

---

# Simplified Web Browser Single Sign-On and Sign-Out Profile for SAML 2.0

Andreas Solberg <andreas.solberg@uninett.no>

Thu Apr 17 06:25:07 2008

## Table of Contents

Differences from the full version profiles .....	2
Differences from the Web Browser SSO Profile .....	2
Differences from the Logout Profile .....	2
Introduction .....	2
Endpoint descriptions .....	3
Single Sign-On step by step .....	3
HTTP Request to Service Provider .....	4
Service Provider Determines Identity Provider .....	4
<AuthnRequest> Is Issued by Service Provider to Identity Provider .....	4
Identity Provider Identifies Principal .....	5
Identity Provider Issues <Response> to Service Provider .....	5
Service Provider Grants or Denies Access to User Agent .....	5
Use of Authentication Request Protocol .....	6
<AuthnRequest> Usage .....	6
<Response> Usage .....	6
<Response> Message Processing Rules .....	7
POST-Specific Processing Rules .....	7
Unsolicited Responses .....	8
Single Log-Out step by step .....	8
<LogoutRequest> Issued by Session Participant to Identity Provider .....	9
Identity Provider Determines Session Participants .....	9
<LogoutRequest> Issued by Identity Provider to Session Participant/Authority .....	9
Session Participant/Authority Issues <LogoutResponse> to Identity Provider .....	9
Identity Provider Issues <LogoutResponse> to Session Participant .....	10
Use of the Single Logout Protocol .....	10
<LogoutRequest> Usage .....	10
<LogoutResponse> Usage .....	10
Simplifications .....	10
Requirements to NameID usage .....	11
Multiple simultaneous sessions .....	11
Use of Attribute Push .....	11
Use of Metadata .....	11
Robustness of HTTP-REDIRECT Single Log-Out .....	12

The Web Browser SSO Profile of SAML 2.0 as well as the Single Log-Out profile have several options for almost everything. Some of the functionality, that is optional to use, is complex to implement, and may seem unnecessary because they does not give new functionality, only alternative ways of achieving the same goals. Consequently it is tempting to implement a subset of the profile.

The *Simplified Web Browser Single Sign-On and Sign-Out Profile for SAML 2.0* is an attempt to standardize a minimalistic subset of the full Web Browser SSO Profile and the Logout-Profile, that will be sufficiently

Simplified Web Browser  
Single Sign-On and Sign-  
secure, and not lack any significant functionalities, fully compatible with the full version of the profile  
at the same time that is really easy to implement.

---

The *Simplified Web Browser Single Sign-On and Sign-Out Profile for SAML 2.0* includes both Single Sign-On and Sign-Out, which is different from the full version of the profiles, which is two separate profiles.

## Important

This document is word in progress, and currently in draft status.

# Differences from the full version profiles

This document contains a lot of copied text from the full version of the respective profiles. If you already know the full version of the profiles, it would be easier to understand the difference by looking at this list of things that differ from the full versions:

## Differences from the Web Browser SSO Profile

- The <AuthnRequest> can only be sent using the HTTP-REDIRECT binding.
- The <Response> can only be sent using the HTTP-POST binding.
- The <AuthnRequest> should not be signed.
- Special considerations of how to ensure security without signing the request. See the section called “<AuthnRequest> Is Issued by Service Provider to Identity Provider”.
- In addition several items listed in the section called “Simplifications”.

## Differences from the Logout Profile

- Only using HTTP-REDIRECT binding for both LogoutRequest and LogoutResponse.
- Support for SP to SP logout negotiation is not required to be supported.
- In addition several items listed in the section called “Simplifications”.

# Introduction

In the Single Sign-On scenario supported by the *Simplified Web Browser Single Sign-On and Sign-Out Profile for SAML 2.0*, a web user either accesses a resource at a service provider, or accesses an identity provider such that the service provider and desired resource are understood or implicit. The web user authenticates (or has already authenticated) to the identity provider, which then produces an authentication assertion (possibly with input from the service provider) and the service provider consumes the assertion to establish a security context for the web user. To implement this scenario, a profile of the SAML Authentication Request protocol is used, in conjunction with the HTTP Redirect and HTTP POST bindings. It is assumed that the user is using a standard browser and can authenticate to the identity provider by some means outside the scope of SAML.

Once a principal has authenticated to an identity provider, the authenticating entity may establish a session with the principal (typically by means of a cookie, URL re-writing, or some other implementation-specific

---

means). The identity provider may subsequently issue assertions to service providers or other relying parties, based on this authentication event; a relying party may use this to establish its own session with the principal.

