SAML 2.0 protocol deployment profile

FOR THE FINNISH PUBLIC SECTOR

Version	Date	Changes
1.0	8.12.2010	Implementation by Ubisecure Solutions, Fujitsu Services and CSC — IT Center for Science. Approved by Ministry of Finance and Ministry of Employment and the Economy.
1.1	21.2.2011	Made it explicit that the SPs must sign the authentication requests. Added a description of Tunnistus.fi to the appendix.

Introduction

This is the specification of the Finnish public sector SAML 2.0 profile for being used in the Finnish public sector identity federation services, for example in VETUMA, tunnistus.fi and Virtu.

Tunnistus.fi and VETUMA are two citizen authentication services in the Finnish public sector. Virtu is an identity federation for authentication of civil servants in the Finnish state government. A short description of the federation services is provided in Appendix.

This specification aims at laying down the common grounds for the SAML 2.0 Web Single sign-on in the federation services in order to, for instance, ease interoperability and procurements. Additionally, the identity federation services may decide to use other alternative or complementary specifications, for example, due to backwards compatibility or sector-specific needs.

The semantics and syntax of the general user attributes to be exchanged in the Finnish public sector identity federation services are defined in a separate “SAML 2.0 attribute profile for the Finnish public sector” document and in the service-specific documentation.

This document

This document specifies a SAML 2.0 Web Browser SSO Deployment profile for the Finnish public sector federation services. The profile is based on the Kantara Initiative eGovernment Implementation profile of SAML 2.0 (version 2.0), and the Interoperable SAML 2.0 Web Browser SSO Deployment Profile (version 0.2).

The document follows the structure of Kantara eGovernment Implementation Profile, which is placed to the first column of the table below. In the second column, related sections from the Interoperable SSO Deployment profile are placed next to the related section of the Implementation profile. The third column contains deployment notices for Tunnistus.fi, VETUMA and Virtu services. The second and third column together with the “Additional extensions” section below the table form the SAML 2.0 profile defined in this document. If the second and third columns are empty, this profile takes no position with regards to the section in the first column. If the second and third columns are in conflict, the third column takes the precedence.

Kantara Initiative eGovernment Implementation Profile of SAML V2.0 [Kantara eGov] version 2.0, June 11, 2010	Interoperable SAML 2.0 Web Browser SSO Deployment Profile [SAML2int] ver 0.2 stable	The Finnish public sector deployment notice
1. Introduction SAML V2.0 is a rich and extensible standard that must be profiled to be used interoperably, and the profiles that typically emerge from the broader standardization process usually remain fairly broad and include a number of options and features that increase the burden for implementers and make deployment-time decisions more difficult. The Kantara Initiative eGovernment Implementation Profile provides a SAML V2.0 conformance specification for Identity Provider and Service Provider implementations operating in eGovernment federations and deployments. The profile is based on the SAML V2.0 specifications created by the Security Services Technical Committee (SSTC) of OASIS, and related specifications approved by that body. It constrains and supplements the base SAML V2.0 features, elements, and attributes required for eGovernment federations and deployments. Implementation profiles define the features that software implementations must support such that deployers can be assured of the ability to meet their own (possibly varied) deployment requirements. Deployment profiles define specific options and constraints to which deployments are required to conform; they guide product configuration and federation operations, and provide criteria against which actual deployments may be tested. This document does not include a deployment profile, but reflects the features deemed necessary or desirable from software implementations in support of a variety of deployment profiles planned and in use. This includes requirements deemed useful to further the eventual goal of		

interfederation between deployments.		
References to SAML 2.0 Specification		

1.1. Notation
 	
Conventional XML namespace prefixes are used throughout the listings in this specification to stand for their respective namespaces as follows, whether or not a namespace declaration is present in the example:
 	
• The prefix saml2: stands for the SAML 2.0 assertion namespace, urn:oasis:names:tc:SAML:2.0:assertion
 	
• The prefix saml2p: stands for the SAML 2.0 protocol namespace, urn:oasis:names:tc:SAML:2.0:protocol
 	
• The prefix md: stands for the SAML 2.0 metadata namespace, urn:oasis:names:tc:SAML:2.0:metadata
 	
• The prefix idpdisc: stands for the Identity Provider Discovery Service Protocol and Profile [IdPDisco] namespace, urn:oasis:names:tc:SAML:profiles:SSO:idp-discovery-protocol
 	
• The prefix mdattr: stands for the Metadata Extension for Entity Attributes Version 1.0 [MetaAttr] namespace, urn:oasis:names:tc:SAML:metadata:attribute
	4. References to SAML 2.0 Specification
	
	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 specification [SAML2Meta], the following syntax is used:
	
	`<md:Metadataelement>`
	
	When referring to elements from the Identity Provider Discovery Service Protocol and Profile [IdPDisco], the following syntax is used:
	
	`<idpdisc:DiscoveryResponse>`	

SAML 2.0 implementation and deployment profile		
2 SAML V2.0 Implementation Profile This profile specifies behavior and options that implementations of a selected set of SAML V2.0 profiles [SAML2Prof] are required to support. The requirements specified are in addition to all normative requirements of the original profiles, as modified by the Approved Errata [SAML2Err], and readers should be familiar with all relevant reference documents. Any such requirements are not repeated here except where deemed necessary to highlight a point of discussion or draw attention to an issue addressed in errata, but remain implied. SAML leaves substantial latitude to	**3. Introduction** This profile specifies behavior and options that deployments of the SAML V2.0 Web Browser SSO Profile [SAML2Prof] are required or permitted to rely on. The requirements specified are in addition to all normative requirements of the original profile, as modified by the Approved Errata [SAML2Err], and readers should be familiar with all relevant reference documents. Any such requirements are not repeated here except where deemed necessary to highlight a point of discussion or draw attention to an issue addressed in errata, but remain implied. This profile addresses the content,	

