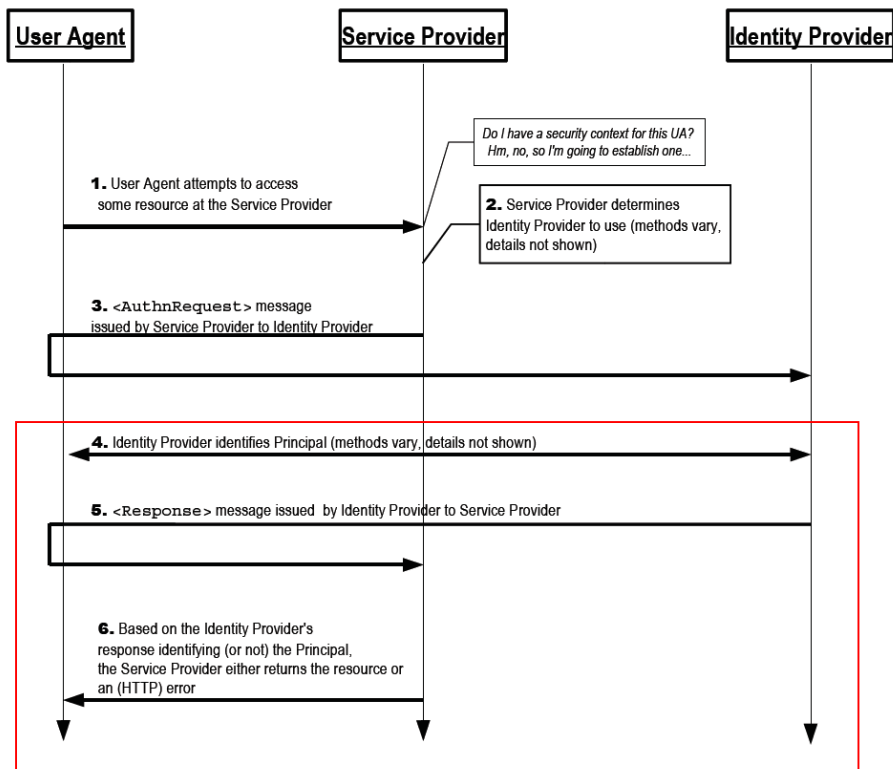


Figure 1: Basic SSO use case (from [SAMLProf])
Note: Only the content in the red rectangle is relevant.

3.2.1.4 Post-Conditions

Upon successful execution of the above steps, the following post-conditions hold:

- The user has received the requested resource without authenticating to the service provider (SSO).
- The service provider has created a security context with the user allowing further resources to be accessed without re-authentication. This property is implementation-specific for the service provider and will not be covered by this profile.

3.3 Profile Description

In this section, each step of the above use case will be described in full details, including specifics of the bindings and processing rules. The actual message exchange is illustrated in the figure below.

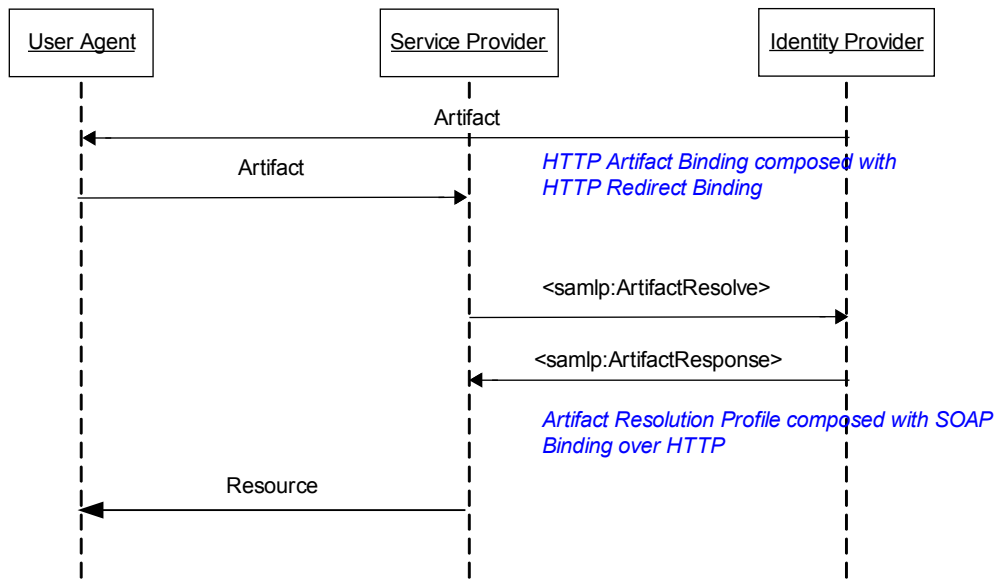


Figure 2: Message Exchange for Artifact Resolution

Requirements for the assertion content are described in section 3.4.

3.3.1 Identity Provider Sends Artifact

Upon successful authentication of the user, the identity provider MUST produce an SAML assertion and artifact and return the artifact to the service provider via the user agent. The HTTP Artifact Binding can either be composed with the HTTP Redirect binding or the HTTP Post binding [SAMLBind] for artifact delivery. In DK-SAML, the HTTP Redirect binding MUST be used.

According to the HTTP Artifact Binding [SAMLBind] the transmission of an artifact to and from the user agent SHOULD be protected with confidentiality. SSL 3.0 or TLS 1.0 MUST therefore be used. However, it is not necessary to require client authentication for the SSL / TLS connection for this step.

The identity provider needs to determine the URL of the service provider’s artifact consumer service to perform the HTTP redirect. According to [ITArch] the service identifier is handed to the identity provider in a cookie from the authentication portal. Since no <AuthnRequest> is sent previously from the service provider this endpoint information cannot be passed in-bound and must therefore be established as meta data. The identity provider may only respond to service providers for which business agreements have been made and whose meta data (service endpoints) are available.

The artifact MUST conform to the `urn:oasis:names:tc:SAML:2.0:artifact-04` artifact type defined in [SAMLBind]. Here, the `TypeCode` is specified to 04 and the source ID and message handles are defined as 20-byte sequences. The source id is created using a SHA-1 hash over the identification URL and the message handle is a random value (either cryptographically strong random or pseudorandom).

3.3.2 Service Provider Resolves Artifact

Upon receiving the artifact, the service provider must map the `SourceID` and `EndpointIndex` from the artifact to the location (URL) of the SAML responder of the identity provider. This mapping must be

facilitated via meta data agreed between identity provider and service provider. Meta data exchange is not defined as part of this profile but left to bilateral agreements between service provider and identity provider (see section 4.2).