In such a situation, the identity provider can act as a session authority and the relying parties as session participants. At some later time, the principal may wish to terminate his or her session either with an individual session participant, or with all session participants in a given session managed by the session authority. The former case is considered out of scope of this specification. The latter case, however, be satisfied using this profile of the SAML Single Logout protocol ([SAMLCore] Section 3.7).

Note that a principal (or an administrator terminating a principal's session) may terminate this "global" session by contacting the session authority.

The Logout protocol messages are send using the asynchronous HTTP Redirect binding.

## Endpoint descriptions

Here are the endpoints that are referred to in this profile:

Single Sign-On Service	This is the authentication request protocol endpoint at the identity provider to which the <AuthnRequest> message (or artifact representing it) is delivered by the user agent.
Assertion Consumer Service	This is the authentication request protocol endpoint at the service provider to which the <Response> message (or artifact representing it) is delivered by the user agent.
Single Logout Service	This is the single logout protocol endpoint at an identity provider or session participant to which the <LogoutRequest> or <LogoutResponse> messages (or an artifact representing them) delivered. The same or different endpoints MAY be used for requests and responses.

## Single Sign-On step by step

The following steps are described by the sign-on part of the profile.

### 1. HTTP Request to Service Provider

In step 1, the principal, via an HTTP User Agent, makes an HTTP request for a secured resource at the service provider without a security context.

### 2. Service Provider Determines Identity Provider

In step 2, the service provider obtains the location of an endpoint at an identity provider for the authentication request protocol that supports its preferred binding. The means by which this is accomplished is implementation-dependent. The service provider MAY use the SAML identity provider discovery profile described in ???.

### 3. <AuthnRequest> issued by Service Provider to Identity Provider

In step 3, the service provider issues an <AuthnRequest> message to be delivered by the user agent to the identity provider. The request is sent using the HTTP-REDIRECT binding.

#### 4. Identity Provider identifies Principal

In step 4, the principal is identified by the identity provider by some means outside the scope of this profile. This may require a new act of authentication, or it may reuse an existing authenticated session.

#### 5. Identity Provider issues <Response> to Service Provider

In step 5, the identity provider issues a <Response> message to be delivered by the user agent to the service provider. The HTTP POST binding must be used to transfer the message to the service provider through the user agent. The message may indicate an error, or will include (at least) an authentication assertion.

#### 6. Service Provider grants or denies access to Principal

In step 6, having received the response from the identity provider, the service provider can respond to the principal's user agent with its own error, or for the principal and return the requested resource.

### Note

IdP-first scenario

An identity provider can initiate this profile at step 5 and issue a message to a service provider without the preceding steps.

## HTTP Request to Service Provider

If the first access is to the service provider, an arbitrary request for a resource can initiate the profile. There are no restrictions on the form of the request. The service provider is free to use any means it wishes to associate the subsequent interactions with the original request. Each of the bindings provide a RelayState mechanism that the service provider MAY use to associate the profile exchange with the original request. The service provider SHOULD reveal as little of the request as possible in the RelayState value unless the use of the profile does not require such privacy measures.

## Service Provider Determines Identity Provider

This step is implementation-dependent. The service provider MAY use the SAML identity provider discovery profile, described in [SAMLProfiles].

### Note

Bridging

The service provider MAY also choose to redirect the user agent to another service that is able to determine an appropriate identity provider. In such a case, the service provider may issue an <AuthnRequest> (as in the next step) to this service to be relayed to the identity provider, or it may rely on the intermediary service to issue an <AuthnRequest> message on its behalf.

## <AuthnRequest> Is Issued by Service Provider to Identity Provider

Once an identity provider is selected, the location of its single sign-on service is determined, based on the SAML binding chosen by the service provider for sending the <AuthnRequest>. Metadata (as in

---

[SAMLMeta]) MAY be used for this purpose. In response to an HTTP request by the user agent, an HTTP response is returned containing an <AuthnRequest> message according to the HTTP-REDIRECT binding, to be delivered to the identity provider's single sign-on service.

It is RECOMMENDED that the HTTP exchanges in this step be made over either SSL 3.0 [SSL3] or TLS 1.0 [RFC2246] to maintain confidentiality and message integrity. The <AuthnRequest> message SHOULD NOT be signed. If the Service Provider requests a specific authncontext or requires ForceAuthN, instead the Service Provider must be informed and verify the actual authncontext and whether ForceAuthN was provider in the signed <Resonse> message.

