Samleikin Deployment Profile

Version (Final): 1.0 - 01.03.2020. Effective Date: 28-05-2020

- 1. Introduction
  - 1.1. Requirements Notation
  - 1.2. References to SAML 2.0 Standards and Profiles
- 2. Deviations from Referenced Profiles
  - 2.1. Kantara Initiative eGovernment Implementation Profile of SAML V2.0 (Version 2.0bis)
  - 2.2. SAML V2.0 Implementation Profile for Federation Interoperability
  - 2.3. SAML V2.0 Implementation Profile for Federation Interoperability
- 3. Requirements for Metadata Content
  - 3.1. Generic
  - 3.2. Service Providers
  - 3.3. Identity Providers
- 4. Attributes
- 5. Authentication Requests
  - 5.1. Binding and Security Requirements
  - 5.2. Message Content
  - 5.3. Processing Requirements
    - 5.3.1. Validation of Destination
    - 5.3.2. Validation of Assertion Consumer Addresses
    - 5.3.3. Identity Provider User Interface
    - 5.3.4. Authentication Context and Levels of Assurance Handling
    - 5.3.5. Single Sign On Processing
    - 5.3.6. Single Logout Processing
- 6. Authentication Response
  - 6.1. Security Requirements
  - 6.2. Message Content
    - 6.2.1. Attribute Release Rules
  - 6.3. Processing Requirements
    - 6.3.1. Signature Validation
    - 6.3.2. Subject Confirmation
    - 6.3.3. Conditions
    - 6.3.4. The Authentication Statement
    - 6.3.5. General Security Validation
  - 6.4. Error Responses
- 8. Changes Between Versions

1. Introduction

This is the SAML deployment profile for Samleikin, the Faroese eID solution. It extends the following two implementation profiles.

- Kantara Initiative eGovernment Implementation Profile of SAML V2.0 (Version 2.0bis) [eGov]
- SAML V2.0 Implementation Profile for Federation Interoperability [fedInterOp]

In addition it extends the deployment profile:

- SAML V2.0 Deployment Profile for Federation Interoperability [SAML2int]

All referenced profiles and specifications in the profiles above and the requirements stated therein are also valid for this deployment profile unless specifically stated. Readers should be familiar with all relevant referenced documents, and any requirements stated are not repeated unless where deemed necessary to clarify or highlight a certain issue. Also, any SAML features specified in referenced SAML documents that are optional are also optional in this profile, unless explicitly specified otherwise.

1.1. Requirements Notation

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

The use of SHOULD, SHOULD NOT, and RECOMMENDED reflects broad consensus on deployment practices intended to foster both interoperability and guarantees of security and confidentiality needed to satisfy the requirements of many organizations that engage in the use of federated identity. Deviating may limit a deployment's ability to technically interoperate without additional negotiation, and should be undertaken with caution.

1.2. References to SAML 2.0 Standards and Profiles

When referring to elements from the SAML 2.0 core specification [SAML2Core], the following syntax is used:

- `<saml2p:Protocolelement>` – for elements from the SAML 2.0 Protocol namespace.
- `<saml2:Assertionelement>` – for elements from the SAML 2.0 Assertion namespace.

When referring to elements from the SAML 2.0 metadata specifications, the following syntax is used:

- `<md:Metadataelement>` – for elements defined in [SAML2Meta].
- `<mdui:Element>` – for elements defined in [SAML2MetaUI].
- `<mdattr:Element>` – for elements defined in [SAML2MetaAttr].

When referring to elements from the W3C XML Signature namespace (`http://www.w3.org/2000/09/xmldsig\#`) the following syntax is used:

- `<ds:Signature>`

2. Deviations from Referenced Profiles

2.1. Kantara Initiative eGovernment Implementation Profile of SAML V2.0 (Version 2.0bis)

The table below states the deviations from the Kantara Initiative eGovernment Implementation Profile of SAML V2.0 (Version 2.0bis) [eGov]. Refer to the original document for the requirement text.

Requirment ID	Deviation
eGov-018	The requirement is mandatory.
eGov-019, eGov-063, eGov-070, eGov-071, eGov-072, eGov-073, eGov-074, eGov-081, eGov-082, eGov-083, eGov-084, eGov-085, eGov-092, eGov-093, eGov-101, eGov-102	Requirements to be disregarded.