The service provider MUST use the artifact resolution profile to retrieve the <Assertion> from the identity provider. The SAML resolution profile allows multiple bindings but the DK-SAML profile requires SOAP binding over HTTP.

It is REQUIRED that all HTTP requests are made over SSL 3.0 or TLS 1.0 in order to maintain confidentiality and message integrity. Furthermore, for the artifact resolution it is REQUIRED that SSL/TLS is used with client authentication to create a mutually-authenticated tunnel. The identity provider must check that the identity of the requesting service provider matches the identity for which the assertion was issued. See section 4.4 for further discussions of security aspects.

3.3.2.1 Requirements for the <Response> message

If the identity provider wishes to return an error, it MUST NOT include any assertions in the <Response> message. Otherwise, if the request is successful, the <Response> element MUST conform to the following:

- The <Issuer> element MAY be omitted, but if present it MUST contain the unique identifier of the issuing identity provider; the `Format` attribute MUST be omitted or have a value of `urn:oasis:names:tc:SAML:2.0:nameid-format:entity`.
- The response MUST contain exactly one <Assertion>. Each assertion's <Issuer> element MUST contain the unique identifier of the issuing identity provider; the `Format` attribute MUST be omitted or have a value of `urn:oasis:names:tc:SAML:2.0:nameid-format:entity`.
- All assertions included in the response message MUST adhere to the requirements described in section 3.4.

3.3.2.2 Processing of the <Response> message

- The message must be processed according to the rules defined in section 4.1.4.3 and 4.1.4.4 of [SAMLProf].

3.4 SAML Assertion Contents

This section will specify requirements for the assertions used in DK-SAML. According to [SAMLCore] the general structure of an SAML 2.0 assertion is:



Figure 3: Structure of a SAML 2.0 Assertion

The following sections describe each of the main elements of the assertion. Since SAML 2.0 provides a great degree of flexibility the main goal of DK-SAML will be to tailor the format to the requirements of the profile. This will facilitate consistency and interoperability – and assure that identity attributes needed in the Danish public sector are properly specified. Note that the `<AuthnDecisionStatement>` in the above figure is not allowed in the DK-SAML profile. This element is on the way to become deprecated in SAML and is addressed in the XACML standard instead.

3.4.1 Main Assertion Element

The assertion must contain exactly one `<AuthnStatement>` and one `<AttributeStatement>` element. All other statements are disallowed since they are outside the scope of the profile.

There are no other profile-specific requirements for the outer element.

3.4.2 Issuer Element

The Issuer element is mandatory and MUST contain a string with the (unique) issuer id. In this profile, the issuer id will be the Uniform Resource Locator of the issuer domain. See section 4.1 for a further discussion of identifiers in the profile.

The element is of type `NameIDType` which defines four other attributes (`NameQualifier`, `SPNameQualifier`, `Format` and `SPProvidedID`). The qualifiers are not needed since the identifiers in

this profile are unique per construction. Further, there is no need to indicate special processing rules via a format attribute and affiliation of issuers are not needed here.

Therefore, none of these four attributes are allowed in DK-SAML.

3.4.3 Signature Element

This element can be used to hold a digital signature over the assertion which provides integrity protection and message authentication of the issuer. It is not a strict necessity to sign assertions in DK-SAML since the desired security properties are achieved by other means (e.g. two-way SSL). See section 4.4.2 for further discussions of security aspects.

The element is therefore optional in DK-SAML and a service provider not capable of processing XML signatures and validating certificate paths can safely ignore it.

3.4.4 Subject Element

An assertion **MUST** contain a `<Subject>` element holding the subject id. The profile is left open regarding the format of subject identifiers – it must be agreed between identity provider and service provider. Encrypted identifiers are however disallowed (see section 4.4.2 for a discussion) in order to avoid processing overhead and key management at service providers.

The element must contain at least one `<SubjectConfirmation>` element containing a Method of `urn:oasis:names:tc:SAML:2.0:cm:bearer`.

The bearer `<SubjectConfirmation>` element described above **MUST** contain a `<SubjectConfirmationData>` element that contains a `Recipient` attribute containing the service provider's assertion consumer service URL and a `NotOnOrAfter` attribute that limits the window during which the assertion can be delivered. It **MUST NOT** contain a `NotBefore` attribute.

3.4.5 Conditions Element

The assertion **MUST** contain an `<AudienceRestriction>` including the service provider's unique identifier as an `<Audience>`.

Other conditions (and other `<Audience>` elements) **MAY** be included at the discretion of the identity provider. (Of course, all such conditions **MUST** be understood by and accepted by the service provider in order for the assertion to be considered valid.)

3.4.6 Advice Element

There are no profile-specific requirements for this element; it can safely be ignored by service providers.

3.4.7 AuthnStatement Element

The assertion **MUST** contain an element describing authentication of the subject to the identity provider.

If the identity provider supports the Single Logout profile, any such authentication statements **MUST** include a `SessionIndex` attribute to enable per-session logout requests by the service provider.

When authenticating subjects using an OCES certificate, the `<AuthnContext>` element **SHOULD** refer to the following authentication context class in an `<AuthContextClassRef>` element:

`urn:oasis:names:tc:SAML:2.0:ac:classes:X509`.

Note that the `<AssuranceLevel>` attribute defined in DK-SAML and used in `<AttributeStatements>` will also provide information about the authentication context. Specifically, it will contain a classification of the authentication strength according to the scheme defined in [ITTAAuthLevel].

Furthermore, a `<SubjectLocality>` element MAY be included to specify the DNS domain and IP address for the system from which the subject was apparently authenticated.

3.4.8 Attribute Statement Element

This element MUST be present in the Assertion and SHOULD contain the *kernel* attributes defined in [ITTAAttr]. The purpose of the kernel attributes is to ensure that different organizations use a common set of attributes to match different accounts for the same user and to provide a consistent naming of attributes. This will simplify integration of user directories and exchange of user attributes between disparate IT systems.

The kernel attributes are selected from the LDAP-schema for "inetOrgPerson" and are:

- sn (surname)
- cn (common name)
- uid (userid)
- mail (the user's email address)

These attributes MUST in DK-SAML be encoded according to the X.500/LDAP Attribute Profile defined in [SAMLProf]. This profile defines a convention for the naming and representation of X.500/LDAP attributes when expressed as SAML attributes. Here, the `Name` attribute will refer the OBJECT IDENTIFIER assigned to the directory attribute type (see e.g. RFC2256 for an overview of attribute types). This will ensure unique SAML attribute names. *Note that all strings encoded this way must be UTF-8 characters strings to ensure a unique encoding.*

For example, a surname attribute containing the value "Petersen" will be encoded as:

```
<saml:Attribute
  xmlns:x500="urn:oasis:names:tc:SAML:2.0:profiles:attribute:X500"
  NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri"
  Name="urn:oid:2.5.4.4" FriendlyName="surName">
  <saml:AttributeValue xsi:type="xs:string"
    x500:Encoding="LDAP">Petersen</saml:AttributeValue>
</saml:Attribute>
```

Implementations SHOULD NOT rely on the `FriendlyName` attribute but instead on the OID included in the `Name` attribute. Should sub-profiles of DK-SAML wish to use other attributes it is recommended to use the same attribute representation when possible. Encrypted attributes are not permitted (see section 4.4.2).