The identity provider MUST process the <AuthnRequest> message as described in [SAMLCore]. This may constrain the subsequent interactions with the user agent, for example if the IsPassive attribute is included.

## Identity Provider Identifies Principal

At any time during the previous step or subsequent to it, the identity provider MUST establish the identity of the principal (unless it returns an error to the service provider). The ForceAuthn <AuthnRequest> attribute, if present with a value of true, obligates the identity provider to freshly establish this identity, rather than relying on an existing session it may have with the principal. Otherwise, and in all other respects, the identity provider may use any means to authenticate the user agent, subject to any requirements element.

## Identity Provider Issues <Response> to Service Provider

Regardless of the success or failure of the <AuthnRequest>, the identity provider SHOULD produce an HTTP response to the user agent containing a <Response> message, using the HTTP-POST binding, to be delivered to the service provider's assertion consumer service.

The location of the assertion consumer service MAY be determined using metadata (as in [SAMLMeta]). The identity provider MUST have some means to establish that this location is in fact controlled by the service provider. A service provider MAY indicate the specific assertion consumer service to use in its <AuthnRequest> and the identity provider MUST honor them if it can.

### Important

The Identity Provider MUST verify that the assertion consumer service is a legal trusted endpoint for the specific service provider. This can be done by in example cross-checking the values in the <AuthnRequest> with the metadata.

It is REQUIRED that the HTTP requests in this step be made over either SSL 3.0 [SSL3] or TLS 1.0 [RFC2246] to maintain confidentiality and message integrity. The <Assertion> element(s) in the <Response> MUST be signed.

The service provider MUST process the <Response> message and any enclosed <Assertion> elements as described in [SAMLCore].

## Service Provider Grants or Denies Access to User Agent

To complete the profile, the service provider processes the <Response> and <Assertion>(s) and grants or denies access to the resource. The service provider MAY establish a security context with the user agent using any session mechanism it chooses. Any subsequent use of the <Assertion>(s) provided are at the discretion of the service provider and other relying parties, subject to any restrictions on use contained within them.

# Use of Authentication Request Protocol

---

The Single Sign-On part of this profile is based on the Authentication Request protocol defined in [SAMLCore].

## <AuthnRequest> Usage

A service provider MAY include any message content described in [SAMLCore], Section 3.4.1. All processing rules are as defined in [SAMLCore]. The <Issuer> element MUST be present and MUST contain the unique identifier of the requesting service provider; the Format attribute MUST be omitted or have a value of `urn:oasis:names:tc:SAML:2.0:nameid-format:entity`.

If the identity provider cannot or will not satisfy the request, it MUST respond with a <Response> message containing an appropriate error status code or codes.

If the service provider wishes to permit the identity provider to establish a new identifier for the principal if none exists, it MUST include a <NameIDPolicy> element with the AllowCreate attribute set to "true". Otherwise, only a principal for whom the identity provider has previously established an identifier usable by the service provider can be authenticated successfully.

Note that the service provider MAY include a <Subject> element in the request that names the actual identity about which it wishes to receive an assertion. This element MUST NOT contain any <SubjectConfirmation> elements. If the identity provider does not recognize the principal as that identity, then it MUST respond with a <Response> message containing an error status and no assertions.

The <AuthnRequest> message SHOULD NOT be signed (as directed by the SAML binding used).

Note that because the <AuthnRequest> is not authenticated and/or integrity protected, the information in it MUST NOT be trusted except as advisory. Whether the request is signed or not, the identity provider MUST ensure that any <AssertionConsumerServiceURL> or <AssertionConsumerServiceIndex> elements in the request are verified as belonging to the service provider to whom the response will be sent. Failure to do so can result in a man-in-the-middle attack.

## <Response> Usage

If the identity provider wishes to return an error, it MUST NOT include any assertions in the <Response> message. Otherwise, if the request is successful (or if the response is not associated with a request), the <Response> element MUST conform to the following:

- The <Issuer> element MAY be omitted, but if present it MUST contain the unique identifier of the issuing identity provider; the Format attribute MUST be omitted or have a value of `urn:oasis:names:tc:SAML:2.0:nameid-format:entity`.
- It MUST contain at least one <Assertion>. Each assertion's <Issuer> element MUST contain the unique identifier of the issuing identity provider; the Format attribute MUST be omitted or have a value of `urn:oasis:names:tc:SAML:2.0:nameid-format:entity`.
- The set of one or more assertions MUST contain at least one <AuthnStatement> that reflects the authentication of the principal to the identity provider.
- At least one assertion containing an <AuthnStatement> MUST contain a <Subject> element with at least one <SubjectConfirmation> element containing a Method of `urn:oasis:names:tc:SAML:2.0:cm:bearer`. If the identity provider supports the Single

Simplified Web Browser  
Single Sign-On and Sign-  
Out Profile for SAML 2.0

---

Logout profile, defined in Section 4.4, any such authentication statements MUST include a attribute to enable per-session logout requests by the service provider.