implementations with regard to how software is architected and combined with authentication and application infrastructure. Where the terms "Identity Provider" and "Service Provider" are used, they should be understood to include the total software footprint intended to provide the desired functionality; no specific assumptions are made as to how the required features are exposed to deployers, only that there is some method for doing so.	exchange, and processing of SAML messages only, and does not address deployment details that go beyond that scope. Furthermore, nothing in the profile should be taken to imply that disclosing personally identifiable information, or indeed any information, is required from an Identity Provider with respect to any particular Service Provider. That remains at the discretion of applicable settings, user consent, or other appropriate means in accordance with regulations and policies. Note that SAML features that are optional, or lack mandatory processing rules, are assumed to be optional and out of scope of this profile if not otherwise precluded or given specific processing rules.	
## Metadata		
2.2. Metadata and Trust management Identity Provider, Service Provider, and Discovery Service implementations MUST support the use of SAML V2.0 Metadata [SAML2Meta] in conjunction with their support of the SAML V2.0 profiles referenced by subsequent sections. Additional expectations around the use of particular metadata elements related to profile behavior may be encountered in those sections.	Identity Providers and Service Providers MUST provide a SAML 2.0 Metadata document representing its entity.	
### Keys in metadata		
2.2.1. Metadata profiles Implementations MUST support the SAML V2.0 Metadata Interoperability Profile Version 1.0 [MetaIOP].	Provided metadata MUST conform to the SAML V2.0 Metadata Interoperability Profile Version 1.0 [MetaIOP].	VETUMA requires a valid certificate issued by a CA approved by VIP. Virtu requires a valid certificate issued by VRK CA for Service Providers.
In addition, implementations MUST support the use of the <md:KeyDescriptor> element as follows: • Implementations MUST support the <ds:X509Certificate> element as input to subsequent requirements. Support for other key representations, and for other mechanisms for credential distribution, is	If a Service Provider forgoes the use of TLS/SSL for its Assertion Consumer Service endpoints, then its metadata SHOULD include a `<md:KeyDescriptor>` suitable for XML Encryption. Note that use of TLS/SSL is RECOMMENDED.	Service Providers in Tunnistus.fi, Virtu and VETUMA MUST use TLS/SSL for Assertion Consumer Service endpoints. During a SAML protocol exchange, the relying party MUST

OPTIONAL. • Implementations MUST support some form of path validation of signing, TLS, and encryption credentials used to secure SAML exchanges against one or more trusted certificate authorities. Support for PKIX [RFC5280] is RECOMMENDED; implementations SHOULD document the behavior of the validation mechanisms they employ, particular with respect to limitations or divergence from PKIX [RFC5280]. • Implementations MUST support the use of OCSP [RFC2560] and Certificate Revocation Lists (CRLs) obtained via the "CRL Distribution Point" X.509 extension [RFC5280] for revocation checking of those credentials. • Implementations MAY support additional constraints on the contents of certificates used by particular entities, such as "subjectAltName" or "DN", key usage constraints, or policy extensions, but SHOULD document such features and make them optional to enable where possible. Note that these metadata profiles are intended to be mutually exclusive within a given deployment context; they are alternatives, rather than complimentary or compatible uses of the same metadata information.		either verify the validity of the metadata file containing the peer entity or verify the validity of the certificate used by the peer entity for protecting the SAML exchange.
Other metadata contents		
Implementations SHOULD support the SAML V2.0 Metadata Extension for Entity Attributes Version 1.0 [MetaAttr] and provide policy controls on the basis of SAML attributes supplied via this extension mechanism.	Metadata documents provided by an Identity Provider MUST include an `<md:IDPSSODescriptor>` element containing all necessary `<md:KeyDescriptor>` and `<md:SingleSignOnService>` elements. The metadata SHOULD include one or more `<md:NameIDFormat>` elements indicating which `<saml2:NameID>` Format values are supported. Metadata documents provided by a Service Provider MUST include an `<md:SPSSODescriptor>` element containing all necessary `<md:KeyDescriptor>` and `<md:AssertionConsumerService>`	Tunnistus.fi and VETUMA Identity Providers support only the transient NameIDFormat. In VETUMA and Tunnistus.fi, metadata documents provided by an Identity and Service Provider MUST include and Virtu MAY include a `<md:SingleLogoutService>` element. See section Single Logout below for

	elements. The metadata SHOULD also include one or more `<md:NameIDFormat>` elements indicating which `<saml2:NameID>` Format values are supported and one or more `<md:AttributeConsumingService>` elements describing the service(s) offered and their attribute requirements. Metadata provided by Service Provider SHOULD also contain a descriptive name of the service that the Service Provider represents (not the company) in at least English. It is RECOMMENDED to also provide the name in other languages which is much used in the geographic scope of the deployment. The name should be placed in the `<md:ServiceName>` in the `<md:AttributeConsumingService>` container. Metadata provided by both Identity Providers and Service Provider SHOULD contain contact information for *support* and for a *technical contact.* The `<md:EntityDescriptor>` element SHOULD contain both a `<md:ContactPerson>` element with a `contactType` of "`support`" and a `<md:ContactPerson>` element with a `contactType` of "`technical`". The `<md:ContactPerson>` elements SHOULD contain at least one `<md:EmailAddress>`. The *support* address MAY be used for generic support questions about the service, while the *technical* contact may be contacted regarding technical interoperability problems. The *technical contact* MUST be responsible for the technical operation of the system(s) reflected in the metadata.	details.
Metadata exchange		
2.2.2. Metadata exchange It is OPTIONAL for implementations to support the generation or exportation of metadata, but implementations MUST support the publication of metadata using the Well-Known-Location method defined in section 4.1 of [SAML2Meta] (under the assumption that entityID values used are suitable for such	How metadata is exchanged is out of scope of this specification. Entities SHOULD publish its metadata using the Well-Known Location method defined in [SAML2Meta].	See Tunnistus.fi and VETUMA documentation to locate the Identity Providers' metadata document and metadata exchange practices. See Virtu

