e-GOVERNMENT BUND-LÄNDER-GEMEINDEN
http://reference.e-government.gv.at

Portalverbundprotokoll Version 2 **S-Profil**	**Konvention** **PVP2-S-Profil** **2.1.3** **20.10.2017** **Ergebnis der AG**

Kurzbeschreibung	Das S-Profil von PVP2 verwendet SAML WebSSO für die Authentifizierung von Benutzern mit Webbrowser. Dadurch wird die direkte Kommunikation des Browsers mit der Anwendung ermöglicht, was in Anwendungsfällen notwendig ist, wo Anwendungen nicht kompatibel mit dem Reverse-Proxy-Verfahren sind, datenschutzrechtliche Probleme bestehen oder SAML WebSSO als Industriestandard unterstützt werden soll. Das S-Profil spezifiziert eine Untermenge von SAML für das Deployment im Verwaltungsportalverbund oder kompatiblen Verbünden.	
Autor(en):	Rainer Hörbe (Wien)	Projektteam / Arbeitsgruppe **AG Integration und Zugänge (AG-IZ)** AG-Leitung: Ing. Dipl.-Ing.(FH) Hannes Wittmann, MSc (Mag. Wien) Stellvertretung: Dipl.-Ing. Dominik Klauser, BSc (BKA)
Beiträge von:	Peter Pfläging. Bernd Zwattendorfer, Peter Pichler	

Version 2.1.3 :	**Angenommen: -**
Version 2.1.2 : 1.6.2015	**Angenommen: -**
Version 2.1.1 : 26.2.2015	**Angenommen: -**
Version 2.1.0 : 3.9.2013	**Angenommen: 21.11.2013 VSt-1712/488**
Version 2.0.0 : 31.8.2011	**Angenommen: 14.10.2011 VST-1712/455**

Inhaltsverzeichnis

1 Einführung

1.1 Die SAML Spezifikation und ihre Profile

Die Kernspezifikation von SAML v2.0 ist eine umfangreiche und erweiterbare Norm, für die Profile der Datenstrukturen und Protokolle erstellt werden müssen, um überhaupt verwendet werden zu können [SAML-profiles-2.0]. Beispiele für diese Profile sind WebSSO, IdP Discovery, Assertion Query und die Attributprofile.

Diese Profile, die für bestimmte Szenarien gemacht werden, bleiben in der Regel ziemlich allgemein und umfassen eine Reihe von Optionen und Funktionen, die bei der Implementierung aufwändig sind und beim Einsatz diffizile Entscheidungen verlangen.

Für SAML gibt es 2 wesentliche Konformitätsprofile:

- *SAML Conformance 2.0* datiert von 2005 und definiert die Profile „IdP", „IdP light", „SP", „SP light" etc. Es gibt keine Interoperabilitätstest zu dieser Spezifikation.
- *Kantara eGoverntment Profile 2.0* [SAMLeGov2.0] von 2010, wird weiterentwickelt und für Interoperabilitätstests eingesetzt. Es verzichtet auch auf selten verwendete Anwendungsfälle wie Enhanced Client Proxy (ECP).

Weil das Kantara eGov Profile aktueller ist und weitere abgeleitete Spezifikationen zur Verfügung stehen, wird es als Grundlage für PVP herangezogen, wobei die Spezifikation in Implementierungs- und Deployment-Profile geteilt wird.

Das *Implementierungsprofil* [SAMLeGov2.0] ist eine Spezifikation für die Konformität von Produkten für Identity- und Service-Provider[1]. Es richtet sich vor allem an die Entwickler der Produkte und beinhaltet Einschränkungen und Ergänzungen der Funktionalität, Elemente und Attribute von [SAML2 *].

Ein *Deploymentprofil* definiert die Anforderungen an Software-Implementierungen die für den Einsatz in einem bestimmten Projekt, Verbund oder einer Inter-Föderation bestimmt sind. Daraus sollen Produktkonfigurationen und Testkriterien abgeleitet werden können.

Das PVP2 S-Profil ist ein Deploymentprofil des eGovernment Implementation Profils der Kantara Initiative. Vergleichbare Profile sind [SAMLint] , [ICAM-WebSSO] (USA) sowie die entsprechenden Profile von Dänemark, Finnland, Kanada und Neuseeland.