- The bearer <SubjectConfirmation> element described above MUST contain a <SubjectConfirmationData> element that contains a Recipient attribute containing the service provider's assertion consumer service URL and a NotOnOrAfter attribute that limits the window during which the assertion can be delivered. It MAY contain an Address attribute limiting the client address from which the assertion can be delivered. It MUST NOT contain a NotBefore attribute. If the containing message is in response to an <AuthnRequest>, then the InResponseTo attribute MUST match the request's ID.
- Other statements and confirmation methods MAY be included in the assertion(s) at the discretion of the identity provider. In particular, <AttributeStatement> elements MAY be included. The <AuthnRequest> MAY contain an AttributeConsumingServiceIndex XML attribute referencing information about desired or required attributes in [SAMLMeta]. The identity provider MAY ignore this, or send other attributes at its discretion.
- The assertion(s) containing a bearer subject confirmation MUST contain an <AudienceRestriction> including the service provider's unique identifier as an <Audience>.
- Other conditions (and other <Audience> elements) MAY be included as requested by the service provider or at the discretion of the identity provider. (Of course, all such conditions MUST be understood by and accepted by the service provider in order for the assertion to be considered valid.) The identity provider is NOT obligated to honor the requested set of in the <AuthnRequest>, if any.

## <Response> Message Processing Rules

The service provider MUST do the following:

- Verify any signatures present on the assertion(s) or the response.
- Verify that the Recipient attribute in any bearer <SubjectConfirmationData> matches the assertion consumer service URL to which the <Response> or artifact was delivered.
- Verify that the NotOnOrAfter attribute in any bearer <SubjectConfirmationData> has not passed, subject to allowable clock skew between the providers.
- Verify that the InResponseTo attribute in the bearer <SubjectConfirmationData> equals the ID of its original <AuthnRequest> message, unless the response is unsolicited (see ???), in which case the attribute MUST NOT be present.
- Verify that any assertions relied upon are valid in other respects.
- If any bearer <SubjectConfirmationData> includes an Address attribute, the service provider MAY check the user agent's client address against it.
- Any assertion which is not valid, or whose subject confirmation requirements cannot be met SHOULD be discarded and SHOULD NOT be used to establish a security context for the principal.
- If an <AuthnStatement> used to establish a security context for the principal contains a SessionNotOnOrAfter attribute, the security context SHOULD be discarded once this time is reached.profile.

## POST-Specific Processing Rules

If the HTTP POST binding is used to deliver the <Response>, the enclosed assertion(s) MUST be signed.

---

The service provider MUST ensure that bearer assertions are not replayed, by maintaining the set of used ID values for the length of time for which the assertion would be considered valid based on the NotOnOrAfter attribute in the <SubjectConfirmationData>.

## Unsolicited Responses

An identity provider MAY initiate this profile by delivering an unsolicited message to a service provider.

An unsolicited <Response> MUST NOT contain an InResponseTo attribute, nor should any bearer <SubjectConfirmationData> elements contain one. If metadata as specified in [SAMLMeta] is used, the <Response> or artifact SHOULD be delivered to the endpoint of the service provider designated as the default.

Of special mention is that the identity provider MAY include a binding-specific "RelayState" parameter that indicates, based on mutual agreement with the service provider, how to handle subsequent interactions with the user agent. This MAY be the URL of a resource at the service provider. The service provider SHOULD be prepared to handle unsolicited responses by designating a default location to send the user agent subsequent to processing a response successfully.

## Single Log-Out step by step

The following steps are described by the sign-out part of the profile.

### 1. <LogoutRequest> issued by Session Participant to Identity Provider

In step 1, the session participant initiates single logout and terminates a principal's session(s) by sending a <LogoutRequest> message to the identity provider from whom it received the corresponding authentication assertion. The request is sent indirectly to the identity provider through the user agent.

### 2. Identity Provider determines Session Participants

In step 2, the identity provider uses the contents of the <LogoutRequest> message (or if initiating logout itself, some other mechanism) to determine the session(s) being terminated.

