



Portalverbundprotokoll Version 2 S-Profil		Konvention
		PVP2-S-Profil 2.0.0.a
		Empfehlung
Kurzbeschreibung	<p>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.</p> <p>Das S-Profil spezifiziert eine Untermenge von SAML für das Deployment im Verwaltungsportalverbund oder kompatiblen Verbänden.</p>	
Autor(en):	Rainer Hörbe (Wien)	Projektteam / Arbeitsgruppe AG Integration und Zugänge (AG-IZ) AG-Leiter: Peter Pfläging (Wien) Stellvertreter: D.I. Peter Reichstädter (BKA)
Beiträge von:	Peter Pfläging, Bernd Zwattendorfer, Peter Pichler	

Version 2.0.0.a: 28.8.2011

Angenommen: 14.10.2011

VST-1712/455

Inhaltsverzeichnis

1 Einführung	3
1.1 Die SAML Spezifikation und ihre Profile	3
1.2 Anwendungsfall	3
1.3 Struktur.....	4
1.4 Konformität	4
1.5 Unterstützung der Sicherheitsklassen.....	4
1.6 Referenzen	4
2 Deployment Requirements (normative)	6
2.1 Profile of the eGov 2.0 Profile	6

1 Einführung

1.1 Die SAML Spezifikation und ihre Profile

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 Web Browser SSO, 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 eGov Profile 2.0* datiert von 2009, 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-Profilen geteilt wird.

Das *Implementierungsprofil* [SAMLLeGov2.0] ist eine Spezifikation für die Konformität von Produkten für Identity- und Service- Provider¹. 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.

Vergleichbare Profile sind [SAMLint] , [ICAM-WebSSO] (USA) sowie die entsprechenden Profile von Dänemark, Finnland, Kanada und Neuseeland.

Das STORK-Protokoll [STORK] basiert ebenfalls auf den Spezifikationen von SAML 2.0 [SAML2 *], definiert im Gegensatz zu PVP aber ein eigenes Profil, das derzeit von den am Markt verfügbaren Produkten noch nicht unterstützt wird. Im Sinne einer zukünftigen Konvergenz wird deshalb auch die STORK Interface Spezifikation im vorliegenden PVP2 S-Profil berücksichtigt. Die Kompatibilität auf Schnittstellenebene ist jedoch nur gegeben wenn eine Gateway-Komponente (Proxy) eingesetzt wird, da einige elementare Protokolleigenschaften unterschiedlich spezifiziert sind.

1.2 Anwendungsfall

Das S-Protokoll dient dem Anwendungsfall wo ein Web-Browser ohne Reverse Proxy mit der Anwendung kommuniziert und das SAML2-Protokoll nutzt. Der Gültigkeitsbereich ist der Verwaltungsportalverbund und zukünftige kompatible Verbünde.

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

Die Struktur von Attributen (PVP Token) wird in einem anderen Dokument der PVP2-Spezifikation festgelegt. Davon ausgenommen ist das Attribut „SecClass“, das im Deployment Profile festgelegt wird; siehe auch 1.5.

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

1.3 Struktur

Dieses Dokument baut auf dem Kantara eGovernment SAML V2.0 Implementation Profile 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 AuthN-request, HTTP-POST und HTTP-Artifact bindings beim 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)

Die Aktualisierung der SAML Spezifikation erfolgt über das umfangreiche Dokument [\[SAML2Errata\]](#) welches bei wesentlichen Fragen unbedingt zu konsultieren ist.

1.4 Konformität

Dieses Deployment-Profil basiert auf eGov 2.0 Profil [\[SAMLeGov2.0\]](#) und wird gemäß Anforderungen in bestimmten Bereichen erweitert bzw. eingeschränkt (z.B. für STORK). 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.