In addition to the kernel attributes DK-SAML requires an `AssuranceLevel` attribute giving the service provider an indication of how strong the user was authenticated. The attribute can have the values "1", "2", "3", "4" and "test" and the semantics of the levels is defined in [ITTAAuthLevel]. The attribute should be encoded with a `NameFormat` set to the namespace of `http://itst.dk/federated/attribute`

Below is given an example of the attribute representation:

```
<saml:Attribute
  NameFormat="http://itst.dk/federated/attribute"
  Name="AssuranceLevel">
  <saml:AttributeValue>2</saml:AttributeValue>
</saml:Attribute>
```

3.4.8.1 Optional Attributes

In [ITTAtrib] two *optional* attributes are further defined:

- `uniqueAccountKey` (a unique key used to match user accounts across systems)
- `cvrNumber` (a reference to the company where the user is employed)

Further, it is envisioned that the Danish social security number (CPR number) will be relevant in many scenarios:

- `cprNumber` (the user's social security number)

When representing these optional attributes, the `Name` attribute SHOULD be one of the values described above (e.g. "`cvrNumber`") and the `NameFormat` attribute MUST be set to `http://itst.dk/federated/attribute`

The `uniqueAccountKey` attribute SHOULD be used according to the recommendations described in [ITTUID] – which builds on XRI from OASIS (Extensible Resource Identifier). It should be encoded as a normal string (`xsi:type="xs:String"`) following the syntax defined here.

The `cvrNumber` attribute SHOULD be encoded as a `positiveInteger` containing exactly 8 digits. A schema definition for this type is:

```
<xsd:simpleType name="cvrNumberType">
  <xsd:restriction base="xsd:positiveInteger">
    <xsd:pattern value="\d{8}"/>
  </xsd:restriction>
</xsd:simpleType>
```

The `cprNumber` attribute SHOULD be encoded as a `positiveInteger` containing exactly 10 digits. Please note that special care must be taken regarding Danish legislation (e.g. `persondataloven`) as well as International legislation (e.g. EU Directive 95/46/EF) when this attribute is exchanged. The user must give his consent for collection and exchange of his CPR number to *each* individual service provider and strong encryption must be used to secure the data in transit.

3.4.8.2 OCES Certificate Attributes

It is anticipated that the Danish OCES PKI (see [OCESPers] and [OCESMedarb]) will frequently be used to authenticate end-users (either by letting the end-user perform an SSL handshake with the identity provider or by letting the end user digitally sign a challenge using e.g. a logon applet in his browser). This section describes attributes from the OCES certificates that may be included in a DK-SAML assertion. *All attributes are optional – but if they are included, they must adhere to the syntax defined in this section.* The attributes must only be used if the user authenticated to the service provider with the certificate and subsequent trust validation to the OCES root and revocation check of the certificate has been performed with the CA.

For OCES person certificates, the most interesting attribute is the PID number which contains a unique identifier for the person. The advantage of PID numbers over CPR numbers is that they can be freely exchanged without risk of violating personal data protection acts. The service provider receiving a PID

number can subsequently ask the user for his CPR number and validate the PID-CPR correspondence by contacting the CA. Alternatively, if the service provider is a government institution with authority to look up CPR numbers it can be done directly without user interaction or consent. With this scheme, the identity provider is thus able to transfer the CPR number *indirectly*. The CPR number is generally a very useful attribute since many systems use it as identifier or primary key.

In OCES person certificates, the PID number is included in a subject serial number. If included in SAML assertions, the PID number should be encoded according to the following syntax:

```
<saml:Attribute
  xmlns:x500="urn:oasis:names:tc:SAML:2.0:profiles:attribute:X500"
  NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri"
  Name="urn:oid:2.5.4.5" FriendlyName="serialNumber">
  <saml:AttributeValue xsi:type="xs:string"
    x500:Encoding="LDAP">PID:9802-2002-2-142339142439
  </saml:AttributeValue>
</saml:Attribute>
```

For OCES employee certificates, the serialNumber attribute contains the CVR of the user's employer plus a unique serial number identifying the employee within the company. If included in the assertion, it should be encoded according to the following syntax:

```
<saml:Attribute
  xmlns:x500="urn:oasis:names:tc:SAML:2.0:profiles:attribute:X500"
  NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri"
  Name="urn:oid:2.5.4.5" FriendlyName="serialNumber">
  <saml:AttributeValue xsi:type="xs:string"
    x500:Encoding="LDAP">CVR:12345678-RID:23423424
  </saml:AttributeValue>
</saml:Attribute>
```

An implementation can distinguish between the two types of serial numbers by looking at the prefix of the attribute value: "PID" identifies OCES person certificates and "CVR" identifies OCES employee certificates.

Other attributes from the subject DN (e.g. organizationalUnitName) can be included in a similar way.

Finally, in cases where the identity provider for some reason wants to deliver the user's entire OCES certificate to the service provider, the below SAML attributes can be used. Here the attribute value must be a base64 encoded string representing the DER encoded X.509 certificate:

```
<saml:Attribute
  xmlns:x500="urn:oasis:names:tc:SAML:2.0:profiles:attribute:X500"
  NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri"
  Name="urn:oid:1.3.6.1.4.1.1466.115.121.1.8" FriendlyName="userCertificate">
  <saml:AttributeValue xsi:type="xs:string"
    x500:Encoding="LDAP">MIIB5DCCAU0CBAJQodoZIhvcNAQ...
  </saml:AttributeValue>
</saml:Attribute>
```

Delivering entire certificates to a service provider in an assertion will however result in additional processing overhead and message footprint.