support). Implementations MUST support the following mechanisms for the importation of metadata: • local file • remote resource at fixed location accessible via HTTP 1.1 [RFC2616] or HTTP 1.1 over TLS/SSL [RFC2818] In the case of HTTP resolution, implementations MUST support use of the "ETag" and "Last-Modified" headers for cache management. Implementations SHOULD support the use of more than one fixed location for the importation of metadata, but MAY leave their behavior unspecified if a single entity's metadata is present in more than one source. Importation of multiple entities' metadata contained within an `<md:EntitiesDescriptor>` element MUST be supported. Finally, implementations SHOULD allow for the automated updating/reimportation of metadata without service degradation or interruption.		documentation to locate the federation's metadata document and metadata exchange practices.
Metadata verification		
2.2.2.1. Metadata verification Verification of metadata, if supported, MUST include XML signature verification at least at the root element level, and SHOULD support the following mechanisms for signature key trust establishment: • Direct comparison against known keys. • Some form of path-based certificate validation against one or more trusted certificate authorities, along with certificate revocation lists and/or OCSP [RFC2560]. Support for PKIX [RFC5280] is RECOMMENDED; implementations SHOULD document the behavior of the validation mechanisms they employ, particular with respect to limitations or divergence from		In Tunnistus.fi and VETUMA, the Identity Providers MUST publish metadata which is signed by a certificate issued by a CA as described in the service documentation. In Virtu, metadata MUST be signed by a certificate issued by VRK CA for Service Providers. Certificate validation instructions provided by the CA MUST be followed.

PKIX [RFC5280].		
## Name identifier support		

2.3. Name Identifiers
In conjunction with their support of the SAML V2.0 profiles referenced by subsequent sections, Identity Provider and Service Provider implementations MUST support the following SAML V2.0 name identifier formats, in accordance with the normative obligations associated with them by [SAML2Core]:
`urn:oasis:names:tc:SAML:2.0:nameid-format:persistent`
`urn:oasis:names:tc:SAML:2.0:nameid-format:transient`
Support for other formats is OPTIONAL.
	Identity Providers MUST support the `urn:oasis:names:tc:SAML:2.0:nameid-format:transient` name identifier format [SAML2Core]. They SHOULD support the `urn:oasis:names:tc:SAML:2.0:nameid-format:persistent` name identifier format [SAML2Core]. Support for other formats is OPTIONAL.
	Service Providers, if they rely at all on particular name identifier formats, MUST support one of the following:
	`urn:oasis:names:tc:SAML:2.0:nameid-format:persistent`
	`urn:oasis:names:tc:SAML:2.0:nameid-format:transient`
	Reliance on other formats by Service Providers is NOT RECOMMENDED.
	Note that these requirements are reflected in additional constraints on message content in subsequent sections.
		In Tunnustus.fi and VETUMA, transient NameIDFormat MUST be used.

## Attribute representation		
2.4. Attributes In conjunction with their support of the SAML V2.0 profiles referenced by subsequent sections, Identity Provider and Service Provider implementations MUST support the generation and consumption of `<saml2:Attribute>` elements that conform to the SAML V2.0 X.500/LDAP Attribute Profile [SAML-X500].	Any `<saml2:Attribute>` elements exchanged via any SAML 2.0 messages, assertions, or metadata MUST contain a `NameFormat` of `urn:oasis:names:tc:SAML:2.0:attrname-format:uri`. The use of LDAP/X.500 attributes and the LDAP/X.500 attribute profile [X500SAMLattr] is RECOMMENDED where possible.	See a separate document SAML 2.0 attribute profile for the Finnish public sector
The ability to support `<saml2:AttributeValue>` elements whose values are not simple strings (e.g., `<saml2:NameID>`, or other XML values) is OPTIONAL. Such content could be base64-encoded as an alternative.	It is RECOMMENDED that the content of `<saml2:AttributeValue>` elements exchanged via any SAML 2.0 messages, assertions, or metadata be limited to a single child text node (i.e., a simple string value).	
	Many identity federation use cases rely on the exchange of a so-called "targeted" or "pair-wise" user identifier that is typically opaque and varies for a given user when accessing different Service Providers. Various approaches to this	In Tunnistus.fi and VETUMA, transient NameIDFormat MUST be used.

	compatible with SAML exist, including the SAML 2.0 "`persistent`" Name Identifier format [SAML2Core], the `eduPersonTargetedID` attribute [eduPerson], and the *Private Personal Identifier claim* [IMI]. This profile RECOMMENDS the use of the `<saml2:NameID>` element (within the `<saml2:Subject>` element), carried within the `<saml2:Subject>` with a `Format` of `urn:oasis:names:tc:SAML:2.0:nameid-format:persistent` when an identifier of this nature is required. If an opaque targeted user identifier is being provided to the Service Provider, it is RECOMMENDED to use a `<saml2:NameID>` construct with a `Format` of `urn:oasis:names:tc:SAML:2.0:nameid-format:persistent` rather than transporting that identifier as an `<saml2:Attribute>`.	
## Browser Single Sign-On		
### Identity Provider Discovery		
2.5.1. Identity Provider Discovery Service Provider and Discovery Service implementations MUST support the Identity Provider Discovery Service Protocol Profile in conformance with section 2.4.1 of [IdPDisco].	If a Service Provider plans to utilize a Discovery Service supporting the Identity Provider Discovery Service Protocol Profile [IdPDisco], then its metadata MUST include one or more `<idpdisc:DiscoveryResponse>` elements in the `<md:Extensions>` element of its `<md:SPSSODescriptor>` element.	Currently, Tunnistus.fi and VETUMA do not utilize Identity Provider Discovery Service Protocol.
### Authentication requests		
2.5.2.1. Binding and Security Requirements Identity Provider and Service Provider implementations MUST support the use of the HTTP-Redirect binding [SAML2Bind] for the transmission of <saml2p:AuthnRequest> messages, including the generation or verification of signatures in conjunction with this binding. Support for other bindings is OPTIONAL.	The `<saml2p:AuthnRequest>` message issued by a Service Provider MUST be communicated to the Identity Provider using the `HTTP-REDIRECT` binding [SAML2Bind].	