1.5 Unterstützung der Sicherheitsklassen

Unterstützung für Sicherheitsklassen, wie sie in [\[SAML2IAPProf\]](#) angegeben sind. Daraus folgen auch die Anforderungen an Metadaten zur Unterstützung für Sicherheitsklassen.

1.6 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/documents/SAML20_Web_SSO_Profile.pdf

[\[SAML2IAPProf\]](#) 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.

[\[SAML2Errata\]](#) 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.

[STORK] D5.8.3b Interface Specification

URL: TO BE PUBLISHED

Elemente der SAML 2.0 core Spezifikation [SAML2Core](#) werden wie folgt referenziert:

- `<saml2p:Protoelement>` - SAML 2.0 Protocol namespace.
- `<saml2:Asserionelement>` - SAML 2.0 Assertion namespace.

Elemente der SAML 2.0 Metadata Spezifikation [SAML2Meta](#):

- `<md:Metadataelement>`

Identity Provider Discovery Service Protocol und Profil [IdPDisco](#):

- `<idpdisc:DiscoveryResponse>`

2 Deployment Requirements (normative)

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

2.1 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]. The STORK interface specification [STORK] is regarded so far it does not introduce significant incompatibilities with available products.

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 supported, restricted or not applicable, and extended for items not mentioned in [SAMLeGov2.0].

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.	Restricted	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].	Restricted	Implementations MUST support the SAML V2.0 Metadata Interoperability Profile Version 1.0 [MetaIOP]. However, the path validation of certificates used with TLS and signatures MAY use PKIX. Verification using metadata keys is optional.
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.	Restricted	

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].	Restricted	Implementations MUST support PKIX [RFC5280]. During a SAML protocol exchange, the relying party MUST the key's validity using either metadata or PKIX.
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.	Restricted	OCSP is preferred and CRLs should be avoided. Note: Whitelisting of certificates is a recommended practice to replace OCSP, but not available with most products.
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.	Restricted	Certificate attributes beyond name and identifier SHOULD not be used ² .
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	

² 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
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	
	Extension	Implementations 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.
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).		
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.		

eGov 2.0 Implementation Profile	PVP 2.0 Implementation	PVP 2.0 Deployment Details
Importation of multiple entities' metadata contained within an <md:EntitiesDescriptor> element Support.		The root element MUST be <EntitiesDescriptor>. <EntitiesDescriptor> below the root element in metadata SHOULD be supported by metadata consumers ³ .
Finally, implementations SHOULD allow for the automated updating/reimportation of metadata without service degradation or interruption.	Restricted	Finally, implementations MUST allow for the automated updating/reimportation of metadata without service degradation or interruption ⁴ .

³ Rationale: Consolidation of metadata in inter-federations.

⁴ Reason: Unless metadata can be refreshed in frequent intervals metadata cannot be relied upon.

2.2.2.1 Metadata Verification		
<p>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:</p> <ul style="list-style-type: none"> • 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]. 	Restricted	<p>The root element MUST be signed, individual <code><EntityDescriptor></code> elements may be signed. Metadata consumers MUST perform a successful path validation of any signed metadata element before it is used.⁵</p> <p>The root element MUST contain the attributes <code>validUntil</code> AND <code>cacheDuration</code>. A metadata consumer MUST honor both attributes⁶. Frequent publication and consumption of metadata serves a similar purpose to that of certificate revocation lists and should be treated with equal importance.</p> <p>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.</p>
	Extension	<p>The <code><AttributeProfile></code> element MAY be specified in applicable SAML metadata instances⁷.</p>
2.3 Name Identifiers		

⁵ Reason: As critical data metadata must be protected by digital signatures.

⁶ Reason: Technically enforceable rules shall be specified in a structure way.

⁷ For the purpose of documentation; current products do not use that information.