[1] In der SAML-Terminologie umfasst der IdP auch den Attribut-Provider

1.2 Anwendungsfall

Das S-Profil dient dem Anwendungsfall bei dem ein Web-Browser ohne Reverse Proxy mit der geschützten Anwendung kommuniziert und das SAML2-Protokoll nutzt.

Der Gültigkeitsbereich ist der Verwaltungsportalverbund und zukünftige kompatible Verbünde. Die Interoperabilität mit eIDAS wurde berücksichtigt.

Kompatibilität zu PVP R-Protokoll muss (über Gateways) möglich sein.

Die Struktur von Attributen (PVP eGovToken bzw. PVP Citizen Token) wird im Dokument „PVP2-eGovernment Attribute Profile“ [PVP2AttrProfile] festgelegt.

1.3 Kompatibilität zu R-Profil (1.x, 2.x) Anwendungen

Die Kompatibilität zum R-Profil muss (über Gateways) soweit als möglich unterstützt werden.

1.4 Struktur

Dieses Dokument baut auf dem eGovernment SAML V2.0 Implementation Profile [SAMLeGov2.0] auf. Dort werden folgende Basisdokumente der OASIS SAML V2.0 Spezifikation verwendet:

- Core
- Metadata
- Metadata Interoperability Profile
- Profiles (Web-SSO Profile, Single Logout Profile)
- IdP Discovery
- Binding (HTTP-Redirect Binding beim SAML AuthN-Request, HTTP-POST und HTTP-Artifact Bindings beim SAML Response)
- Artifact Resolution Profile
- Holder-of-Key Web Browser SSO Profile
- XML Signature (signature and digest algorithms)
- XML Encryption (block encryption, key transport & agreement algorithms)

1.5 Konformität

Dieses Deployment-Profil basiert auf dem eGov 2.0 Profil [SAMLeGov2.0] und wird gemäß Anforderungen in bestimmten Bereichen erweitert bzw. eingeschränkt (z.B. für eIDAS). Die normativen Anforderungen in Bezug auf die entsprechenden Abschnitte des eGov 2.0 Profils sind im Abschnitt 2 dieses Dokuments angegeben.
HINWEIS: Tests der Interoperabilität durch externe Stellen, wie sie etwa die Kantara Initiative durchgeführt, können zur Bestätigung der Konformität von Produkten mit dem eGov 2.0 Profil helfen.

Die Aktualisierung der SAML Spezifikation erfolgt über das umfangreiche Dokument [PVP2AttrProfile] PVP2 e-Government Attribute Profile

[SAML2Errata] welches bei wesentlichen Fragen unbedingt zu konsultieren ist.

1.6 Unterstützung der Sicherheitsklassen

Unterstützung für Sicherheitsklassen, wie sie in [SAML2IAProf] angegeben sind. Daraus folgen auch die Anforderungen an Metadaten zur Unterstützung für Sicherheitsklassen.

1.7 Referenzen

[SAMLeGov2.0] eGovernment Implementation Profile of SAML V2.0 (2010)
http://kantarainitiative.org/confluence/download/attachments/38929505/kantara-report-egov-saml2-profile-2.0.pdf

[SAMLint] Interoperable SAML 2.0 Web Browser SSO Deployment Profile
http://saml2int.org/profile

[ICAM-WebSSO] ICAM SAML 2.0 Web Browser Single Sign-on (SSO) Profile
http://www.idmanagement.gov/sites/default/files/documents/SAML20_Web_SSO_Profile.pdf

[PVP-SMA] Portalverbund Sicherheitsmaßnahmen (Algorithmen)

http://reference.e-government.gv.at

[SAML2IAProf] OASIS Committee Specification 01, SAML V2.0 Identity Assurance Profiles Version 1.0, November 2010.
http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-assurance-profile-cs-01.pdf

[SAML2 *] Die SAML-Spezifikation der OASIS umfasst eine Reihe von Dokumenten, die unter http://www.oasis-open.org/specs/#samlv2.0 abgerufen werden können.