3.4.9 An Assertion Example

Below is given an example of an assertion conforming to the DK-SAML profile:

```
<saml:Assertion ID="idvalue31231231231312"
    IssueInstant="2001-12-31T12:00:00"
    Version="2.0"
    xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

  <saml:Issuer>http://SomeIdentityProvider.dk</saml:Issuer>

  <saml:Subject>
    <saml:NameID>Hans Jensen</saml:NameID>
    <saml:SubjectConfirmation Method="urn:oasis:names:tc:SAML:2.0:cm:bearer">
      <saml:SubjectConfirmationData
        Recipient="http://SomeServiceProvider.dk"
        NotOnOrAfter="2001-12-31T12:00:00">
      </saml:SubjectConfirmationData>
    </saml:SubjectConfirmation>
  </saml:Subject>

  <saml:Conditions>
    <saml:AudienceRestriction>
      <saml:Audience>http://SomeServiceProvider.dk</saml:Audience>
    </saml:AudienceRestriction>
  </saml:Conditions>

  <saml:AuthnStatement AuthnInstant="2005-01-31T12:00:00Z"
    SessionIndex="29393948329">
    <saml:AuthnContext>
      <saml:AuthnContextClassRef>
        urn:oasis:names:tc:SAML:2.0:ac:classes:X509
      </saml:AuthnContextClassRef>
    </saml:AuthnContext>
  </saml:AuthnStatement>

  <saml:AttributeStatement>

    <saml:Attribute
      xmlns:x500="urn:oasis:names:tc:SAML:2.0:profiles:attribute:X500"
      NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri"
      Name="urn:oid:2.5.4.4"
      FriendlyName="surName">
      <saml:AttributeValue xsi:type="xs:string"
        x500:Encoding="LDAP">
        Jensen
      </saml:AttributeValue>
    </saml:Attribute>

    <saml:Attribute
      xmlns:x500="urn:oasis:names:tc:SAML:2.0:profiles:attribute:X500"
      NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri"
      Name="urn:oid:2.5.4.3"
      FriendlyName="CommonName">
      <saml:AttributeValue xsi:type="xs:string"
        x500:Encoding="LDAP">
        CN=Hans Jensen
      </saml:AttributeValue>
    </saml:Attribute>

    <saml:Attribute
      xmlns:x500="urn:oasis:names:tc:SAML:2.0:profiles:attribute:X500"
      NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri"
      Name="urn:oid:0.9.2342.19200300.100.1.1"
      FriendlyName="uid">
      <saml:AttributeValue xsi:type="xs:string"
        x500:Encoding="LDAP">
        HansJ
      </saml:AttributeValue>
    </saml:Attribute>

    <saml:Attribute
```

```
xmlns:x500="urn:oasis:names:tc:SAML:2.0:profiles:attribute:X500"
NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri"
Name="urn:oid:0.9.2342.19200300.100.1.3"
FriendlyName="mail">
<saml:AttributeValue xsi:type="xs:string"
                    x500:Encoding="LDAP">
                    HansJ@email.dk
</saml:AttributeValue>
</saml:Attribute>

<saml:Attribute
  NameFormat=" http://itst.dk/federated/attribute "
  Name="cprNumber">
  <saml:AttributeValue>2702672834</saml:AttributeValue>
</saml:Attribute>

<saml:Attribute
  NameFormat="http://itst.dk/federated/attribute"
  Name="AssuranceLevel">
  <saml:AttributeValue>2</saml:AttributeValue>
</saml:Attribute>

<saml:Attribute
  xmlns:x500="urn:oasis:names:tc:SAML:2.0:profiles:attribute:X500"
  NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri"
  Name="urn:oid:2.5.4.5" FriendlyName="serialNumber">
  <saml:AttributeValue xsi:type="xs:string" x500:Encoding="LDAP">
  PID:9802-2002-2-142339142439
  </saml:AttributeValue>
</saml:Attribute>

</saml:AttributeStatement>
</saml:Assertion>
```

4.Profile Discussion

This chapter contains analysis and rationale behind the choices made in the DK-SAML profile. The main goal has been to limit the flexibility of the profile, reduce the burden on service providers (e.g. regarding certificate processing) and achieve a high degree of interoperability while at the same time providing the mechanisms needed to solve the SSO problem at hand.

A further goal has been to minimize the need for central coordination and governance regarding (unique) identifiers, certificates, trust, agreements etc.

4.1 Identifiers for Service- and Identity Providers

According to [ITArch] identifiers for the selected authentication service (ASid) and (business) service (Sid) are exchanged between the authentication portal, the authentication service and the (business) service.

In E-Authentication [EgovIntf] these identifiers are centrally managed by the E-Authentication Program Management Office. The identifiers are here positive integers with associated meta data describing e.g. service endpoints. To avoid central management it has for this profile been decided to use URLs extended with a number as identifiers. Per construction these identifiers will be unique and there will be no need for central management to ensure uniqueness. However, these identifiers will be a bit longer than the identifiers used in E-Authentication.

Example:

If a service provider owns the domain `www.xyzservices.dk` he must enumerate his services and assign identifiers being `http://www.xyzservices.dk/SID/<servicenumber>`, where the `<servicenumber>` will be 1, 2, 3, 4... etc.

Similarly, an authentication service provider who owns the domain `www.acmelogin.dk` must enumerate the authentication services and assign identifiers being `http://www.acmelogin.dk/AID/<servicenumber>`

Each service provider / authentication service provider will be responsible for unique enumeration of services within their own domain.

These identifiers are exchanged using HTTP redirects through the user agent with identifiers passed in query string parameters. To avoid encoding problems these identifiers must be URL encoded during transport.

Furthermore, SAML requires that identity providers have unique identifiers. These identifiers are used both in assertions and protocol messages. In the DK-SAML profile we will use the domain name as unique identifier (e.g. `http://www.acmelogin.dk`). This identifier will coincide with the identity established via (server) SSL / TLS certificates.

4.2 Meta Data

Participants in the federation architecture need information about other parties in order to fulfill their job. In this section we will describe which information is required. *However, it is outside the scope to define how meta data is represented or exchanged.* This will be left to parties to decide through bilateral agreements. Other federation initiatives (like E-Authentication and Liberty Alliance) have a more formal approach to meta data representation, -exchange and -governance.

4.2.1 Meta Data about services and service providers

For each service there must exist meta data describing the required assurance level and the target URL where the service can be accessed:

ServiceID	Required Assurance Level	Service Description	Target URL
http://sp1.dk/SID/1	1	...	http://sp1.dk/application1
http://sp1.dk/SID/2	3	...	http://sp1.dk/application2
http://sp2.dk/SID/1	2	...	http://sp2.dk/app1

For each service provider it is further necessary to record the URL of the assertion consumer service plus the client certificate it will use for authentication to the authentication service:

Service Provider ID	Assertion Consumer URL	Client Certificate
http://sp1.dk	https://sp1.dk/saml-artifact/receiver
http://sp2.dk	https://sp2.dk/saml-artifact/receiver

In the interest of simplicity, it is assumed that each service provider has only one assertion consumer URL which is shared for all applications. When the SAML re-direct from a service provider to identity provider is performed, it will necessary to specify which application (ServiceID) that was actually selected by the user. In SAML 1.1 under the Browser/Artifact profile the HTTP request delivering the artifact (section 4.1.1.5, Step 3) includes a TARGET parameter which specifies the target application. Even though the TARGET parameter is not part of SAML 2.0, this profile will use a TARGET parameter to transfer the service ID such that the service provider can return the right resource after the artifact and assertion have been obtained and validated.