	Identity Providers MAY omit the verification of signatures in conjunction with this binding.	Service Providers MUST sign the Authentication requests. Currently, Tunnistus.fi and VETUMA do verify the signature and reject requests with an invalid or missing signature.
	The endpoints at which an Identity Provider receives a `<saml2p:AuthnRequest>` message, and all subsequent exchanges with the user agent, SHOULD be protected by TLS/SSL.	Tunnistus.fi and VETUMA Identity Providers have a TLS endpoint for Authentication requests and all subsequent exchanges with the user agent. Virtu registers only TLS/SSL endpoints for Identity Providers.

2.5.2.2. Message Content
In addition to standard core- and profile-driven requirements, Service Provider implementations MUST support the inclusion of at least the following `<saml2p:AuthnRequest>` child elements and attributes (when appropriate):
`AssertionConsumerServiceURL`
`ProtocolBinding`
`ForceAuthn`
`IsPassive`
`AttributeConsumingServiceIndex`
`<saml2p:RequestedAuthnContext>`
`<saml2p:NameIDPolicy>`
Identity Provider implementations MUST support all `<saml2p:AuthnRequest>` child elements and attributes defined by [SAML2Core], but MAY provide that support in the form of returning appropriate errors when confronted by particular request options. However, implementations MUST fully support the options enumerated above, and be configurable to utilize those options in a useful manner as defined by
	The `<saml2p:AuthnRequest>` message issued by a Service Provider MUST contain an `AssertionConsumerServiceURL` attribute identifying the desired response location.
	The `ProtocolBinding` attribute, if present, MUST be set to `urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST`.
	The `<saml2p:AuthnRequest>` message MUST NOT contain a `<saml2:Subject>` element.
	The `<saml2p:AuthnRequest>` message SHOULD contain a `<saml2p:NameIDPolicy>` element with an `AllowCreate` attribute of "`true`".
	Its `Format` attribute, if present, SHOULD be set to one of the following values:
	`urn:oasis:names:tc:SAML:2.0:nameid-format:persistent`
	`urn:oasis:names:tc:SAML:2.0:nameid-format:transient`
		In Tunnustus.fi and VETUMA, transient NameIDFormat MUST be used.

[SAML2Core].		
Implementations MAY limit their support of the `<saml2p:RequestedAuthnContext>` element to the value "exact" for the `Comparison` attribute, but MUST otherwise support any allowable content of the element.	The `<saml2p:AuthnRequest>` message MAY contain a `<saml2p:RequestedAuthnContext>` element, but SHOULD do so only in the presence of an arrangement between the Identity and Service Providers regarding the Authentication Context definitions in use. The `Comparison` attribute SHOULD be omitted or be set to "`exact`".	SPs MUST request a specific level of assurance with the “exact” compare operator. The SP may request more than one level in priority order. E.g. this is useful when a level 2 is required but the SP is willing to accept (and perhaps pay for) a level 3 if a level 2 is not possible.
Identity Provider implementations MUST support verification of requested `AssertionConsumerServiceURL` locations via comparison to `<md:AssertionConsumerService>` elements supplied via metadata using case-sensitive string comparison. It is OPTIONAL to support other means of comparison (e.g., canonicalization or other manipulation of URL values) or alternatve verification mechanisms.	In verifying the Service Provider's Assertion Consumer Service, it is RECOMMENDED that the Identity Provider perform a case-sensitive string comparison between the requested `<saml2p:AssertionConsumerServiceURL>` value and the values found in the Service Provider's metadata. It is OPTIONAL to apply any form of URL canonicalization, which means the Service Provider SHOULD NOT rely on differently canonicalized values in these two locations. As an example, the Service Provider SHOULD NOT use a hostname with port number (such as `https://sp.example.no:80/acs`) in its request and without (such as `https://sp.example.no/acs`) in its metadata.	
Responses		
2.5.3.1. Binding and Security Requirements Identity Provider and Service Provider implementations MUST support the use of the HTTP-POST and HTTP-Artifact bindings [SAML2Bind] for the transmission of `<saml2p:Response>` messages. Support for other bindings, and for artifact types other than `urn:oasis:names:tc:SAML:2.0:artifact-04`, is OPTIONAL.	The `<saml2p:Response>` message issued by an Identity Provider MUST be communicated to the Service Provider using the `HTTP-POST` binding [SAML2Bind].	
	In the absence of a `<saml2p:NameIDPolicy>` Format	In Tunnistus.fi and VETUMA, transient