<p>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]:</p> <p>urn:oasis:names:tc:SAML:2.0:nameid-format:persistent urn:oasis:names:tc:SAML:2.0:nameid-format:transient</p>	Restricted	<p>An equivalent of „transient“ is not supported in PVP 1.X and should not be used unless backwards compatibility is not an issue. In addition the following Name Identifiers MUST be supported:</p> <ul style="list-style-type: none"> urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified
Support for other formats is OPTIONAL.	Support	
2.4 Attributes		
<p>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].</p>	Restricted	The Attribute Assertion MUST support the [PVP 2.x eGov Token Profile] for eGovernment use cases.
	Extension	<p>The only attribute name formats to be used are:</p> <p>urn:oasis:names:tc:SAML:2.0:attrname-format:uri urn:oasis:names:tc:SAML:2.0:profiles:attribute:basic</p> <p>The SAML V2.0 X500-attribute name profile SHOULD be used.</p>

<p>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.</p>	<p>Support 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).</p> <p>XML values in <saml2:AttributeValue> elements MAY be supported⁸.</p>	
	Extension	Service Providers MUST be able to accept mandatory attributes with empty values.
2.5 Browser Single Sign-On		
<p>This section defines an implementation profile of the SAML V2.0 Web Browser SSO Profile [SAML2Prof].</p>	Support	
2.5.1 Identity Provider Discovery		
<p>Service Provider and Discovery Service implementations MUST support the Identity Provider Discovery Service Protocol Profile in conformance with section 2.4.1 of [IdPDisco].</p>	Extend	<p>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></p>
2.5.2 Authentication Requests		
2.5.2.1 Binding and Security Requirements		

⁸ used by STORK

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	In addition implementations MAY 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. ⁹
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	
AssertionConsumerServiceURL	Restricted	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	Support	
ForceAuthn	Support	Note: ForceAuthn MAY be used to require the IDP to force the end user to authenticate to the IDP regardless of the end user's authentication session status at the IDP.

⁹ Rationale: Use if requests are large, e.g. because key material is passed in the request instead with metadata.

IsPassive	Support	<p>Note: IsPassive MAY be used if the SP does not wish for the IDP to take direct control of the end user's browser (i.e., show the end user a page).</p> <p>If IsPassive is true, the end user MUST be able to authenticate in some passive manner, otherwise the resulting response MUST NOT contain an <Assertion>.</p> <p>This feature allows the SP to determine whether it should alert the end user that he or she is about to interact with the IDP. An example of a passive situation is: the SP discovers through the common domain cookie that the end user may have an active session at a particular IDP.</p>
AttributeConsumingServiceIndex	Restricted	<p>AttributeConsumingServiceIndex MAY be included in <samlp:AuthnRequest>. The SP requests this way a set of attributes.</p>
		<p>The following attribute SHOULD additionally be present:</p> <ul style="list-style-type: none"> • ProviderName
<saml2p:RequestedAuthnContext>	Support	
<saml2p:NameIDPolicy>	Restricted	<p><samlp:NameIDPolicy> MUST exist. It may have one of the following values:</p> <p>urn:oasis:names:tc:SAML:2.0:nameid-format:persistent urn:oasis:names:tc:SAML:2.0:nameid-format:transient urn:oasis:names:tc:SAML:2.0:nameid-format:unspecified</p>

<p>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].</p>	Support	
<p>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.</p>	Restricted	<p>MUST support Levels of Assurance as specified in [SAML2 Assur]. SPs MUST request a specific Level of Assurance that needs to match exactly. There is no ordering like with SecClass 0 – 3. Therefore SP needs to state all acceptable LoA (= SecClass). E.g. when a level 2 is required the SP needs to specify that it accepts level 3 as well.</p> <p>IDPs MUST provide the exact LoA requested and MUST reject any LoA request that is not defined as a SecClass. IDPs that are not configured to support the LoA requested MUST reject the authentication request with an appropriate status code.</p> <p>Remark: PVP 1.x relied on the scale of 0 – 3 with an implicit hierarchy, with any SecClass greater or equal the required one accepted. The [SAML2 Assur] profile negotiates from two lists of policies, requiring that the intersecting set is not empty.</p> <p>The URIs are: http://www.ref.gv.at/ns/names/agiz/pvp/secclass/0 http://www.ref.gv.at/ns/names/agiz/pvp/secclass/1 http://www.ref.gv.at/ns/names/agiz/pvp/secclass/2 http://www.ref.gv.at/ns/names/agiz/pvp/secclass/3</p>