4.2.2 Meta Data about authentication services and providers

For each authentication service we need to record the assurance level it offers and the target URL of the authentication service:

Authentication Service ID	Offered Assurance Level	Target URL
http://authservice.dk/AID/1	3	https://authservice.dk/logontype1
http://authservice.dk/AID/2	1	https://authservice.dk/logontype2
http://anotherservice.dk/AID/1	2	https://authservice.dk/authentication

For each identity provider it is further necessary to record the URL of the SAML responder URL (where the artifact can be resolved to an assertion):

Identity Provider ID	SAML Responder URL
http://authservice.dk	https://authservice.dk/saml/responder
http://anotherservice.dk	https://anotherservice.dk/saml/responder

In the artifact resolution protocol, the authentication service (=identity provider) must use an SSL / TLS server certificate for authentication. This certificate should have the subject common name equal to the domain name - which again will be equal to the identity provider identifier used in SAML assertions and protocol messages (see section 4.1).

4.2.3 Other meta data

Each authentication service must maintain data about the user's preferences. One important element here is whether the user has opted in or not to Single Sign-On. Other preferences could determine which data the user allows to be exchanged with service providers. Every authentication service must publish its privacy policy and make it available to its users.

4.3 Protocol and Binding Selection

4.3.1 Artifact Binding

The DK-SAML profile uses the HTTP Artifact Binding defined in [SAMLBind]. This binding is composable with the HTTP Redirect binding and the HTTP POST binding. In the interest of simplicity this profile only allows use of the HTTP Redirect binding for artifact delivery. This eliminates the need for signed assertions - see section 4.1.4.5 in [SAMLProf].

4.4 Security Considerations

4.4.1 Transport Level Security

DK-SAML leverages security mechanisms from the HTTP(s) transport bindings in order to ensure authentication, confidentiality and integrity of in-transit artifacts and assertions.

More specifically, the following requirements exist for transport level security:

- The HTTP connection used for artifact delivery must be secured with SSL 3.0 / TLS 1.0. The connection is not required to use client authentication (just server authenticated SSL / TLS). This will ensure confidentiality and integrity of the artifact during transport.
- The SOAP over HTTP connection used for artifact resolution must be secured with two-way SSL 3.0 / TLS 1.0. This will protect the assertion during transport and ensure that only the intended service provider can receive the assertion. The identity provider must ensure that the client certificate matches the entity for which the assertion was produced (the Sid). Furthermore, only SSL / TLS cipher suites providing strong encryption are allowed since sensitive personal data such as CPR numbers can be exchanged.

Please note that the security of the entire SSO architecture will not be better than the security of the authentication mechanism used by the identity provider to authenticate the end-user (which is outside the scope of SAML). However, use of the `AssuranceLevel` attribute means that compromise of weak authentication methods or credentials (e.g. a user loses a static password) will only have limited effect.

The use of SSL / TLS requires that trust mechanisms are established between the parties. Typically, this is done by requiring each entity to maintain a store of trusted peer certificates and/or trusted CA certificates. Secure connections MUST only be allowed from parties who own a private key whose public key can be validated with this store i.e. a certificate path to a trusted certificate can be established.

It is outside the scope of this profile to specify how these trust mechanisms are set up.

4.4.2 Signing and Encryption of SAML elements

Security mechanisms are built into SAML assertions themselves – which are thus independent of transport / binding security mechanisms. Among these are XML encryption and XML digital signing.

Digital signing of an entire assertion is possible via the `<ds:Signature>` element. The advantage of signing an assertion is that it will be integrity-protected end-to-end (beyond the point where the SSL session is terminated – typically in a DMZ) and that the protection will out-live any SSL sessions. Signing assertions will also allow service providers to store them as evidence – should an identity provider later repudiate having issued an assertion. However, verification of digital signatures could potentially put unnecessary burdens on service providers since the transport level security is sufficient for the HTTP Artifact binding used in DK-SAML. The use of signatures is therefore *optional* and a service provider is allowed to ignore a signature element when processing an assertion.

Encryption is available for both whole artifacts (`<saml:EncryptedAssertion>`), attributes (`<saml:EncryptedAttribute>`), and identifiers (e.g. `<saml:EncryptedID>`). Since these more advanced security mechanisms are really not needed in this profile, they are disallowed for the sake of simplicity. Encrypted content cannot be ignored by service providers and supporting it would require complicated key management. It is further believed that most available SAML implementations on market today have good support for transport security whereas availability of content encryption is far more limited.

4.4.3 Other Security Mechanisms

There exist a number of additional security mechanisms used by the profile. These are intended to ensure that assertions are not misused (e.g. against a wrong service provider):

- The `<SubjectConfirmationData>` element of the assertion contains a `Recipient` attribute referring the service provider. This ensures that an assertion can only be used at the service provider for which it was intended.
 - It further contains a `NotOnOrAfter` attribute (which is mandatory) that limits the window during which the assertion can be delivered. Thus a stolen assertion could only be used within a small time window (less than 15 minutes).
- The `<AuthnStatement>` element MAY include a `<SubjectLocality>` element to specify the DNS domain and IP address for the system from which the subject was apparently authenticated. This will prevent stolen session cookies or artifacts to be used by an attacker.
- The `<Conditions>` element MUST contain an `<AudienceRestriction>` referring to the service provider's id. Again this prevents use of the assertion at a wrong service provider.

Note that a service provider must enforce a one-time semantics for assertions to ensure that an assertion cannot be re-played. Use of two-way SSL / TLS for artifact resolution ensures that no man-in-the-middle will be able to obtain the assertion.

4.4.4 Securing Session Cookies

With the security mechanisms described above, the most vulnerable point in the SSO architecture is probably the session cookie established by the identity provider. Should an attacker be able to steal this cookie he will be able to SSO to services at or below the given assurance level until the session times out. The session cookie mechanism is not part of the SAML profile but lies on the boundary since it is used to perform SSO and obtain new artifacts and assertions.

The cookie can be compromised if the internal browser state is exposed (e.g. if cookies are stored in persistent storage in an un-secure fashion). The architecture therefore relies on the browser to protect the session cookie established by the identity provider.