	attribute in the Service Provider's <saml2p:AuthnRequest> message, or a <md:NameIDFormat> element in the Service Provider's metadata, the Format of the <saml2:NameID> SHOULD be set to urn:oasis:names:tc:SAML:2.0:nameid-format:transient.	NameIDFormat MUST be used.
Identity Provider and Service Provider implementations MUST support the generation and consumption of unsolicited <saml2p:Response> messages (i.e., responses that are not the result of a <saml2p:AuthnRequest> message).	Service Providers MUST support unsolicited <saml2p:Response> messages (i.e., responses that are not the result of an earlier <saml2p:AuthnRequest> message).	Tunnistus.fi and VETUMA do not send unsolicited Responses.
Identity Provider implementations MUST support the issuance of <saml2p:Response> messages (with appropriate status codes) in the event of an error condition, provided that the user agent remains available and an acceptable location to which to deliver the response is available. The criteria for "acceptability" of a response location are not formally specified, but are subject to Identity Provider policy and reflect its responsibility to protect users from being sent to untrusted or possibly malicious parties. Note that this is a stronger requirement than the comparable language in [SAML2Prof].		
Identity Provider and Service Provider implementations MUST support the signing of <saml2:Assertion> elements in responses; support for signing of the <saml2p:Response> element is OPTIONAL.	Whether encrypted or not, the <saml2:Assertion> element issued by the Identity Provider MUST itself be signed directly using a <ds:Signature> element within the <saml2:Assertion>.	
Identity Provider and Service Provider implementations MUST support the use of XML Encryption via the <saml2:EncryptedAssertion> element when using the HTTP-POST binding; support for the <saml2:EncryptedID> and <saml2:EncryptedAttribute> elements is OPTIONAL.	The endpoint(s) at which a Service Provider receives a <saml2p:Response> message SHOULD be protected by TLS/SSL. If this is not the case, then Identity Providers SHOULD utilize XML Encryption and return a <saml2:EncryptedAssertion> element in the <saml2p:Response> message. The use of the <saml2:EncryptedID> and <saml2:EncryptedAttribute> elements is NOT RECOMMENDED; when possible, encrypt the entire	Tunnistus.fi, Virtu and VETUMA require TLS/SSL for Assertion Consumer Service endpoints. Currently, Tunnistus.fi and VETUMA Identity Providers do not send encrypted assertions, responses, attributes or nameIDs.

	assertion.	
Response content		
2.5.3.2. Message Content The Web Browser SSO Profile allows responses to contain any number of assertions and statements. Identity Provider implementations MUST allow the number of `<saml2:Assertion>`, `<saml2:AuthnStatement>`, and `<saml2:AttributeStatement>` elements in the `<saml2p:Response>` message to be limited to one. In turn, Service Provider implementations MAY limit support to a single instance of those elements when processing `<saml2p:Response>` messages.	Assuming a successful response, the `<saml2p:Response>` message issued by an Identity Provider MUST contain exactly one assertion (either a `<saml2:Assertion>` or an `<saml2:EncryptedAssertion>` element). The assertion MUST contain exactly one `<saml2:AuthnStatement>` element and MAY contain zero or one `<saml2:AttributeStatement>` elements.	
Identity Provider implementations MUST support the inclusion of a `Consent` attribute in `<saml2p:Response>` messages, and a `SessionIndex` attribute in `<saml2:AuthnStatement>` elements.		
Service Provider implementations that provide some form of session semantics MUST support the `<saml2:AuthnStatement>` element's `SessionNotOnOrAfter` attribute.		
Service Provider implementations MUST support the acceptance/rejection of assertions based on the content of the `<saml2:AuthnStatement>` element's `<saml2:AuthnContext>` element. Implementations also MUST support the acceptance/rejection of particular `<saml2:AuthnContext>` content based on the identity of the Identity Provider. [IAP] provides one such mechanism via SAML V2.0 metadata and is RECOMMENDED; though this specification is in draft form, the technical details are not expected to change prior to eventual approval.		To avoid man-in-the-middle attacks, a Service Provider which has used a RequestedAuthenticationContext in the AuthenticationRequest MUST verify that the AuthnContext of the Response satisfies its needs.
	The `<saml2:Subject>` element of the assertions issued by an Identity Provider SHOULD contain a `<saml2:NameID>` element. The `<saml2:Subject>` element MUST NOT include a `<saml2:BaseID>` nor a	The assertions issued by Tunnistus.fi and VETUMA Identity Providers contain a NameID element. Its format is transient.

	`<saml2:EncryptedID>`.	
## Artifacts		
2.5.4. Artifact Resolution Pursuant to the requirement in section 2.5.3.1 for support of the HTTP-Artifact binding [SAML2Bind] for the transmission of `<saml2p:Response>` messages, implementations MUST support the SAML V2.0 Artifact Resolution profile [SAML2Prof] as constrained by the following subsections. **2.5.4.1. Artifact Resolution Requests** Identity Provider and Service Provider implementations MUST support the use of the SAML SOAP (using HTTP as a transport) binding [SAML2Bind] for the transmission of `<saml2p:ArtifactResolve>` messages. Implementations MUST support the use of SAML message signatures and TLS server authentication to authenticate requests; support for TLS client authentication, or other forms of authentication in conjunction with the SAML SOAP binding, is OPTIONAL. **2.5.4.2. Artifact Resolution Responses** Identity Provider and Service Provider implementations MUST support the use of the SAML SOAP (using HTTP as a transport) binding [SAML2Bind] for the transmission of `<saml2p:ArtifactResponse>` messages. Implementations MUST support the use of SAML message signatures and TLS server authentication to authenticate responses; support for TLS client authentication, or other forms of authentication in conjunction with the SAML SOAP binding, is OPTIONAL.		Artifacts MUST NOT be used.
## Browser Holder of Key Single Sign-On		
This section defines an implementation profile of the SAML V2.0 Holder-of-Key Web Browser SSO Profile Version 1.0 [HoKSSO]. The implementation requirements		

