SAML V2.0 Change Notify Protocol Version 1.0

Committee Specification Draft 02 / Public Review Draft 02

17 May 2011

Specification URIs:

This version:

http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml2-notify-protocol/v1.0/csprd02/sstc-saml2-notify-protocol-v1.0-csprd02.odt (Authoritative)

http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml2-notify-protocol/v1.0/csprd02/sstc-saml2-notify-protocol-v1.0-csprd02.html

http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml2-notify-protocol/v1.0/csprd02/sstc-saml2-notify-protocol-v1.0-csprd02.pdf

Previous version:

http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml2-notify-protocol/v1.0/csprd01/sstc-saml2-notify-protocol-v1.0-csprd01.odt (Authoritative)

http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml2-notify-protocol/v1.0/csprd01/sstc-saml2-notify-protocol-v1.0-csprd01.html

http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml2-notify-protocol/v1.0/csprd01/sstc-saml2-notify-protocol-v1.0-csprd01.pdf

Latest version:

http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml2-notify-protocol/v1.0/sstc-saml2-notify-protocol-v1.0.odt (Authoritative)

http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml2-notify-protocol/v1.0/sstc-saml2-notify-protocol-v1.0.html

http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml2-notify-protocol/v1.0/sstc-saml2-notify-protocol-v1.0.pdf

Technical Committee:

OASIS Security Services (SAML) TC

Chairs:

Thomas Hardjono, M.I.T.

Nate Klingenstein, Internet2

Editors:

Phil Hunt, Oracle Corporation

Thinh Nguyenphu, Nokia Siemens Networks

Related work:

This specification is related to:

- Security Assertion Markup Language (SAML) v2.0 OASIS Standard
- XML schemas: sstc-saml2-notify-protocol/v1.0/csprd02/xml/

Declared XML namespaces:

urn:oasis:names:tc:SAML:2.0:notify

Abstract:

The SAML V2.0 Change Notify Protocol describes request and response messages for informing SAML endpoints about available changes to subjects and attributes associated with subjects.

Status:

This document was last revised or approved by the OASIS Security Services (SAML) TC on the above date. The level of approval is also listed above. Check the "Latest version" location noted above for possible later revisions of this document.

Technical Committee members should send comments on this specification to the Technical Committee's email list. Others should send comments to the Technical Committee by using the "Send A Comment" button on the Technical Committee's web page at http://www.oasis-open.org/committees/security/.

For information on whether any patents have been disclosed that may be essential to implementing this specification, and any offers of patent licensing terms, please refer to the Intellectual Property Rights section of the Technical Committee web page (http://www.oasis-open.org/committees/security/ipr.php).

Citation format:

When referencing this specification the following citation format should be used:

[SSTC-SAML2-NOTIFY-PROTOCOL-V1.0]

SAML V2.0 Change Notify Protocol Version 1.0. 17 May 2011. OASIS Committee Specification Draft 02 / Public Review Draft 02. http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml2-notify-protocol/v1.0/csprd02/sstc-saml2-notify-protocol-v1.0-csprd02.html

Notices

All capitalized terms in the following text have the meanings assigned to them in the OASIS Intellectual Property Rights Policy (the "OASIS IPR Policy"). The full Policy may be found at the OASIS website.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published, and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this section are included on all such copies and derivative works. However, this document itself may not be modified in any way, including by removing the copyright notice or references to OASIS, except as needed for the purpose of developing any document or deliverable produced by an OASIS Technical Committee (in which case the rules applicable to copyrights, as set forth in the OASIS IPR Policy, must be followed) or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by OASIS or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and OASIS DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY OWNERSHIP RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

OASIS requests that any OASIS Party or any other party that believes it has patent claims that would necessarily be infringed by implementations of this OASIS Committee Specification or OASIS Standard, to notify OASIS TC Administrator and provide an indication of its willingness to grant patent licenses to such patent claims in a manner consistent with the IPR Mode of the OASIS Technical Committee that produced this specification.