2.2. SAML V2.0 Implementation Profile for Federation Interoperability

The table below states the deviations from the SAML V2.0 Implementation Profile for Federation Interoperability [fedInterOp]. Refer to the original document for the requirement text.

Requirement ID	Deviation
IIP-IDP11, IIP-IDP13, IIP-IDP14, IIP-IDP15, IIP-IDP16	Requirements to be disregarded.

2.3. SAML V2.0 Implementation Profile for Federation Interoperability

The table below states the deviations from the SAML V2.0 Deployment Profile for Federation Interoperability [SAML2int]. Refer to the original document for the requirement text.

Requirement ID	Deviation
SDP-ALG01	Digest SHALL be: http://www.w3.org/2001/04/xmlenc#sha512 Signature SHALL be: http://www.w3.org/2001/04/xmldsig-more#rsa-sha512 Block encryption SHALL be: http://www.w3.org/2009/xmlenc11#aes384-gcm Key transport digest SHALL be no less than: http://www.w3.org/2000/09/xmldsig#sha256
SDP-SP13, SDP-SP14	Requirements to be disregarded.

3. Requirements for Metadata Content

3.1. Generic

All services that are represented in the Metadata SHALL include a `<md:Organization>` element with mandatory child elements, which includes at least one of each of the elements `<md:OrganizationName>`, `<md:OrganizationDisplayName>` and `<md:OrganizationURL>`.

The `<md:OrganizationName>` element SHALL hold a registered name of the organization, which matches the agreement with the federation operator.

The `<md:OrganizationDisplayName>` element SHALL contain a display name of the organization and SHALL NOT contain a service name that is unrelated to the name of the organization.

All services represented in the metadata SHALL include asymmetric public keys in the form of a certificate, which supports both signature validation and encryption. RSA public keys are accepted and the minimum length of keys SHALL be no less than 3072 bits.

The same public key MAY support both signature validation and encryption, indicated by an absent `"use"` attribute.

3.2. Service Providers

Any needs for particular attributes from Identify Providers, when present, MUST be expressed through <md:AttributeConsumingService> elements assigned a specific index value, with underlying <md:RequestedAttribute> elements in the Service Provider metadata. The <md:RequestedAttribute> elements in the Service Provider metadata, when present, hold a list of requested and/or required attributes. A Service Provider MUST specify what AttributeConsumingService Index is used as part of a SAML request message sent to and IdP.

Metadata for a Service Provider SHALL contain an <mdui:UIInfo> extension, extending the <md:SPSSODescriptor> element. This <mdui:UIInfo> element SHALL at least contain a <mdui:DisplayName> element with the language attribute fo (Faroese), representing the Service Provider name that has been approved by the federation operator. The <mdui:UIInfo> element SHALL also contain a reference to a logotype image (<mdui:Logo>) and SHOULD contain a <mdui:Description> element with the language attribute fo (Faroese).

The above elements represented in Faroese SHALL also be represented with the language attribute en (English).

A Service Provider MAY sign authentication request messages sent to Identity Providers. A Service Provider that signs authentication requests messages MAY also ensure that a receiving Identity Provider will only accept valid signed requests from this Service Provider by assigning the AuthnRequestsSigned attribute of the <md:SPSSODescriptor> to a value of true.

Section E7, “Metadata for Agreeing to Sign Authentication Requests”, of [SAML v2.0 Errata 05] specifies the following concerning the AuthnRequestsSigned attribute:

> *Optional attribute that indicates whether the <saml2p:AuthnRequest> messages sent by this service provider will be signed. If omitted, the value is assumed to be false. A value of false (or omission of this attribute) does not imply that the service provider will never sign its requests or that a signed request should be considered an error. However, an identity provider that receives an unsigned <saml2p:AuthnRequest> message from a service provider whose metadata contains this attribute with a value of true MUST return a SAML error response and MUST NOT fulfill the request.*

Furthermore, a Service Provider MUST require assertions that are issued to it, to be signed. This is done by assigning the WantAssertionsSigned attribute of the <md:SPSSODescriptor> to a value of true.