defined in section 2.5 for the non-holder-of-key profile apply to implementations of this profile.		
## SAML 2.0 Proxying		
2.7. SAML 2.0 Proxying Section 3.4.1.5 of [SAML2Core] defines a formalized approach to proxying the SAML 2.0 Authentication Request protocol between multiple Identity Providers. This section defines an implementation profile for this behavior suitable for composition with the Single Sign-On profiles defined in sections 2.5 and 2.6. The requirements of the profile are imposed on Identity Provider implementations acting as a proxy. These requirements are in addition to the technical requirements outlined in section 3.4.1.5.1 of [SAML2Core], which also MUST be supported. ### 2.7.1. Authentication Requests Proxying Identity Provider implementations MUST support the mapping of incoming to outgoing `<saml2p:RequestedAuthnContext>` and `<saml2p:NameIDPolicy>` elements, such that deployers may choose to pass through values or map between different vocabularies as required. Proxying Identity Provider implementations MUST support the suppression/eliding of `<saml2p:RequesterID>` elements from outgoing `<saml2p:AuthnRequest>` messages to allow for hiding the identity of the Service Provider from proxied Identity Providers. ### 2.7.2. Responses Proxying Identity Provider implementations MUST support the mapping of incoming to outgoing `<saml2:AuthnContext>` elements, such that deployers may choose to pass through values or map between different vocabularies as required.	Identity Providers that act as a proxy (per section 3.4.1.5.1 of [SAML2Core]) MUST support `<saml2p:AuthnRequest>` messages that do not contain a `<saml2p:Scoping>` element.	

Proxying Identity Provider implementations MUST support the suppression of `<saml2:AuthenticatingAuthority>` elements from outgoing `<saml2:AuthnContext>` elements to allow for hiding the identity of the proxied Identity Provider from Service Providers.		
## Single Logout		
2.8. Single logout This section defines an implementation profile of the SAML V2.0 Single Logout Profile [SAML2Prof]. For clarification, the technical requirements for each message type below reflect the intent to normatively require initiation of logout by a Service Provider using either the front- or back-channel, and initiation/propagation of logout by an Identity Provider using the back-channel. ### 2.8.1. Logout Requests **2.8.1.1. Binding and Security Requirements** Identity Provider implementations MUST support the SAML SOAP (using HTTP as a transport) binding [SAML2Bind] for the issuance of `<saml2p:LogoutRequest>` messages, and MUST support the SAML SOAP (using HTTP as a transport) and HTTP-Redirect bindings [SAML2Bind] for the reception of `<saml2p:LogoutRequest>` messages. Service Provider implementations MUST support the SAML SOAP (using HTTP as a transport) binding [SAML2Bind] for both issuance and reception of `<saml2p:LogoutRequest>` messages. Support for other bindings is OPTIONAL. Implementations MUST support the use of SAML message signatures and TLS server authentication to authenticate `<saml2p:LogoutRequest>`		See the Single Logout definitions for Tunnistus.fi, VETUMA and Virtu in the end of the document.

messages; support for TLS client authentication, or other forms of authentication in conjunction with the SAML SOAP binding, is OPTIONAL. Identity Provider and Service Provider implementations MUST support the use of XML Encryption via the `<saml2:EncryptedID>` element when using the HTTP-Redirect binding. **2.8.1.2. User Interface Behavior** Identity Provider implementations MUST support both user-initiated termination of the local session only and user-initiated Single Logout. Upon receipt of a `<saml2p:LogoutRequest>` message via a front-channel binding, Identity Provider implementations MUST support user intervention governing the choice of propagating logout to other Service Providers, or limiting the operation to the Identity Provider. Of course, implementations MUST return status information to the requesting entity (e.g. partial logout indication) as appropriate. Service Provider implementations MUST support both user-initiated termination of the local session only and user-initiated Single Logout. Identity Provider implementations MUST also support the administrative initiation of Single Logout for any active session, subject to appropriate policy. ## 2.8.2. Logout Responses **2.8.2.1. Binding and Security Requirements** Identity Provider implementations MUST support the SAML SOAP (using HTTP as a transport) and HTTP-Redirect bindings [SAML2Bind] for the issuance of `<saml2p:LogoutResponse>` messages, and MUST support the SAML SOAP (using HTTP as a transport) binding [SAML2Bind] for the reception of `<saml2p:LogoutResponse>` messages. Service Provider implementations MUST support the SAML SOAP (using HTTP as a transport) binding [SAML2Bind] for		

both issuance and reception of `<saml2p:LogoutResponse>` messages. Support for other bindings is OPTIONAL. Implementations MUST support the use of SAML message signatures and TLS server authentication to authenticate `<saml2p:LogoutResponse>` messages; support for TLS client authentication, or other forms of authentication in conjunction with the SAML SOAP binding, is OPTIONAL.		
## Conformance classes		

2. Conformance classes
 	
3.1. Standard
 	
Conforming Identity Provider and/or Service Provider implementations MUST support the normative requirements in sections 2.2, 2.3, 2.4, and 2.5.
 	