OASIS invites any party to contact the OASIS TC Administrator if it is aware of a claim of ownership of any patent claims that would necessarily be infringed by implementations of this specification by a patent holder that is not willing to provide a license to such patent claims in a manner consistent with the IPR Mode of the OASIS Technical Committee that produced this specification. OASIS may include such claims on its website, but disclaims any obligation to do so.

OASIS takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Information on OASIS' procedures with respect to rights in any document or deliverable produced by an OASIS Technical Committee can be found on the OASIS website. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this OASIS Committee Specification or OASIS Standard, can be obtained from the OASIS TC Administrator. OASIS makes no representation that any information or list of intellectual property rights will at any time be complete, or that any claims in such list are, in fact, Essential Claims.

The names "OASIS" and “SAML” are trademarks of OASIS, the owner and developer of this specification, and should be used only to refer to the organization and its official outputs. OASIS welcomes reference to, and implementation and use of, specifications, while reserving the right to enforce its marks against misleading uses. Please see http://www.oasis-open.org/who/trademark.php for above guidance.

Table of Contents

118
119

1 Introduction

The Change Notify Protocol is a message exchange protocol by which a service provider (e.g. web service provider, identity provider) notifies a federated service provider of changes to principals and related attributes in a federated system. After notification, the receiver of the notification is then able to take an appropriate action to effect appropriate changes to affected principals.

This message exchange protocol uses the SAML Protocols V2.0 [SAML2Core] and bindings [SAML2Bind].

1.1 Notation

This specification uses normative text. The keywords "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this specification are to be interpreted as described in [RFC2119]:

> …they MUST only be used where it is actually required for interoperation or to limit behavior which has potential for causing harm (e.g., limiting retransmissions)…

These keywords are thus capitalized when used to unambiguously specify requirements over protocol and application features and behavior that affect the interoperability and security of implementations. When these words are not capitalized, they are meant in their natural-language sense.

```
Listings of XML schemas appear like this.
```

```
Example code listings appear like this.
```

Conventional XML namespace prefixes are used throughout the listings in this specification to stand for their respective namespaces as follows, whether or not a namespace declaration is present in the example:

Prefix	XML Namespace	Comments
`saml:`	urn:oasis:names:tc:SAML:2.0:assertion	This is the SAML V2.0 assertion namespace defined in the SAML V2.0 core specification [SAML2Core].
`samlp:`	urn:oasis:names:tc:SAML:2.0:protocol	This is the SAML V2.0 core protocol namespace defined in the SAML V2.0 core specification [SAML2Core].
`samln:`	urn:oasis:names:tc:SAML:2.0:notify	This is the new Change Notify protocol namespace defined in this document.
`xs:`	http://www.w3.org/2001/XMLSchema	This is the XML Schema namespace [Schema1].
`xsi:`	http://www.w3.org/2001/XMLSchema-instance	This is the XML Schema namespace for schema-related markup that appears in XML instances [Schema1].

This specification uses the following typographical conventions in text: `<SAMLElement>`, `<ns:ForeignElement>`, `Attribute`, Datatype, `OtherCode`.

1.2 Terminology

Notify Issuer The issuer of a change notification request is a SAML Requester. The issuer MAY be any SAML entity, including but not limited to a relying party or an identity provider.

Notify Target The target of a change notification is a SAML Responder. The responder MAY be any SAML entity, including but not limited to a relying party or an identity provider.

Subject Any principle or entity that can be referenced by a SAML Name Identifier. A subject is the object about which change notifications are made.

1.3 Normative References

[RFC2119] S. Bradner. *Key words for use in RFCs to Indicate Requirement Levels.* IETF RFC 2119, March 1997. http://www.ietf.org/rfc/rfc2119.txt

[RFC2246] T. Dierks. *The TLS Protocol Version 1.0*. IETF RFC 2246, January 1999, See http://www.ietf.org/rfc/rfc2246.txt

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

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

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

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

[Schema1] H. S. Thompson et al. *XML Schema Part 1: Structures.* World Wide Web Consortium Recommendation, May 2001. See http://www.w3.org/TR/2001/REC-xmlschema-1-20010502/

[SSL3] A. Frier et al. *The SSL 3.0 Protocol*. Netscape Communications Corp, November 1996.