Note that the response message that carries the assertion will always be signed, so the Service Provider should only require signed assertions in case that it wants to preserve the proof of authenticity of an assertion separate from the response.

3.3. Identity Providers

The <mdattr:EntityAttributes> element of an Identity Provider’s metadata SHALL contain an attribute according to [SAML2IAP] with Name="urn:oasis:names:tc:SAML:attribute:assurance-certification" holding at least one attribute value identifying a Level of Assurance (LoA) level for which the Identity Provider has been approved and where the value is one of the identifiers defined in section 3.1.1 of [EidRegistry] and whose meaning are defined in [eIDAS].

Metadata for an Identity Provider SHALL contain an <mdui:UIInfo> extension, extending the <md:IDPSSODescriptor> element. This <mdui:UIInfo> element SHALL at least contain a<mdui:DisplayName> element with the language attribute fo (Faroese), representing the Identity Provider service name that has been approved by the federation operator. The <mdui:UIInfo> element SHALL also contain a reference to a logotype image (<mdui:Logo>) and SHOULD contain a <mdui:Description>element with the language attribute fo (Faroese).

It is RECOMMENDED that the above elements represented in Faroese also be represented with the language attribute en (English).

An Identity Provider MAY require authentication request messages to be signed. This is indicated by assigning the WantAuthnRequestsSigned attribute of the <md:IDPSSPDescriptor> element to a value of true. See further section E7, “Metadata for Agreeing to Sign Authentication Requests”, of [SAML v2.0 Errata 05].

4. Attributes

Attribute specifications for the Faroese eID Framework is defined in [AttrProf].

The content of <saml2:AttributeValue> elements exchanged via any SAML 2.0 messages or assertions SHOULD be limited to a single child text node.

5. Authentication Requests

5.1. Binding and Security Requirements

The endpoints, at which an Identity Provider receives a <saml2p:AuthnRequest> message, and all subsequent exchanges with the user agent, MUST be protected by TLS.

[SAML2Int] specifies that a <saml2p:AuthnRequest> message MUST be communicated to the Identity Provider using the HTTP-REDIRECT binding. This profile will also allow the usage of the HTTP-POST binding for sending <saml2p:AuthnRequest> messages (see section 3.5 of [SAML2Bind]), meaning that Identity Providers conformant with this profile MUST support the HTTP-POST binding.

An Identity Provider that requires <saml2p:AuthnRequest> messages to be signed MUST not accept messages that are not signed, or where the verification of the signature fails. In these cases the Identity Provider MUST respond with an error.

An Identity Provider that itself does not require authentication messages to be signed MUST still accept and verify signed request messages from Service Providers that indicate, in their metadata, that they sign request messages (see 2.1.2 above). If this signature verification fails, the Identity Provider MUST return a SAML error response and MUST NOT fulfill the request.

An Identity Provider that receives a request message that is not signed from a Service Provider that has indicated, in its metadata, that it will only send signed request messages (see 2.1.2 above) MUST respond with an error.

The signature for authentication request messages is applied differently depending on the binding. The HTTP-REDIRECT binding requires the signature to be applied to the URL-encoded value rather than being placed within the XML-message (see section 3.4.4.1 of [SAML2Bind]). For the HTTP-POST binding the <saml2p:AuthnRequest> element MUST be signed using a <ds:Signature> element within the<saml2:AuthnRequest>.

5.2. Message Content

[SAML2Int] specifies that a <saml2p:AuthnRequest> message SHOULD contain an AssertionConsumerServiceURL attribute identifying the desired response location. The Service Provider MUST NOT use any other values for this attribute than those listed in its metadata record as <md:AssertionConsumerService> elements for the HTTP-POST binding (see section 4.1.6 of [SAML2Prof]).

The Destination attribute of the <saml2p:AuthnRequest> message MUST contain the URL to which the Service Provider has instructed the user agent to deliver the request. This is useful to prevent malicious forwarding of signed requests from being accepted by unintended Identity Providers.

A Service Provider SHOULD explicitly specify one requested authentication context element (<saml2p:RequestedAuthnContext>), containing one or more <saml2:AuthnContextClassRef> elements that each contains an authentication context URI[1] representing a defined Level of Assurance under which the authentication process should be performed.