<p>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.</p>	Support	
2.5.3 Responses		
2.5.3.1 Binding and Security Requirements		
<p>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.</p>		
<p>Support for other bindings, and for artifact types other than urn:oasis:names:tc:SAML:2.0:artifact-04, is OPTIONAL.</p>	Support	
<p>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).</p>	Support	Note: This is required for IdP-initiated Authentication

<p>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].</p>	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.	Support	
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	Note: As all responses are transmitted over encrypted channels ¹⁰ , Identity Providers are not required to encrypt assertions, responses or attributes. NameIDs SHOULD NOT be encrypted ¹¹ .
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.	Restrict	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.

¹⁰ Assuming HTTP-Redirect is not used. If it would be used, assertions would have to be encrypted to be protected from appearing in log files.

¹¹ Perceived use cases always include attributes that might identify the subject. Therefore NameID encryption is insufficient and the whole assertion needs to be protected.

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: Consent is not required for A2A and B2B use cases.
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.	Extend	To avoid man-in-the-middle attacks, a Service Provider that has used a RequestedAuthenticationContext in the AuthenticationRequest MUST verify that the AuthnContext of the Response satisfies its needs.
	Extend	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 <saml2:EncryptedID>.
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 restricted by the following subsections.	Support	

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].	Restricted	IdPs and SPs MUST support [HoKSSO] for SecClass 3 and above.
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		
<p>This section defines an implementation profile of the SAML V2.0 Single Logout Profile [SAML2Prof].</p> <p>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.</p>	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.	Support	

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	
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.	Support	
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.	Support	
Service Provider implementations MUST support both user-initiated termination of the local session only and user-initiated Single Logout.	Support	

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.	Support	
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.	Support	
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	
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		

<p>Implementations MUST support the signature and digest algorithms identified by the following URIs in conjunction with the creation and verification of XML Signatures [XMLSig]:</p> <p>http://www.w3.org/2001/04/xmldsig-more#rsa-sha256 (defined in [RFC4051])</p> <p>http://www.w3.org/2001/04/xmlenc#sha256 (defined in [XMLEnc])</p>	Support	
<p>Implementations SHOULD support the signature and digest algorithms identified by the following URIs in conjunction with the creation and verification of XML Signatures [XMLSig]:</p> <p>http://www.w3.org/2001/04/xmldsig-more#ecdsa-sha256 (defined in [RFC4051])</p>	Support	
<p>Implementations MUST support the block encryption algorithms identified by the following URIs in conjunction with the use of XML Encryption [XMLEnc]:</p> <p>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</p>	Support	
<p>Implementations MUST support the key transport algorithms identified by the following URIs in conjunction with the use of XML Encryption [XMLEnc]:</p> <p>http://www.w3.org/2001/04/xmlenc#rsa-1_5 http://www.w3.org/2001/04/xmlenc#rsa-oaep-mgf1p</p>	Support	

<p>Implementations SHOULD support the key agreement algorithms identified by the following URIs in conjunction with the use of XML Encryption [XMLEnc]:</p> <p>http://www.w3.org/2009/xmlenc11#ECDH-ES defined in [XMLEnc11])</p> <p>(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.)</p>		
<p>Support for other algorithms is OPTIONAL.</p>		
<p>3.2 Standard with Logout</p>		
<p>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.</p>	<p>Support</p>	
<p>3.3 Full</p>		
<p>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.</p>	<p>Support</p>	
<p>End of table</p>		