3.2. Signature and Encryption Algorithms
 	
Implementations MUST support the signature and digest algorithms identified by the following URIs in conjunction with the creation and verification of XML Signatures [XMLSig]:
 	
`http://www.w3.org/2001/04/xmldsig-more#rsa-sha256` (defined in [RFC4051])
 	
`http://www.w3.org/2001/04/xmlenc#sha256` (defined in [XMLEnc])
 	
Implementations SHOULD support the signature and digest algorithms identified by the following URIs in conjunction with the creation and verification of XML Signatures [XMLSig]:
 	
`http://www.w3.org/2001/04/xmldsig-more#ecdsa-sha256` (defined in [RFC4051])
 	
Implementations MUST support the block encryption algorithms identified by the following URIs in conjunction with the use of XML Encryption [XMLEnc]:
 	
`http://www.w3.org/2001/04/xmlenc#tripledes-cbc`
 	
`http://www.w3.org/2001/04/xmle`		

nc#aes128-cbc http://www.w3.org/2001/04/xmlenc#aes256-cbc Implementations MUST support the key transport algorithms identified by the following URIs in conjunction with the use of XML Encryption [XMLEnc]: http://www.w3.org/2001/04/xmlenc#rsa-1_5 http://www.w3.org/2001/04/xmlenc#rsa-oaep-mgf1p Implementations SHOULD support the key agreement algorithms identified by the following URIs in conjunction with the use of XML Encryption [XMLEnc]: http://www.w3.org/2009/xmlenc11#ECDH-ES (defined in [XMLEnc11]) (This is a Last Call Working Draft of XML Encryption 1.1, and this normative requirement is contingent on W3C ratification of this specification without normative changes to this algorithm's definition.) Support for other algorithms is OPTIONAL. **3.2. Standard with Logout** Conforming Identity Provider and/or Service Provider implementations MUST meet the conformance requirements in section 3.1, and MUST in addition support the normative requirements in section 2.8. **3.3. Full** Conforming Identity Provider and/or Service Provider implementations MUST meet the conformance requirements in section 3.1, and MUST in addition support the normative requirements in sections 2.6, 2.7, and 2.8.		

Additional extensions

Authentication context for strong authentication

This profile defines the following authentication context:

authnContextClassRef URI	Description

http://www.valtiokonttori.fi/vip/AuthnContext/strong	Strong authentication as defined by the act on strong authentication (laki vahvasta sähköisestä tunnistamisesta ja sähköisestä allekirjoituksesta, 7.8.2009/617).

In VETUMA Service, Government IT service unit (VIP) and in Tunnistus.fi, KATVE consortium defines the permitted authentication mechanisms and which of them count as authentication context class strong.

In Virtu federation, the home organization decides when the authentication it has performed counts as authentication context class strong.

Single logout

The Single Logout protocol of SAML 2.0 as defined in section 3.7 of SAML 2.0 Core specification is

- REQUIRED for Identity and Service Providers in Tunnistus.fi and VETUMA
- OPTIONAL for Identity and Service Providers in Virtu federation. Providers manifest their support to Single Logout by providing a Single Logout endpoint in their metadata.

All Service Providers that support Single Logout MUST both be able to initiate a logout, send a Logout request to the Identity Provider, as well as handle an incoming Logout request from the Identity Provider.

HTTP-REDIRECT binding MUST be used for logout requests and responses. Logout requests and responses MUST be signed.

To ensure the user experience, an Identity Provider MUST provide means for a user to assure of a succesful logout. In the event of an unsuccessful logout to a Service Provider, the Identity Provider MUST instruct the user of steps involved in finishing the logout process. A Service Provider registering a Single Logout endpoint MUST make sure that the end user's session is terminated in the application level, as well.

References

[Kantara eGov] Kantara Initiative. eGovernment Implementation profile of SAML 2.0. Version 2.0 Draft Recommendation, June 11 2010.

[SAML2int] Interoperable SAML 2.0 Web Browser SSO Deployment Profile. Version 0.2 Stable. http://saml2int.org/profile/0.2

[SAML2Bind] OASIS Standard, Bindings for the OASIS Security Assertion Markup Language (SAML) V2.0, March 2005. http://docs.oasis-open.org/security/saml/v2.0/saml-bindings-2.0-os.pdf

[SAML2Core] OASIS Standard, Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.0, March 2005. http://docs.oasis-open.org/security/saml/v2.0/saml-core-2.0-os.pdf

[SAML2Err] OASIS Approved Errata, SAML V2.0 Errata, Dec 2009. http://www.oasis-open.org/committees/download.php/37166/sstc-saml-approved-errata-2.0-02.pdf

[SAML2Meta] OASIS Standard, Metadata for the OASIS Security Assertion Markup Language (SAML) V2.0, March 2005. http://docs.oasis-open.org/security/saml/v2.0/saml-metadata-2.0-os.pdf

[SAML2Prof] OASIS Standard, Profiles for the OASIS Security Assertion Markup Language (SAML) V2.0, March 2005. http://docs.oasis-open.org/security/saml/v2.0/saml-profiles-2.0-os.pdf

[SAMLX500], [X500SAMLattr] OASIS Committee Specification, SAML V2.0 X.500/LDAP Attribute Profile, March 2008. http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-attribute-x500.pdf

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

[IAP] OASIS Committee Draft, Identity Assurance Profiles, Version 1.0, September 2009. http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-assurance-profile-cd-01.pdf

[IdPDisco] OASIS Committee Specification, Identity Provider Discovery Service Protocol and Profile, March 2008. http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-idp-discovery.pdf

[MetaAttr] OASIS Committee Specification, SAML V2.0 Metadata Extension for Entity Attributes Version 1.0, August 2009. http://docs.oasis-open.org/security/saml/Post2.0/sstc-metadata-attr.pdf

[MetaIOP] OASIS Committee Specification, SAML V2.0 Metadata Interoperability Profile Version 1.0, August 2009. http://docs.oasis-open.org/security/saml/Post2.0/sstc-metadata-iop.pdf

[XMLEnc] D. Eastlake et al. XML Encryption Syntax and Processing. World Wide Web Consortium Recommendation. http://www.w3.org/TR/2002/REC-xmlenc-core-20021210/

[XMLEnc11] D. Eastlake et al. XML Encryption Syntax and Processing Version 1.1. World Wide Web Consortium Last Call Working Draft. http://www.w3.org/TR/2010/WD-xmlenc-core1-20100513/

[XMLSig] D. Eastlake et al. XML-Signature Syntax and Processing, Second Edition. World Wide Web Consortium Recommendation, June 2008. http://www.w3.org/TR/xmldsig-core/

Appendix: Short descriptions of the relying federation services

Tunnistus.fi citizen and company authentication service

Tunnistus.fi is a joint authentication service of the Tax Administration, Ministry of Employment and the Economy and the Social Insurance Institution of Finland. The service provides reliable person and company authentication and has been operational since January 2004. The service is a Liberty Interoperable tested and certified IdP Proxy for authentication services provided by Finnish banks and Electronic ID cards provided by Finnish Population Register Centre (PRC).