A present <saml2p:RequestedAuthnContext> element MUST specify exact matching by means of either an absent Comparison attribute or a Comparison attribute with the value set to exact. This means that the Identity Provider is forced to return an assertion with exactly one of the requested <saml2:AuthnContextClassRef> in the request as the declared <saml2:AuthnContext>, or return an error response. If the Service Provider requires the Identity Provider to return specifically one out of a selection of acceptable authentication context URIs, then all of these URIs MUST be included in the request.

```
<saml2p:
RequestedAuthnContext
Comparison="exact">
  <saml2:
AuthnContextClassRef>
http://id.samleiki.fo/loa/1.
0/substantial</saml2:
AuthnContextClassRef>
</saml2p:
RequestedAuthnContext>
```

Example of how an Authentication Context URI identifier representing a requested Level of Assurance is included in an authentication request message.

Identity Providers conformant with this profile MUST support the ForceAuthn and IsPassive attributes received in <saml2p:AuthnRequest> messages.

Service Providers SHOULD include the ForceAuthn attribute in all <saml2p:AuthnRequest> messages and explicitly set its value to true or false, and not rely on its default value. The reason for this is to avoid accidental SSO.

5.3. Processing Requirements

5.3.1. Validation of Destination

An Identity Provider receiving a <saml2p:AuthnRequest> message MUST verify that the Destination attribute is present, and that it is consistent with URLs configured in the Identity Provider's metadata.

5.3.2. Validation of Assertion Consumer Addresses

If the AssertionConsumerServiceURL attribute is present in the <saml2p:AuthnRequest> message, its value MUST be verified to be consistent with one of the <md:AssertionConsumerService> elements having the HTTP-POST binding found in the Service Provider's metadata entry. If this is not the case, the request must be rejected.

If the attribute is not present in the <saml2p:AuthnRequest> message, the Identity Provider MUST obtain the desired response location from the Service Provider's metadata entry. This location is found in an <md:AssertionConsumerService> element with HTTP-POST binding that is marked as default (has the isDefault attribute set), or if no element has the isDefault attribute set, the one with the lowest index value (see section 2.4.4.1 of [SAML2Meta]).

5.3.3. Identity Provider User Interface

Where the requirements for user interfaces defined for the federation requires presentation of information elements related to the Service Provider, these information elements MUST be obtained from the<mdui:UIInfo> element in the Service Provider's metadata entry. Implementers of this profile MUST be capable of handling display information stored in the <mdui:DisplayName>, <mdui:Logo> and the <mdui:Description> elements.

5.3.4. Authentication Context and Levels of Assurance Handling

The Samleikin federation defines a number of authentication context identifiers (URI), where each such identifier specifies a defined Level of Assurance and may define specific requirements on the authentication process. There can be multiple authentication context URIs representing the same Level of Assurance, but one authentication context URI always identifies one defined Level of Assurance. For example, requests for authentication from a Signature Service that requires a sign message to be displayed as part of the authentication process will request a different authentication context URI (see section 7) than a typical Service Provider just requesting authentication of a user, even if the requested Level of Assurance is the same.

Identity Providers SHALL exclusively use one of the requested authentication contexts in <saml2p:AuthnRequest> in the <saml2:AuthnContextClassRef> element under the <saml2p:RequestedAuthnContext> element, when present, to determine the requested authentication process and Level of Assurance. The Identity Provider SHALL respond with an error <saml2p:StatusCode> with the value urn:oasis:names:tc:SAML:2.0:status:Requester [SAML2Core] if no requested authentication context is supported. If no requested authentication context is present in the <saml2p:AuthnRequest>, the Identity Provider MAY return the result of a default authentication process that is consistent with the Identity Providers metadata.

5.3.5. Single Sign On Processing

An Identity Provider conformant to this profile MAY issue an assertion relying on a previously established security context (active session) instead of authenticating the user. However, the Identity Provider MUST NOT re-use an already existing security context in the following cases:

- When the security context has expired, i.e., the time elapsed since the security context was established is too long given the SSO-policy stipulated by the federation.
- When the <saml2p:AuthnRequest> contains a ForceAuthn attribute with the value of true.
- If the original authentication process, which led to the establishment of the security context, was performed using a weaker Level of Assurance that what is requested in the current<saml2p:AuthnRequest> message.