1.4 Non-normative References

[OpenID] OpenID Community, OpenID Authentication 2.0, December 5, 2007. http://openid.net/specs/openid-authentication-2_0.html

[Portable] Joseph Smarr, Plaxo, 5 August 2008. http://portablecontacts.net/draft-spec.html

[RFC2251] M. Wahl, T. Howes, S. Kille, Lightweight Directory Access Protocol (v3), IETF RFC 2251, December 1997. http://www.ietf.org/rfc/rfc2251.txt

[SPMLv2] G. Cole et al. OASIS Service Provisioning Language (SPML) Version 2, 1 April 2006. http://www.oasis-open.org/committees/download.php/17708/pstc-spml-2.0-os.zip

[WS-Trust] Anthony Nadalin, Marc Goodner, et. al., OASIS WS-Trust 1.3 Specification, March 2007. http://docs.oasis-open.org/ws-sx/ws-trust/200512

2 SAML V2.0 Change Notify Protocol

2.1 Required Information

This section describes all of the required information of a profile as defined in section 2.1 of the Profile the Profiles for the OASIS Security Assertion Markup Language (SAML) V2.0 [SAML2Prof].

Identification: `urn:oasis:names:tc:SAML:2.0:notify`

Contact information: security-services-comment@lists.oasis-open.org

Description: Given below.

Updates: None.

2.2 Description

The SAML Change Notify Protocol is a two-step message exchange protocol by which a Notify Issuer (SAML Requester) notifies a Notify Target server (SAML Responder) of changes to Subjects and related attributes. The Notify Issuer and Notify Target server each MAY be a Service Provider and/or Identity Provider. After a change notification has been received, the Issuer and Target servers are able to negotiate secondary actions to propagate changes, if appropriate, in a protocol agnostic fashion. This message exchange protocol uses the SAML Protocols V2.0 [SAML2Core] and SAML Profile specifications [SAML2Prof].

In typical SAML scenarios, user information is propagated through the use of the Browser SSO Profile [SAML2Prof] and similar profile variants. However, except for just-in-time SSO provisioning, and for the SAML Name Identifier Management Protocol [SAML2Core], there is no clear common method by which federated SAML entities can inform each other of changes to user principals and attributes that occur over time. Change Notify Protocol allows service providers to coordinate subject changes while maintaining separate state and administrative control. Instead of initiating specific data change commands, Change Notify Protocol simply informs service providers about changes that may be of interest.

Further, Change Notify Protocol allows service providers to infer more meaning information than that available from existing SAML protocol features. For example, while the `<Terminate>` option of `<ManageNameIDRequest>` is used for de-federation, Change Notify Protocol adds functionality to distinguish between de-federation and a de-provisioning event. Some examples include:

- An enterprise provisioning and de-provisioning accounts to cloud service providers
- An enterprise updating employee roles and attributes persisted in the cloud
- An IDP informing RPs that retained information (e.g. from a past SAML Attribute Query) requires updating.

There are many instances where service providers that generate identity related attributes wish to inform IDPs of available changes. Some examples include:

- A service provider migrating legacy database/directory users to a federated provider
- A service provider transferring a user from one IDP to another
- A service provider generating or updating attribute data for which it is deemed authoritative

As part of the Change Notify request, the Notify Issuer specifies one or more protocol URIs that it wants to use to facilitate transfer or management of data. Examples include:

- SAML AttributeQuery (for back-channel mode)
- SAML Web SSO (for front-channel mode)
- SPMLv2 [SPMLv2]
- PortableContacts [Portable]

- Other

The request also includes information on the nature of the change, the affected subjects, and affected attributes.

The Notify Target responds with a Change Notify Protocol response message that indicates acknowledgment and the chosen data transfer protocol.

2.3 Assumptions

It is assumed that the Notify Issuer and Notify Target have agreements with each other that permits the exchange of attributes and extended status information between parties.