Tunnistus.fi provides two authentication mechanisms for the citizen user to choose: authentication with the Finnish Citizen's Electronic ID Card (HST-card) and authentication at the eService of the user's bank. For corporate, government and organizational users tunnistus.fi, through the KATSO-service, also provides Username-Password and One-Time Password authentication mechanisms. The authentication service supports SAML 2.0 and acts as an IDP for the SAML-based public sector eServices. It readily provides a large number of SAML SPs for service integration and deployment.

VETUMA citizen authentication service

VETUMA is an authentication service that can be used by Finnish public sector eServices to authenticate citizens. It provides two authentication mechanisms for the user to choose: authentication with the Finnish Citizen's Electronic ID Card (HST-card) ands authentication at the eService of the user's bank. Both of these meet the criteria of strong authentication as defined in the Act on Strong Electronic Identification and Electronic Signatures (617/2009). VETUMA supports SAML 2.0 acting as an IDP for the SAML-based public sector eServices. VETUMA is provided by the Government IT Shared Service Centre (VIP) in the State Treasury of Finland.

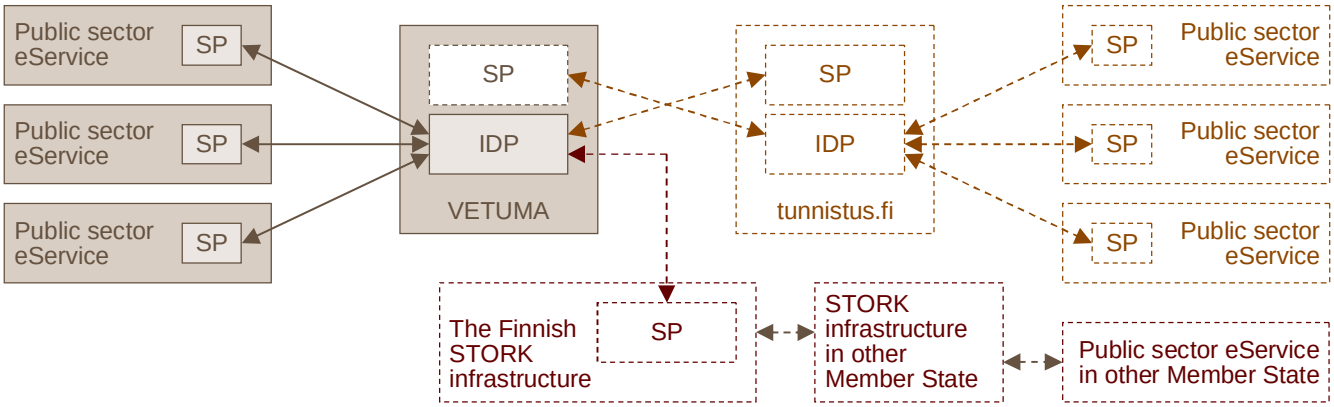

Figure 1. VETUMA – a Finnish public sector IDP

VETUMA will also be used in the pan-European STORK initiative (Secure Identity ***Across*** Borders Linked). Its role there is to authenticate Finnish citizens for public sector eServices in other Member States. Moreover, VETUMA will provide single sign-on with another public sector authentication service, namely tunnistus.fi.

Virtu federation for authentication and authorisation of civil servants

Virtu federation is the identity federation for authentication and authorization of civil servants in the Finnish government services. Virtu federation is a service coordinated by the State Treasury of Finland.

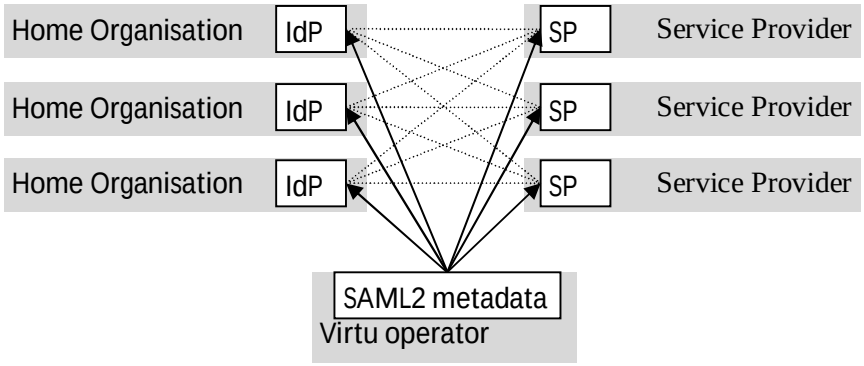

Figure 2. The technical architecture of Virtu federation consists of SAML 2.0 Identity Providers (IdP) operated by civil servants' home organisations, SAML 2.0 Service Providers (SP) operated by organizations providing services to the civil servants and the SAML 2.0 metadata managed by the federation operator.

The technical architecture (Figure 1) of Virtu federation is based on a full mesh of Identity Providers (IdP) and Service Providers (SP) who exchange SAML 2.0 assertions directly. As a subcontractor of the State Treasury, the operator of Virtu federation manages and distributes the federation's SAML 2.0 metadata containing the Providers registered to the federation.