There are additional steps which can be taken to mitigate such attacks:

- An identity provider SHOULD check that all SSO requests bound to a particular session cookie originate from the same client IP address. This will prevent an attacker from using a stolen cookie at another system.

- Use of the <SubjectLocality> attribute has a similar effect but the check occurs at the service provider side. It MUST be checked by the service provider if present.
- A service provider can force a fresh re-authentication before access is granted to critical services. This is done by sending a special message to the identity provider. However, this hurts the user-friendliness of the system.

Stealing an artifact is less critical since these are generally fixed towards a given service provider and typically has very short lifetime (typically shorter than an SSO session). Hence, an attacker has limited possibilities for misusing the artifact.

4.5 Error Handling

The architecture is intended to provide encapsulation of multiple federation schemes (e.g. SAML, Liberty Alliance, WS-Federation) such that the architecture will remain the same even if the underlying federation technologies evolve or change. Even though the Danish e-authentication initiative currently only supports SAML the encapsulation property is still desirable in order to avoid tight coupling with SAML.

To encapsulate SAML there should a mapping between SAML-specific error-handling to an architecture-wide error handling scheme. This is done by having the authentication portal implement an error handling service (URL) supporting a number of standardized error codes. It will further ensure a consistent user experience and central logging of problems.

Should an identity- or service provider encounter a problem during processing or creation of SAML messages, they should convert the SAML error to a generic error code and re-direct the user to the authentication portal with this error code.

One example is the SAML responder at the identity provider. If it refuses a request the SAML HTTP Artifact Binding specifies that a <samlp:StatusCode> with a certain value is returned. Upon receiving this error message, the service provider should re-direct the user to the authentication portal's error handling URL with a proper error code and message identifying the problem.

In addition, each entity is recommended to keep their own trace log since not all details of the error and its context will be transferred to the portal.

4.6 Governance and Management of the Profile

The profile is intended to require a minimal amount of central management and governance by IT- og Telestyrelsen. It is largely left to identity providers and service providers to establish bilateral agreements defining both business issues and technical issues (e.g. identifiers, certificates, end-point meta data).

The table below describes a few management / governance areas and how they are to be handled:

Area	Comment
Profile Versioning	The versioning and content of the base DK-SAML profile is maintained solely by IT- og Telestyrelsen in Denmark.
Identifiers and certificates	Participants can freely choose unique identifiers according to the syntax and rules defined in this profile. Per construction there will be no need to centrally manage these identifiers to ensure uniqueness.
New attributes and sub-profiles	Identity providers can freely add identity attributes to the profile and even establish sub-profiles containing specific sets of attributes (e.g. for the healthcare sector). Special attention must be paid to Danish and International legislation (e.g. "Persondataloven").
Compliance to profile	There will (so far) be no central authority to evaluate whether a given implementation is compliant with this profile. Prototype implementations are expected to highlight problems areas regarding interoperability which can result in a

	further clarification of the profile.
Trust	Trust will be handled via business agreements between the participants and further via technical agreements defining which certificates to trust. It is recommended to use OCES company certificates for this purpose.
Meta Data	IT- og Telestyrelsen will <i>not</i> maintain a central repository with meta data (e.g. service end points) and will not specify mechanisms for automated meta data exchange. It must be handled via agreements between the involved parties.

Note that the American E-Authentication initiative has a more strict governance model where identifiers are centrally approved and certificates for trust establishment are also issued by the governing authority.

4.7 Privacy

The privacy issues take its starting point in any user's right to be ensured that private or personal data is treated in accordance with Danish and International Privacy Legislation.

Private or personal data is e.g. data about racial or ethnical background, political, religious or philosophical beliefs, union membership, or data about health or sexual affairs. A CPR-number is as such considered private information and must be treated with special care.

This will not be a thorough discussion of all the necessary precautions but the most important will be noted in this chapter. They fall in 3 sections:

- Registration of private or personal data
- Information and choices that must be given to the user
- Transfer of private or personal data such as attributes about the user between Identity Provider and Service Provider.

The Service Provider should always follow the procedures necessary due to the nature of the service provided.

4.7.1 Registration and use of private or personal data

Generally it is not allowed to register any private or personal data about a physical person unless there is a law that says it is allowed or the party making the registration has a genuine need to register the data in order to carry out otherwise legal business.

However, if the person whose data is to be registered gives his or hers explicit consent to the registration and use of the data, the data can be registered, used and transferred to third parties only for the consented purpose and with sufficient security precautions. Security precautions are stated in **[Sikkerhedsbekendtgørelsen] (Bkg. nr. 528 af 15. juni 2000 som ændret ved bkg. nr. 201 af 22. marts 2001)**

The Identity Provider therefore has to get the users consent to the registration and use of any private or personal data involved in the authorisation of the user. If the Service Providers register and use private or personal data about the user, each Service Provider has to explicitly get the user's consent. If the Identity Provider or Service Provider is a Public Authority special restrictions may apply.

Registration and use of private or personal data must be reported to Datatilsynet prior to the first release of the service.

Publicly available data such as common name and PID are not considered private or personal.

4.7.2 Information and choices that must be given to the user

The user will by default opt-in to Single Sign-on, but each Identity Provider must provide a mechanism by which the user can opt-out. If the user has opted-out the Identity Provider must remember this and give the user the possibility to opt-in at a later point in time.

The Identity Provider must inform the user of which data is collected about the user and the purpose of collecting the data.

If private or personal data about the user is about to be transferred to any third party, the Identity Provider must inform the user hereof and give the user the possibility to abort the action.

The Identity Provider can choose to inform the user of which data regarding the user is transferred to which Service Providers in a general consent-form. If no user-consent is given the data must not be transferred to a third party.

The user is to be informed of the consequences of the choices made regarding transfer of private and personal data, e.g. that the user will not be able to be verified at a sufficient security level to access the Service chosen.

4.7.3 Transfer of private or personal data between Identity Provider and Service Provider.

The Identity provider has the responsibility to ensure that attributes regarding a user can lawfully be transferred to the Service Providers.

If the data is publicly available – like the kernel attributes – the attributes can be transferred, when the user is informed of the transferral either by the initial agreement with the Identity Provider or in a general text at the Identity Providers site.

If however the data is personal or private – like which occupation the user has or the user's CPR-number – consent to the registration and transfer of the attribute is required from the user.

The Service Provider must be able to assume that the consent is given, if the Identity Provider forwards the attribute to the Service Provider.