If the Identity Provider user interface contains some sort of user consent, or information, concerning which attributes, or any other information, that is included in an assertion being issued, the Identity Provider MUST preserve this functionality if a <saml2p:AuthnRequest> message requesting a different set of attributes (or any other information) compared to what was delivered in the assertion at the time of establishing the security context. The Identity Provider may require re-authentication or display a user interface for consent/information in these cases.

5.3.6. Single Logout Processing

Service Providers SHOULD support the SAML V2.0 SingleLogout profile [SAML2Prof], as updated by [SAML2Errata]. Service Providers that claim support for this profile MUST be capable of issuing logout requests. It is OPTIONAL to support consumption of logout requests and responses.

The intent is to allow for the minimum support possible while still enabling applications to initiate logouts in a federated environment. Thus, implementations must be able to generate a request to the Identity Provider, but may ignore (or not even support) responses, and may also omit support for inbound logout requests entirely.

Service Providers that support the SAML V2.0 SingleLogout profile MUST support the HTTP-Redirect binding for logout requests and responses.

Service Providers that support the SAML V2.0 SingleLogout profile MUST support decryption of <saml:EncryptedID> elements in logout requests. In order to fully support key rollover, Service Providers MUST be configurable with at least two decryption keys. When decrypting encrypted identifiers, they MUST attempt to use each decryption key (in unspecified order) until the identifier is successfully decrypted or there are no more keys, in which case decryption fails.

Service Providers that support the SAML V2.0 SingleLogout profile MUST support the consumption of peer configuration values from SAML metadata, without additional inputs or separate configuration, for any element listed in the "Use of Metadata" section for the Single Logout Profile in [SAML2Prof] (section 4.4.5).

6. Authentication Response

6.1. Security Requirements

The endpoint(s) at which a Service Provider receives a <saml2p:Response> message MUST be protected by TLS, no less than version 1.2.

The <saml2p:Response> message issued by the Identity Provider MUST be signed using a <ds:Signature> element within the <saml2p:Response> element.

The <saml2:Assertion> element issued by the Identity Provider SHALL be signed using a <ds:Signature> element within the <saml2:Assertion>.

Identity Providers SHALL utilize XML Encryption and return a <saml2:EncryptedAssertion> element in the <saml2p:Response> message. The elements <saml2:EncryptedID> and <saml2:EncryptedAttribute> MUST NOT be used; instead the entire assertion MUST be encrypted.

Service Providers SHOULD NOT accept unsolicited <saml2p:Response> messages (i.e., responses that are not the result of an earlier <saml2p:AuthnRequest> message). Service Providers that do accept unsolicited response messages MUST ensure, by other means, that the security and processing requirements of this profile (section 6.3) can be fully satisfied.

6.2. Message Content

The <saml2:Response> message MUST contain an <saml2:Issuer> element containing the unique identifier (entityID) of the issuing Identity Provider.

The AuthnInstant attribute of the <saml2:AuthnStatement> element MUST be assigned the time when the actual authentication took place. This time may differ from the IssueInstant attribute of the assertion itself, which holds the time when the assertion was issued. This is especially important in cases of re-use of already established security contexts at the Identity Provider side (Single Sign On).

Each identity assertion MUST have a <saml:Subject> element that specifies the principal that is the subject of all of the statements in the assertion.

The value of the <saml:NameID> element under the <saml:Subject> element MUST hold a pseudonym identifier of the subject, which SHALL be:

- Unique for the IdP – SP combination being the issuer and recipient for the assertion.
- Constructed in a manner that does not reveal the registered identity of the subject.

The <saml2:Subject> element MUST contain one <saml2:SubjectConfirmation> element containing a Method of urn:oasis:names:tc:SAML:2.0:cm:bearer. This element MUST contain a <saml2:SubjectConfirmationData> element that contains at least the following:

- An InResponseTo attribute matching the request's ID.
- A Recipient attribute containing the Service Provider's assertion consumer service URL.
- A NotOnOrAfter attribute containing a time instant at which the subject no longer can be confirmed.