[PVP2AttrProfile] PVP2 e-Government Attribute Profile

[SAML2Errata] Freigegebene Version des SAML 2.0 Errata Dokuments.
http://docs.oasis-open.org/security/saml/v2.0/sstc-saml-approved-errata-2.0.pdf
Im gleichen Verzeichnis findet man auch den aktuellen CD (committee draft) des Dokuments.

[SAML2SP-ReqInt] Service Provider Request Initiation Protocol and Profile Version 1.0
http://docs.oasis-open.org/security/saml/Post2.0/sstc-request-initiation.pdf

[IDP-DISCO-SPaP] Identity Provider Discovery Service Protocol and Profile

https://www.oasis-open.org/standards#saml-ipd

[eIDAS-MsgFormat] eIDAS SAML Message Format Version 1.0
https://joinup.ec.europa.eu/sites/default/files/eidas_message_format_v1.0.pdf

1.8 Verwendete Namespace Präfixe

Elemente der SAML 2.0 Core Spezifikation SAML2Core werden wie folgt referenziert:

- `<saml2p:Protocolelement>` - SAML 2.0 Protocol namespace [SAML2 *].
- `<saml2:Assertionelement>` - SAML 2.0 Assertion namespace [SAML2 *].

Elemente der SAML 2.0 Metadata Spezifikation SAML2Meta [SAML2 *]:

- `<md:Metadataelement>`

Elemente des Identity Provider Discovery Service Protocol und Profile [IDP-DISCO-SPaP]

- `<idpdisc:DiscoveryResponse>`

1.9 Änderungshistorie

Version 2.1.3

- STORK Vorgaben wurden durch die entsprechenden eIDAS Vorgaben ersetzt

- Editorale Änderungen (Referenzen vereinheitlicht, Neue Überschrift „Verwendete Namespace Präfixe")

Version 2.1.2:

- Synchronisierung der Versionsnummer mit dem Dokumentenset PVP 2.1.2
- Keine Änderung des Inhalts.

2 Deployment Requirements (normative)

Um die Vergleichbarkeit mit internationalen Normen und Profilen zu erhalten, bleibt der folgende Teil in Englisch.

2.1 Deployment Profile of the eGov 2.0 Profile

This specification is derived from the SAML 2 specifications [SAML2 *] and the Kantara Initiative SAML2 eGovernment Implementation Profile Version 2.0 [SAMLeGov2.0].

This deployment profile requires, unless otherwise specified, the conformance to [SAMLeGov2.0]. Unless this document specifies particular properties of SAML2, OASIS SAML 2.0 standards apply.

The following table lists the requirements of [SAMLeGov2.0] sections 2 and 3, which are classified as "*support*", "*restriction*" or *not applicable*, and "*extension*" for items not mentioned in [SAMLeGov2.0].

148

eGov 2.0 Implementation Profile	PVP 2.0 Implementation	PVP 2.0 Deployment Details
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.	Restriction	Identity Providers and Service Providers MUST provide a SAML 2.0 Metadata document representing its entity. Note: If products do not support metadata publication and consumption with appropriate signing and verification, import/export scripts need to be implemented to conform to the standard.
2.2.1 Metadata Profiles		
Implementations MUST support the SAML V2.0 Metadata Interoperability Profile Version 1.0 [MetaIOP].	Support	NOTE: This does not require native product support in implementations. It can be implemented in an auxiliary system.
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 OPTIONAL.	Restriction	Certificates in metadata must be valid X.509 certificates[2]. Different keys for signing and encryption SHOULD be used.

[2] While implementations with native support for [MetaIOP] may use only the public key contained in the <ds:X509Certificate> element, other implementations need to export them to valid PKIX certificates, e.g. with respect to time constraints and path validation.

eGov 2.0 Implementation Profile	PVP 2.0 Implementation	PVP 2.0 Deployment Details
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].	Restriction	The authoritative references for certificates used for TLS, signing, and encryption are the `<KeyDescriptor>` elements in metadata, with public keys certified by metadata signing keys. In addition, implementations MAY validate PKIX-type certificate paths [RFC5280]. In this case both trust chains (metadata and the embedded KeyDescriptor's PKIX) MUST be valid.
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.	Support	
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.	Restriction	Use of certificate attributes is out of scope of this specification[3].
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.	n/a	
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.	Support	