Furthermore, special care must be taken regarding security when these types of attributes are exchanged. Not only must the user give his consent for collection and exchange of his private attributes to each individual service provider, strong encryption must also be used to secure the data in transit. [Sikkerhedsbekendtgørelsen] gives instructions for other security measures to be applied.

4.8 Single Logout

SAML 2.0 supports the concept of single logout and describes both a Single Logout Protocol in [SAMLCore] and a Single Logout Profile in [SAMLProf]. These allow identity- and service providers to terminate multiple user sessions by exchanging <LogoutRequest> and <LogonResponse> messages. In this way, a user can perform near-simultaneous logout to all service providers whose session originate from a particular identity provider (i.e. "single logout"). The user may either contact a service provider or an identity provider to initiate the logout.

The possible variations in the Single Logout Profile pertain to which binding that is selected (e.g. SOAP binding, HTTP Redirect, HTTP POST, or Artifact binding). In DK-SAML we strongly recommend the use of the Artifact binding: this is consistent with the other parts of DK-SAML and can use the same transport level security and trust mechanisms already established (i.e. use of client-authenticated SSL for artifact resolution). If other bindings were used, the <LogonRequest> messages had to be digitally signed (and verified!) which would put an extra burden on identity- and service providers.

However, since it is an important design goal to build the SSO architecture independent of any particular federation technology [ITArch], we should ideally perform logout in the layer above SAML. This could in theory be achieved by defining a set of re-directs and HTTP parameters in the interface specification,

similar to the hand-offs described in [EgovIntf]. But since this approach would replicate the features already present in SAML 2.0 just for the sake of abstraction, DK-SAML will for now implement single logout using SAML alone.

It is left open to decide whether support for single logout should be mandatory or optional for service- and identity providers. Studies of current commercial products can later show whether this is a realistic requirement to state or whether it will narrow the set of compliant products too much.

5. Implementation Considerations

This chapter contains considerations for the implementation of the DK-SAML profile at a service provider. An important aspect is here the integration with existing security systems protecting the service provider's applications. Often a service provider has already deployed (web) applications with associated access control systems and will subsequently need to add support for DK-SAML.

The description aims at enabling service providers to identify the required tasks, highlight areas where customization of standard products can be foreseen, identify and match requirements against different vendor's products, and be the basis for planning and estimating of development- and implementation projects.

5.1 Logical Security Architecture

In order to structure the discussion we will assume a *logical* security architecture as depicted below in Figure 4. The architecture is described at a high level of abstraction and is not bound to any specific products or platforms – in fact it can be seen as a security *pattern* which can be realized in many different ways with real products. The components shown are thus generic components that should ideally be present in any access control system. Furthermore, no operational restrictions such as placement of components on physical servers are assumed.

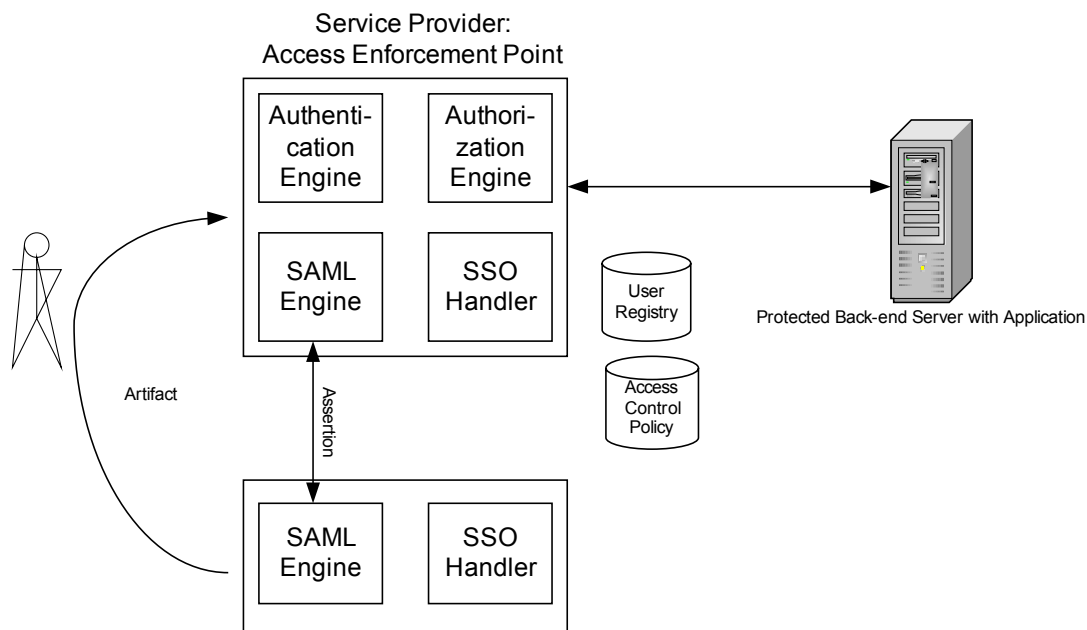


Figure 4: Logical Architecture for Access Control System

The responsibilities of the components (seen from the service provider) are:

- **Access Enforcement Point.** This component mediates all external requests to the protected back-end applications. It will ensure that a user is properly authenticated and authorized before access to an application is granted, and it will maintain the authentication session with the user. The access enforcement point relies on a number of sub-components described below.
- **Authentication Engine.** This component is responsible for authentication of end users and systems and cooperates with the SAML engine for SSO purposes. The authentication engine will

return a local user id along with group and role membership upon successful authentication. These data are put in the user's internal credential and feeds into the authorization engine.

- **SAML Engine.** This component supports creation and processing of SAML messages including artifact delivery and assertion resolution. It can build or populate the user credential based on the information extracted from a SAML assertion. The component further supports a number of SAML bindings (e.g. HTTP, Artifact, SOAP).
- **SSO Handler.** This component handles the HTTP redirects and cookies which are part of the overall SSO architecture described in [ITArch] and more detailed in [EgovInf]. For example, if a user attempts to access the application directly without being authenticated at an identity provider, this handler must redirect the user to the authentication portal with the application ID <AAid> set on the query string. The SSO handlers are generally HTTP handlers which get triggered by redirects containing parameters in query strings and cookies.
- **Authorization Engine.** After successful authentication and establishment of the user's credentials (e.g. group and role membership), the authorization component decides whether the user is actually allowed to access the requested resource. This is done by lookup in the access control policy store.
- **Access Control Policy Store.** This data store contains the access control policy for the service provider. It includes a representation of the back end applications and resources which are being protected and defines who (user IDs, groups, roles) are allowed to access which resources (possibly also specifying the detailed operations allowed).
- **User Registry.** This data store contains the user identities (physical or logical) along with their membership of groups or roles. The access control policy is formulated in terms of local user identities from this store, and all external requests are mapped to a specific, local user ID. The mapping can be coarse-grained (e.g. mapping all citizens authenticated with an OCES person certificate to the same user ID) or fine-grained (e.g. having individual user IDs for each physical person).