The <saml2:SubjectConfirmationData> MUST also contain an Address attribute containing the network address from which an attesting entity (user) can present the assertion.

The assertion MUST contain a <saml2:Conditions> element containing the following attributes and elements:

- A <saml2:AudienceRestriction> element including the requesting Service Provider's unique identifier (entityID) as an <saml2:Audience> value.
- A NotBefore attribute specifying the earliest time instant at which the assertion is valid.
- A NotOnOrAfter attribute specifying the time instant when the assertion expires.

An Identity Provider conformant to this profile MUST, in its issued assertions, include an authentication context URI indicating under which Level of Assurance the assertion was issued. This identifier MUST be placed under the <saml2:AuthnStatement> element as the value of an <saml2:AuthnContextClassRef> element that is part of the <saml2:AuthnContext> element.

```
<saml2:AuthnStatement AuthnInstant=
"2020-03-01T09:22:00" SessionIndex=
"b07b804c-7c29-ea16-7300-4f3d6f7928ac"
>
  <saml2:AuthnContext>
    <saml2:AuthnContextClassRef>
http://id.samleiki.fo/loa/1.0
/substantial</saml2:
AuthnContextClassRef>
    ...
  </saml2:AuthnContext>
</saml2:AuthnStatement>
```

Example of how an Authentication Context URI identifier representing a Level of Assurance is included in an authentication statement.

An Identity Provider that acts as a proxy for other Identity Providers SHOULD include the <saml2:AuthenticatingAuthority> element under the <saml2:AuthnContext> element. This element will contain the entityID of the Identity Provider that was involved in authenticating the principal.

```
<saml2:AuthnStatement AuthnInstant=
"2017-03-15T09:22:00" SessionIndex=
"b07b804c-7c29-ea16-7300-4f3d6f7928ac"
>
  <saml2:AuthnContext>
    ...
    <saml2:AuthenticatingAuthority>
https://innrita2.samleiki.fo/auth</
saml2:AuthenticatingAuthority>
  </saml2:AuthnContext>
</saml2:AuthnStatement>
```

Example of how the entityID of an Identity Provider that provided the authentication for the principal is included in an authentication statement.

6.2.1. Attribute Release Rules

An Identity Provider determines which attributes to include in the <saml2:AttributeStatement> element of an assertion based on the Service Provider requirements and its agreements with the user being authenticated. Service Provider attribute preferences and requirements are specified in <md:AttributeConsumingService> elements declared in the Service Provider metadata. In the <md:AttributeConsumingService> element, sub-elements of type <md:RequestedAttribute> define the allowed attributes for each Service Provider. Note that each AttributeConsumingService MUST be given a unique index value that Service Providers MUST specify what index of the AttributeConsumingService in every SAML request message. A Service Provider MAY also specify one singular particular AttributeCosumingService index as IsDefault and not specifying a particular Index in a SAML request message. An Identity Provider MUST in the latter case use the AttributeConsumingService that is defined with the IsDefault parameter in the SAML metadata.

```
<md:AttributeConsumingService index="1">
  <md:ServiceName xml:lang="en">Name</md:ServiceName>
  <md:ServiceDescription xml:lang="en">Description</md:ServiceDescription>
  <md:RequestedAttribute FriendlyName="displayName" isRequired="false" Name="urn:oid:
2.16.840.1.113730.3.1.241" NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri"/>
  <md:RequestedAttribute FriendlyName="surname" isRequired="true" Name="urn:oid:2.5.4.4"NameFormat="urn:oasis:
names:tc:SAML:2.0:attrname-format:uri"/>
  <md:RequestedAttribute FriendlyName="givenName" isRequired="true"Name="urn:oid:2.5.4.42" NameFormat="urn:
oasis:names:tc:SAML:2.0:attrname-format:uri"/>
</md:AttributeConsumingService>
```

Example of how the AttributeConsumingService sub-elements RequestedAttribute are defined in metadata for a Service Provider.

An Identity Provider SHALL also register what attributes it can issue as part of its SAML metadata by using EntityAttributes.