[3] Rationale: X.509 Attributes are not honored in a consistent way across implementations

eGov 2.0 Implementation Profile	PVP 2.0 Implementation	PVP 2.0 Deployment Details
	Extension	Deployments SHOULD be able to consume metadata from more than one location. This allows the participation in multiple federations, like intra- and extranet federations with a single provider.
	Extension	Implementations MUST specify at least one `<SingleLogoutService>` in their metadata.
	Extension	The default AssertionConsumerServiceIndex and the default AttributeConsumingServiceIndex SHOULD be indicated by the attribute `isDefault="true"` within SAML Metadata.
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 [SAML2 Meta] (under the assumption that entityID values used are suitable for such support).	Restriction	The `EntityID` MUST be a URL that is in the control of the provider. Providers SHOULD NOT use `EntityIDs` longer than 80 characters.

eGov 2.0 Implementation Profile	PVP 2.0 Implementation	PVP 2.0 Deployment Details
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 Support.	Restriction	The root element MUST be `<EntitiesDescriptor> or <EntityDescriptor>. <EntitiesDescriptor>` below the root element in metadata SHOULD be supported by metadata consumers[4].
Finally, implementations SHOULD allow for the automated updating/reimportation of metadata without service degradation or interruption.	Restriction	Finally, deployments MUST allow for the automated updating/reimportation of metadata. This SHOULD be performed without service degradation or interruption[5].

[4] Rationale: Consolidation of metadata in inter-federations.

[5] Rationale: Unless metadata can be refreshed in frequent intervals metadata cannot be relied upon. If deployments lack native support then semi-automated solutions (operator notification on manual execution of scripts and tools) are conformant if organizational procedures vouch for timely update of metadata.

149

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 PKIX [RFC5280].	Restriction	The root element MUST be signed, individual `<EntityDescriptor>` elements may be signed. Metadata consumers MUST perform a successful path validation of any signed metadata element before it is used.[6] The root element MUST contain the attribute `validUntil`. A metadata consumer MUST honor this attribute[7]. Frequent publication and consumption of metadata serves a similar purpose to that of certificate revocation lists and should be treated with equal importance. Consumers of metadata MUST use a specific certificate provided by the federation operator to verify the metadata root element. The certificate's revocation status MUST be checked according to the certificate policy.
	Extension	The `<AttributeProfile>` element MAY be specified in applicable SAML metadata instances[8].
2.3 Name Identifiers		

[6] Reason: As critical data metadata must be protected by digital signatures.

[7] Reason: CacheDuration has been found problematic in saml2int deployments and should be ignored.

[8] For the purpose of documentation; current products do not use that information.

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	Restriction	In addition the following Name Identifier format MUST be supported: • urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified[9]
Support for other formats is OPTIONAL.	Support	
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].	Extension	The support for SAML Attribute profiles is defined in other documents and may require additional attribute name formats.[10]
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.	Support	

[9] For compatibility with systems that use attributes instead of NameID semantics

[10] For eGovernment use cases refer to [PVP2AttrProfile]

	Extension	Mandatory attributes MUST not contain empty values. Attribute assertions SHOULD NOT contain empty optional attributes.
2.5 Browser Single Sign-On		
This section defines an implementation profile of the SAML V2.0 Web Browser SSO Profile [SAML2Prof].	Support	
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].	Extension	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. The support of Identity Provider Discovery is needed if more than one IDP is involved.
2.5.2 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.	Extension	In addition implementations MUST support the use of the HTTP-Post Binding [SAML2Bind] for the transmission of `<saml2p:AuthnRequest>` messages, including the generation or verification of signatures in conjunction with this binding under following conditions:[11] a) In the role of an IdP: Consumption of `<saml2p:AuthnRequest>` b) In the role of an SP: Generation of a `<saml2p:AuthnRequest>` if requested attributes are not communicated via metadata.
Support for other bindings is OPTIONAL.	Support	
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):	Support	

[11] Rationale for POST binding: Use if requests are large, e.g. because key material is passed in the request as URL parameter instead with metadata. Otherwise Redirect binding is preferred because of better performance.