5.2SSO Scenario

This section describes the sequence through a normal SSO scenario where the tasks performed by the components in the above architecture are highlighted. The sequence shows how the components interact with each other to perform a specific task. Focus will primarily be on the tasks performed by the service provider.

The sequence is as follows:

1. The user visits the authentication portal and selects an application from a service provider and an authentication service from an identity provider.
2. The user gets re-directed to the identity provider with a parameter specifying the application / service provider.
3. The user authenticates to the identity provider if not already authenticated.
4. The identity provider creates a SAML artifact and assertion and re-directs the user to the service provider (HTTP redirect with artifact in query string). The query string further contains a target parameter specifying the application target.
5. The service provider SAML engine gets invoked and extracts the SAML artifact from the query string.
6. The artifact is processed and the identity provider end-point is looked up in the meta data.
7. The service provider invokes the identity provider to retrieve the assertion (back channel).
8. The assertion is processed by the SAML engine and the service provider extracts identity data and puts them in the user's credential which is returned to the authentication engine. The user is here mapped to a local user ID and an authentication session is created.

9. The authorization engine is invoked and based on the user credential and information in the policy data store an access control decision is made.
10. If authorization is granted, the user sent the requested application resource.

5.3 Service Provider Tasks

Based on the above descriptions, we will now highlight the tasks the service provider needs to perform with focus on DK-SAML. The intention is to identify areas where customization or custom development can be foreseen and to provide an idea of the work required.

5.3.1 SSO Handler Tasks

Each service provider needs to implement an HTTP handler supporting the SSO Architecture. Since the SSO architecture in [ITArch] and [EgovIntf] is a new layer invented to encompass several different underlying SSO schemes (SAML, Liberty Alliance etc.) it must be implemented from scratch.

The SSO handler needs to:

- Accept incoming HTTP redirects.
- Parse the query string and examine the parameters describing application- and service IDs.
- Forward the request to the SAML engine (e.g. when an artifact is delivered).
- Forward the request the authentication portal if the required parameters are not specified or an error occurs.

5.3.2 SAML Engine Tasks

The SAML engine needs to perform the following tasks:

- Receive SAML artifacts via the artifact binding composed with HTTP redirect binding.
- Receive logout requests (if single logout is supported).
- Process artifacts and lookup identity provider URL in meta data (to get the end point where the artifact can be resolved).
- Resolve SAML artifact to assertion via SOAP binding over two-way SSL. Note that trust stores with certificates needs to be setup.
- Process and validate assertions with standard SAML processing rules.
- Extract user attributes from assertion and perform DK-SAML specific processing. Several of the attributes in DK-SAML are local Danish attributes (some defined in IT- & Telestyrelsen's name space). Special handling of these attributes is therefore expected – e.g. plug-ins to the SAML engine or XSLT instructions to extract and validate the attributes.

It is believed that most commercial products containing SAML implementations can perform nearly all of the above tasks except handling of the DK-SAML specific attributes. Some customization must therefore be anticipated but for many products this can probably be accomplished without any programming.

5.3.3 Authentication Engine Tasks

This component needs to perform the following tasks during SSO:

- Receive validated assertion from SAML engine.
- Map assertion attributes to local user id.
- Add user attributes from assertion to user credential.
- Create session with end user.

Thus, most of the authentication process is actually handled by the SAML engine. Customization can be required for mapping attributes in the assertion to a local user ID and if the attributes from the assertion should be made available in some other form for back-end applications. Typically, attributes from the assertion would be made part of the user credential which is then shared with back end systems.

5.3.4 Authorization Engine Tasks

The authorization engine needs to perform the following tasks:

- Perform authorization decisions based on user credential and requested resource. This involves lookup in the authorization policy store and evaluating the policy defined for the resource against the user's group and role memberships.
- Return resource (target application) if authorization was granted.

This is believed to be standard-functionality found in most mature products. However, if authorization decisions needs to be based on custom attributes (e.g. assurance level) a plug-in or customization is probably required.

References

- [SAMLCore]** "Assertion and Protocols for the OASIS Security Assertion Markup Language 2.0", OASIS Standard. <http://docs.oasis-open.org/security/saml/v2.0/saml-core-2.0-os.pdf>
- [SAMLProf]** "Profiles for the OASIS Security Assertion Markup Language (SAML) V2.0", OASIS Standard. <http://docs.oasis-open.org/security/saml/v2.0/saml-profiles-2.0-os.pdf>
- [SAMLBind]** "Profiles for the OASIS Security Assertion Markup Language (SAML) V2.0", OASIS Standard. <http://docs.oasis-open.org/security/saml/v2.0/saml-profiles-2.0-os.pdf>
- [ITTArch]** "Anbefaling om fælles arkitektur for tværgående autenticitetssikring". <http://www.oio.dk/files/Horing.B.st.tvergaendeautenticitetssikring.v3.pdf>
- [ITTAAuthLevel]** "Vejledning vedrørende niveauer af autenticitetssikring". <http://www.oio.dk/files/Horing.B.st.niv.autenticitetssikring.v3.pdf>
- [ITTAattrib]** "Anbefaling til kerneattributter for bruger". <http://www.oio.dk/files/Horing.B.st.kerneattributter.v3.pdf>
- [ITTUID]** "Anbefaling til unik id-nøgle". <http://www.oio.dk/files/Horing.B.st.id-nogle.v3.pdf>
- [EgovTechApp]** "Technical Approach for the Authentication Service Component Version 1.0.0 June 28, 2004". <http://www.cio.gov/eauthentication/documents/TechApproach.pdf>
- [EgovSAMLProf]** "SAML Artifact Profile as an Adopted Scheme for E-Authentication". <http://www.cio.gov/eauthentication/documents/SAMLprofile.pdf>
- [EgovIntf]** "E-Authentication Interface Specifications for the SAML Artifact Profile". <http://www.cio.gov/eauthentication/documents/SAMLspec.pdf>
- [NistEIAuth]** "Electronic Authentication Guideline, NIST Special Publication 800-63 Version 1.0.1". http://www.csrc.nist.gov/publications/nistpubs/800-63/SP800-63v6_3_3.pdf
- [OCESPers]** "Certifikatpolitik for OCES-personcertifikater", Version 3.0, IT- og Telestyrelsen
- [OCESMedarb]** "Certifikatpolitik for OCES-medarbejdercertifikater", Version 4.0, IT- og Telestyrelsen