For all declared EntityAttributes, the Identity Provider MUST possess the ability to deliver the mandatory attributes of the underlying attribute set. See [AttrProf] for details.

```
<md:Extensions>
<mdattr:EntityAttributes>
<saml:Attribute FriendlyName="urn:mace:dir:attribute-def:eduPersonPrincipalName" Name="urn:oid:
1.3.6.1.4.1.5923.1.1.1.6" NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri"
</saml:Attribute>
</mdattr:EntityAttributes>
</md:Extensions>>
```

Example of how an Identity Provider declare supported attributes in metadata.

The Service Provider is responsible for checking that an Identity Provider is capable of providing necessary attributes before sending a request and to verify that it received all attributes necessary for providing a requested service. An Identity Provider receiving a request for more attributes than it can provide SHOULD return an assertion with the attributes it can provide according to its defined attribute release policy, leaving it up to the Service Provider to decide how to proceed, e.g., by denying service to the authenticated user, provide limited services or to use other resources to collect necessary attributes.

6.3. Processing Requirements

This profile mandates a correct processing of a <saml2p:Response> message in order to ensure proper protection from the security threats described in [SAML2Sec]. Processing requirements are listed in [SAML2Core], [SAML2Prof] and [SAML2Sec]. This document will list the necessary requirements that apply to this profile.

After the Service Provider has decrypted the assertion from the received response message the following requirements apply. Any verification that fails MUST lead to that the Service Provider rejects the response message and does not use the assertion.

Some of the processing requirements below are defined in order to protect from MITM- or MITB-attacks where unsigned authentication requests may be changed before being sent to the Identity Provider. However, a Service Provider MUST implement all of the specified processing requirements even if it sends signed authentication request messages.

6.3.1. Signature Validation

The signature present on the <saml2p:Response> message, and optionally on the <saml2:Assertion>, MUST be successfully verified.

The public key being used to verify the signature MUST appear in the issuing Identity Provider's metadata record (as a <ds:X509Certificate> or <ds:KeyValue> element under the <ds:KeyInfo> element).

6.3.2. Subject Confirmation

Based on the InResponseTo attribute of the <saml2:SubjectConfirmationData> the Identity Provider MUST be able to obtain the corresponding <saml2p:AuthnRequest> message, or a secure context containing corresponding information from the request (for future processing of the assertion).

The Recipient attribute from the bearer <saml2:SubjectConfirmationData> element MUST match the location to which the <saml2p:Response> message was delivered **and** match the value the AssertionConsumerServiceURL attribute included in the request message, or if this attribute was not provided in the request message, the default response location specified in the Service Provider's metadata entry.

The time from the `NotOnOrAfter` attribute from the bearer `<saml2:SubjectConfirmationData>` MUST NOT have passed compared with the time instant at which the subject is confirmed (i.e., when the assertion is validated).

If the `Address` attribute is assigned to the bearer `<saml2:SubjectConfirmationData>` element, the Service Provider MAY choose to check the user agent's client address against it. Practical issues regarding the Service Provider's network setup and the risk of introducing false negatives makes this an optional step in the validation phase.

6.3.3. Conditions

The Service Provider MUST assert that the value of the `<saml2:Audience>` element under the `<saml2:AudienceRestriction>` element matches the unique entityID of the Service Provider.

The Service Provider MUST verify that the time instant at which the assertion is validated is within the range given by the `NotBefore` and `NotOnOrAfter` attributes of the `<saml2:Conditions>` element (allowing for a reasonable clock skew).

6.3.4. The Authentication Statement

The Service Provider MUST assert that the `<saml2:AuthnStatement>` contains a `<saml2:AuthnContext>` element that holds a `<saml2:AuthnContextClassRef>` element having as its value the authentication context URI indicating under which Level of Assurance the authentication was performed. The Level of Assurance declared in the assertion MUST be equal to, or stronger[1] than, the Level of Assurance requested by the Service Provider.

> *[1]: A stronger Level of Assurance identifier is simply a LoA having a higher value than what it is compared with, i.e., `http://id.samleiki.fo/loa/1.0/high` is stronger than `http://id.samleiki.fo/loa/1.0/substantial`.*

6.3.5. General Security Validation

In order to protect itself from replay attacks, the Service Provider MUST ensure that the same assertion is not processed more than once within the time it is valid (with respect to the `NotOnOrAfter` attribute of the `<saml2:Conditions>` element).