AssertionConsumerServiceURL	Restriction	SPs SHOULD not provide AssertionConsumerServiceURL. The IdP MUST obtain this from the metadata. If included, the IdP MUST compare the value with metadata and issue an error if the value does not match. Comparison is a case-sensitive exact string match. Providers should not rely on any form of canonicalization.
ProtocolBinding	Restriction	SPs SHOULD not use ProtocolBinding so that the IdP MAY determine the binding from SP-Metadata.
ForceAuthn	Support	Note: If true, the IdP MUST force re-authentication.
IsPassive	Support	Note: If true, the IdP MUST NOT take control over the browser.
AttributeConsumingServiceIndex	Restriction	AttributeConsumingServiceIndex MAY be included in <samlp:AuthnRequest>. The SP requests this way a set of attributes when the choice of several sets is given.
<saml2p:RequestedAuthnContext>	Support	
<saml2p:NameIDPolicy>	Support	<samlp:NameIDPolicy> SHOULD exist.
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 [SAML2Core].	Extension	<saml2p:AuthnRequest> messages MUST be signed[12].

[12] Rationale: Signatures provide stronger authentication than DNS-based means (URL-lookup in metadata)

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.	Restriction	MUST support Levels of Assurance (LoA) as specified in [SAML2 Assur]. SPs SHOULD request a specific LoA that is defined as a URI. Implementations requesting a LoA MUST limit the value of the Comparison attribute of `<saml2p:RequestedAuthnContext>` to `"minimum"` in case of eIDAS-LoA and to `"exact"` for SecClass. IdPs that are not configured to support the minimum LoA requested MUST reject the authentication request with an appropriate status code. The following URIs are valid: SecClass: http://www.ref.gv.at/ns/names/agiz/pvp/secclass/0 http://www.ref.gv.at/ns/names/agiz/pvp/secclass/0-1 http://www.ref.gv.at/ns/names/agiz/pvp/secclass/0-2 http://www.ref.gv.at/ns/names/agiz/pvp/secclass/0-3 eIDAS Levels of Assurance[13]: http://eidas.europa.eu/LoA/low http://eidas.europa.eu/LoA/substantial http://eidas.europa.eu/LoA/high

[13] Proposed URIs to be approved and published by eIDAS COMMISSION IMPLEMENTING REGULATION (EU) 2015/1502 (http://eur-lex.europa.eu/legal-content/EN/TXT/?uri=OJ:JOL_2015_235_R_0002) and the technical specification.

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 alternative verification mechanisms.	Support	
	Extension	If a use case requires dynamic attribute requests, the SP MUST support dynamic attribute requests within `<saml2p:AuthnRequest>`. This should be rare, because the need for different attribute sets can be communicated using the `<AttributeConsumingServiceIndex>` element. Static attribute requests MUST be communicated in Metadata.
2.5.3 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.	Restriction	Identity Provider and Service Provider implementations MUST support the use of the HTTP-POST binding [SAML2Bind] for the transmission of `<saml2p:Response>` messages. The use of other bindings [SAML2Bind] is optional.
Support for other bindings, and for artifact types other than urn:oasis:names:tc:SAML:2.0:artifact-04, is OPTIONAL.	Support	
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).	Restriction	Unsolicited response messages MUST not be accepted. Instead implementations MUST rely on the SAML request initiation protocol [SAML2SP-ReqInt].

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].	Support	
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.	Extension	Identity Provider and Service Provider deployments MUST sign the `<saml2:Response>` element; Signing of the `<saml2:Assertion>` element is OPTIONAL[14].

[14] Signing the assertion should be limited to cases where the assertion will be further processed separated from the response, e.g. for archiving.

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.	Support	Identity Provider and Service Provider implementations MUST support the use of XML Encryption via the `<saml2:EncryptedAssertion>` element; support for the `<saml2:EncryptedID>` and `<saml2:EncryptedAttribute>` elements is OPTIONAL
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 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.	Restriction	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 one `<saml2:AttributeStatement>` element.
Identity Provider implementations MUST support the inclusion of a Consent attribute in <saml2p:Response> messages, and a SessionIndex attribute in <saml2:AuthnStatement> elements.	Support	Note: Current Austrian e-Government use cases do not require the Consent attribute.
Service Provider implementations that provide some form of session semantics MUST support the <saml2:AuthnStatement> element's SessionNotOnOrAfter attribute.	Support	