The identity provider uses the browser cookies to identify the existing session associated with this user.

If there are no other service providers associated with the user's session, the profile proceeds with step 5. Otherwise, steps 3 and 4 are repeated for each session participant identified.

### 3. <LogoutRequest> issued by Identity Provider to Service Provider

In step 3, the identity provider issues a <LogoutRequest> message to a service provider related to one or more of the session(s) being terminated. The request is sent indirectly to the service provider through the user agent.

### 4. Service Provider issues <LogoutResponse> to Identity Provider

In step 4, a service provider terminates the principal's session(s) as directed by the request (if possible) and returns a <LogoutResponse> to the identity provider. The response is sent indirectly to the identity provider through the user agent.

### 5. Identity Provider issues <LogoutResponse> to Session Participant

In step 5, the identity provider issues a <LogoutResponse> message to the original requesting session participant. The response may be returned directly to the session participant or indirectly through the user agent (if consistent with the form of the request in step 1).

## Note

---

An identity provider (acting as session authority) can initiate this profile at step 2 and issue a <LogoutRequest> to all session participants, also skipping step 5.

## <LogoutRequest> Issued by Session Participant to Identity Provider

If the logout profile is initiated by a session participant, it examines the authentication assertion(s) it received pertaining to the session(s) being terminated, and collects the SessionIndex value(s) it received from the identity provider. If multiple identity providers are involved, then the profile MUST be repeated independently for each one.

To initiate the profile, the session participant issues a <LogoutRequest> message to the identity provider's single logout service request endpoint containing one or more applicable <SessionIndex> elements. At least one element MUST be included. Metadata (as in [SAMLMeta]) MAY be used to determine the location of this endpoint and the bindings supported by the identity provider.

The HTTP Redirect binding provides a RelayState mechanism that the session participant MAY use to associate the profile exchange with the original request. The session participant SHOULD reveal as little information as possible in the RelayState value unless the use of the profile does not require such privacy measures.

## Identity Provider Determines Session Participants

If the logout profile is initiated by an identity provider, or upon receiving a valid <LogoutRequest> message, the identity provider processes the request as defined in [SAMLCore]. It MUST examine the identifier and <SessionIndex> elements and determine the set of sessions to be terminated.

The identity provider then follows steps 3 and 4 for each entity participating in the session(s) being terminated, other than the original requesting session participant (if any), as described in Section 3.7.3.2 of [SAMLCore].

## <LogoutRequest> Issued by Identity Provider to Session Participant/Authority

To propagate the logout, the identity provider issues its own <LogoutRequest> to a session authority or participant in a session being terminated. The request is sent using a SAML binding consistent with the capability of the responder and the availability of the user agent at the identity provider.

In general, the binding with which the original request was received in step 1 does not dictate the binding that may be used in this step except that as noted in step 1, using a synchronous binding that bypasses the user agent constrains the identity provider to use a similar binding to propagate additional requests.

Profile-specific rules for the contents of the <LogoutRequest> message are included in the section called “<LogoutRequest> Usage”.

## Session Participant/Authority Issues <LogoutResponse> to Identity Provider

The session participant/authority MUST process the <LogoutRequest> message as defined in [SAMLCore]. After processing the message or upon encountering an error, the entity MUST issue a

<LogoutResponse> message containing an appropriate status code to the requesting identity provider to complete the SAML protocol exchange.

---

The <LogoutResponse> (or artifact) is returned through the user agent to the identity provider's single logout service response endpoint. Metadata (as in [SAMLMeta]) MAY be used to determine the location of this endpoint and the bindings supported by the identity provider. Any asynchronous binding supported by both entities MAY be used.

## Identity Provider Issues <LogoutResponse> to Session Participant

After processing the original session participant's <LogoutRequest> as described in the previous steps the identity provider MUST respond to the original request with a <LogoutResponse> containing an appropriate status code to complete the SAML protocol exchange.

The response is sent to the original session participant, using a SAML binding consistent with the binding used in the original request, the capability of the responder, and the availability of the user agent at the identity provider. Assuming an asynchronous binding was used in step 1, then any binding supported by both entities MAY be used.

Profile-specific rules for the contents of the <LogoutResponse> message are included in the section called “<LogoutResponse> Usage”.

## Use of the Single Logout Protocol

### <LogoutRequest> Usage

The <Issuer> element MUST be present and MUST contain the unique identifier of the requesting entity; the Format attribute MUST be omitted or have a value of format:entity.