In order to prevent stolen assertions and user impersonation, the Service Provider MUST implement a validation that rejects an assertion if the time given in its `IssueInstant` attribute compared to the time when the response message is received is too great. This time is typically on the order of seconds, and limits the time window when a stolen assertion could be used.

If the Service Provider included the attribute `ForceAuthn` with a value of `true` in the authentication request, the Service Provider MUST ensure that the `AuthnInstant` attribute of the `<saml2:AuthnStatement>` element is greater than the time when the request was sent (allowing for a reasonable clock skew).

6.4. Error Responses

If the Identity Provider returns an error, it MUST NOT include any assertions in the `<saml2p:Response>` message.

An Identity Provider conformant with this profile SHOULD NOT make use of any other `<saml2p:StatusCode>` values than those specified in section 3.2.2.2 of [SAML2Core]. The top-level `<saml2p:StatusCode>` value may only be one of the following error identifiers:

- `urn:oasis:names:tc:SAML:2.0:status:Requester` – The request could not be performed due to an error on the part of the Service Provider.
- `urn:oasis:names:tc:SAML:2.0:status:Responder` – The request could not be performed due to an error on the part of the Identity Provider.
- `urn:oasis:names:tc:SAML:2.0:status:VersionMismatch` – The Identity Provider could not process the request because the version of the request message was incorrect.

If the user cancels an authentication process the Identity Provider SHOULD indicate this by assigning the second-level status code to `http://id.samleiki.fo/loa/1.0/cancel`.

If an Identity Provider displays information describing an error in its user interface it MUST also offer ways for the end user to confirm this information (for example, by including an OK-button). When the end user acknowledges taking part of the information (i.e., clicks on the OK-button), the `<saml2p:Response>` message is posted back to the Service Provider according to the HTTP POST binding [SAML2Bind].

7. Normative References

[RFC2119]

> *Bradner, S., Key words for use in RFCs to Indicate Requirement Levels, March 1997.*

[SAML2Int]

> *SAML V2.0 Deployment Profile for Federation Interoperability.*

[FedInterOp]

> *SAML V2.0 Implementation Profile for Federation Interoperability.*

[eGov]

> *Kantara Initiative eGovernment Implementation Profile of SAML V2.0 (Version 2.0bis).*

[SAML2Core]

> *OASIS Standard, Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.0, March 2005.*

[SAML v2.0 Errata 05]

> *SAML Version 2.0 Errata 05. 01 May 2012. OASIS Approved Errata.*

[SAML2Bind]

> *OASIS Standard, Bindings for the OASIS Security Assertion Markup Language (SAML) V2.0, March 2005.*

[SAML2Prof]

> *OASIS Standard, Profiles for the OASIS Security Assertion Markup Language (SAML) V2.0, March 2005.*

[SAML2Meta]

> *OASIS Standard, Metadata for the OASIS Security Assertion Markup Language (SAML) V2.0, March 2005.*

[SAML2Sec]

> *Security and Privacy Considerations for the OASIS Security Assertion Markup Language (SAML) V2.0, March 2005.*

[SAML2IAP]

> *SAML V2.0 Identity Assurance Profiles Version 1.0, 05 November 2010.*

[MetaIOP]

> *OASIS Committee Specification, SAML V2.0 Metadata Interoperability Profile Version 1.0, August 2009.*

[SAML2MetaUI]

> *OASIS Draft, SAML V2.0 Metadata Extensions for Login and Discovery User Interface Version 1.0, September 2010.*

[SAML2MetaAttr]

> *OASIS Committee Specification, SAML V2.0 Metadata Extension for Entity Attributes Version 1.0, August 2009.*

[IdpDisco]

> *OASIS Committee Specification, Identity Provider Discovery Service Protocol and Profile, March 2008.*

[EidRegistry]

> *Samleikin Registry of Identifiers*

[AttrProf]

> *Samleikin Attribute Specification*

[eIDAS]

> *REGULATION (EU) No 910/2014 OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL of 23 July 2014 on electronic identification and trust services for electronic transactions in the internal market and repealing Directive 1999/93/EC. Including implementation acts of the regulation and associated technical specifications.*

8. Changes Between Versions