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.	Restriction	[IAP] is not relevant in the context of PVP.
	Extension	The `<saml2:Subject>` element of the assertions issued by an Identity Provider MUST contain a `<saml2:NameID>` element.
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 by the following subsections.	Restriction	Deployments MAY support the HTTP-Artifact Binding. For use cases with high assurance requirements or sufficient client bandwidth the use of this binding is not recommended.
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.	Support	
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.	Support	

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.	Support	
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.	Support	
2.6 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].	Restriction	IdPs and SPs SHOULD support [HoKSSO] for SecClass 3. This requirement does not apply to IdPs serving clients that use unmanaged devices.
The implementation requirements defined in section 2.5 for the non-holder-of-key profile apply to implementations of this profile.	Support	
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.	Support	

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 Support.	Support	
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.	Support	
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.	Support	
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.	Support	
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.	Support	
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.	Support	
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.	Restriction	Identity Provider implementations MUST support the HTTP-Redirect binding [SAML2Bind] for both issuance and reception of `<saml2p:LogoutRequest>` messages. The support of the SAML SOAP (using HTTP as a transport) binding [SAML2Bind] is optional.
Service Provider implementations MUST support the SAML SOAP (using HTTP as a transport) binding [SAML2Bind] for both issuance and reception of <saml2p:LogoutRequest> messages.	Restriction	Service Provider implementations MUST support the HTTP-Redirect binding [SAML2Bind] for both issuance and reception of `<saml2p:LogoutRequest>` messages. The support of the SAML SOAP (using HTTP as a transport) binding [SAML2Bind] is optional.
Support for other bindings is OPTIONAL.	Support	
Implementations MUST support the use of SAML message signatures and TLS server authentication to authenticate <saml2p:LogoutRequest> messages; support for TLS client authentication, or other forms of authentication in conjunction with the SAML SOAP binding, is OPTIONAL.	Support	

Identity Provider and Service Provider implementations MUST support the use of XML Encryption via the <saml2:EncryptedID> element when using the HTTP-Redirect binding.	Extension	Identity Provider and Service Provider implementations SHOULD support the use of XML Encryption via the `<saml2:EncryptedID>`.
		If the IdP has provided a `SessionIndex` attribute in the `<saml2:Assertion>`, the SP MUST include the `<saml2p:SessionIndex>` element in the SAML logout request.
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.	Restriction	Logout MUST never be local-only[15] (or partially) and always implies global Single Logout at all authenticated service providers.
Service Provider implementations MUST support both user-initiated termination of the local session only and user-initiated Single Logout.	Restriction	User-initiated termination of the local session will not be supported.
Identity Provider implementations MUST also support the administrative initiation of Single Logout for any active session, subject to appropriate policy.	Restriction	Administrative logout MAY be supported.
	Extension	The Single Logout process result MUST be displayed to the user by the SLO initiator (Service Provider or Identity Provider).

[15] Rationale: Users may feel secure, but leave other session unprotected.

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.	Restriction	Identity Provider implementations MUST support the HTTP-Redirect binding [SAML2Bind] for both issuance and reception of `<saml2p:LogoutResponse>` messages. The support of the SAML SOAP (using HTTP as a transport) binding [SAML2Bind] is optional.
Service Provider implementations MUST support the SAML SOAP (using HTTP as a transport) binding [SAML2 Bind] for both issuance and reception of <saml2p:LogoutResponse> messages.	Restriction	Service Provider implementations MUST support the HTTP-Redirect binding [SAML2Bind] for both issuance and reception of `<saml2p:LogoutResponse>` messages. The support of the SAML SOAP (using HTTP as a transport) binding [SAML2Bind] is optional.
Support for other bindings is OPTIONAL.	Support	
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.	Support	
2.9 Single Session Timeout		
	Extension	SP and IDP SHOULD support the synchronization of timeouts of a user's SP session so that the activity at one SP will restart the timeout at the IDP and prevent logout at other SPs[16].

[16] Remark: While a basic solution would just send an AuthnRequest/isPassive=true every other few minutes, this is not fully compatible with a broad range of products.

3 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.	Support	
3.1.1 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])		For the support and use of signature and encryption algorithms see [PVP-SMA].
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/xmlenc#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.	Support	
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.	Support	
End of table		