The requester MUST authenticate itself to the responder and ensure the message or using a binding-specific mechanism.

The principal MUST be identified in the request using an identifier that strongly matches the identifier in the authentication assertion the requester issued or received regarding the session being terminated, per the matching rules defined in Section 3.3.4 of [SAMLCore].

If the requester is a session participant, it MUST include at least one <SessionIndex> element in the request. If the requester is a session authority (or acting on its behalf), then it MAY omit any such elements to indicate the termination of all of the principal's applicable sessions.

### <LogoutResponse> Usage

The <Issuer> element MUST be present and MUST contain the unique identifier of the responding entity; the Format attribute MUST be omitted or have a value of urn:oasis:names:tc:SAML:2.0:nameid-format:entity.

The responder MUST authenticate itself to the requester and ensure message integrity, either by signing the message or using a binding-specific mechanism.

## Simplifications

## Requirements to NameID usage

---

The Service Provider and Identity Provider is not required to understand anything by the transient format. However it should be able to treat any other NameID format as a transient value.

## Multiple simultaneous sessions

The Service Provider is not required to handle more than one session per user per cookie-space. That means, Single Log-Out can be simplified, since the receiver of an request can discard the `<SessionIndex>` value of the `<LogoutRequest>`, and instead look up the current session using the cookie of the web browser.

## Use of Attribute Push

Instead of implementing the attribute exchange profile, attributes should be attached in an assertion in the `<Response>` message in the Authenticaiton Request Protocol.

## Use of Metadata

[SAMLMeta] defines an endpoint element, `<md:SingleSignOnService>`, to describe supported bindings and location(s) to which a service provider may send requests to an identity provider using this profile.

The `<md:IDPSSODescriptor>` element's `WantAuthnRequestsSigned` attribute MAY be used by an identity provider to document a requirement that requests be signed. The `<md:SPSSODescriptor>` element's `AuthnRequestsSigned` attribute MAY be used by a service provider to document the intention to sign all of its requests.

The providers MAY document the key(s) used to sign requests, responses, and assertions with `<md:KeyDescriptor>` elements with a use attribute of `sign`. When encrypting SAML elements, `<md:KeyDescriptor>` elements with a use attribute of `encrypt` MAY be used to document supported encryption algorithms and settings, and public keys used to receive bulk encryption keys.

The indexed endpoint element `<md:AssertionConsumerService>` is used to describe supported bindings and location(s) to which an identity provider may send responses to a service provider using this profile. The `index` attribute is used to distinguish the possible endpoints that may be specified by reference in the `<AuthnRequest>` message. The `index` attribute is used to specify the endpoint to use if not specified in a request.

The `<md:SPSSODescriptor>` element's `WantAssertionsSigned` attribute MAY be used by a service provider to document a requirement that assertions delivered with this profile be signed. This is in addition to any requirements for signing imposed by the use of a particular binding. Note that the identity provider is not obligated by this, but is being made aware of the likelihood that an unsigned assertion will be insufficient.

If the request or response message is delivered using the HTTP Artifact binding, the artifact issuer MUST provide at least one `<md:ArtifactResolutionService>` endpoint element in its metadata.

The `<md:IDPSSODescriptor>` MAY contain `<md:NameIDFormat>`, `<md:AttributeProfile>`, and `<saml:Attribute>` elements to indicate the general ability to support particular name identifier formats, attribute profiles, or specific attributes and values. The ability to support any such features during a given authentication exchange is dependent on policy and the discretion of the identity provider.

## Simplified Web Browser

### Single Sign-On and Sign-

---

The `<md:SPSSODescriptor>` element MAY also be used to document the service provider's need or desire for SAML attributes to be delivered along with authentication information. The actual inclusion of attributes is always at the discretion of the identity provider. One or more `<md:AttributeConsumingService>` elements MAY be included in its metadata, each with an `index` attribute to distinguish different services that MAY be specified by reference in the `<AuthnRequest>` message. The `isDefault` attribute is used to specify a default set of attribute requirements.

[SAMLMeta] defines an endpoint element, `<md:SingleLogoutService>`, to describe supported bindings and location(s) to which an entity may send requests and responses using Single Log-Out.

A requester, if encrypting the principal's identifier, can use the responder's `<md:KeyDescriptor>` element with a `use` attribute of encryption to determine an appropriate encryption algorithm and settings to use, along with a public key to use in delivering a bulk encryption key.

# Robustness of HTTP-REDIRECT Single Log-Out

TODO!!!!