Such agreements might include:

- Definitions of how Change Notify Protocol operations are to be issued and interpreted by parties. For example, what happens when a Notify Target receives a RetireSubject notification. Does it delete the subject, disable the subject, or suspend the subject?
- Definitions of what notifications will be issued for which entities between servers.
- Definitions of how many transactions may be included in a single request-response exchange, and how frequently they may occur.
- Definitions of how updates between parties impacts and supports overall subject provisioning and management.
- Definitions of which protocols are to be used within specific circumstances. For example, after receiving notification of a large number of NewSubjects, the responder MAY wish to make a dynamic decision to use SPML instead of SAML AttributeQuery to process the subjects at a later time.

Exact terms of such an agreement are out of scope of this specification, However, the exact interpretation of the Change Notify request and response messages, processing, and profile are defined in this specification.

2.4 Status URIs

In addition to the Status URIs defined in [SAML2Core], the following top-level `<samlp:StatusCode>` is defined related to Change Notify protocol:

`urn:oasis:names:tc:SAML:2.0:status:notify:protocol`
: The request could not be performed as the protocol was unavailable at the time of the request for the subjects, and/or notification elements requested.

2.5 Protocol URIs

In the protocol, the issuer and target MAY negotiate a protocol to implement changes indicated in change notify requests. The protocols supported MAY include but are not limited to the following URIs:

`urn:oasis:names:tc:SAML:2.0:notify:protocol:SAML:BackChannel`
: In back-channel (synchronous) mode, this URI indicates that Notify Target will query the Notify Issuer for the affected SAML Identifier using SAML `AttributeQuery`. When initiated in front-channel (asynchronous/mixed) mode, indicates that information will be exchanged via a back-channel by using SAML AttributeQuery . For `<RetireSubject>` elements, indicates that SAML `<ManageNameIDRequest>` will be used.

`urn:oasis:names:tc:SAML:2.0:notify:protocol:SAML:FrontChannel`
: In front-channel mode (asynchronous) mode, `this URI` indicates that information will be exchanged via the `<AuthnRequest>/<Response>` SAML protocol using the any supported profile (e.g. web SSO) of the Authentication Request protocol. If target initiated, the request will begin with an `<AuthnRequest>`. If initiated by the Issuer, the Issuer will simply use an unsolicited `<Response>` message to transfer the user. For `<RetireSubject>` elements, no further action will be taken.

`urn:oasis:names:tc:SAML:2.0:notify:protocol:STS`
: Indicates that change information will be exchanged via WS-Trust protocol [WS-Trust]. Typically the Target initiates WS-Trust transactions to the endpoint defined by the issuer.

`urn:oasis:names:tc:SAML:2.0:notify:protocol:OpenID`
: Indicates that change information will be exchanged via OpenID protocol [OpenID]. Typically the Target initiates OpenID transactions to the OpenID endpoint defined by the issuer.

`urn:oasis:names:tc:SAML:2.0:notify:protocol:SPMLv2`
: Indicates that change information will be exchanged via SPMLv2 protocol [SPMLv2]. Typically the issuer initiates SPML transactions to the endpoint defined by the Target.

`urn:oasis:names:tc:SAML:2.0:notify:protocol:LDAPv3`
: Indicates that change information will be exchanged via LDAPv3 protocol. If the Notify Issuer is declared the initiator, then the Notify Issuer will follow with one or more LDAP Add, Modify, and/or Delete operations, as defined in [RFC2251]. If the Notify Target is declared the initiator, the target will initiate action with one or more LDAP Search operations.

`urn:oasis:names:tc:SAML:2.0:notify:protocol:PortableContact`
: Indicates that the `<saml:Subject>`s will be transferred by the Notify Target using the PortableContacts specification [Portable] using the endpoint specified by the issuer.

`urn:oasis:names:tc:SAML:2.0:notify:protocol:Other`
: Indicates that change information will be exchanged via a protocol negotiated via end-point URIs.

`urn:oasis:names:tc:SAML:2.0:notify:protocol:None`
: Indicates that no transactional action will take place.

2.6 Element <ChangeNotifyRequest>

Used by a Notify Issuer to send a `<ChangeNotifyRequest>` message that SHALL contain one or more of the following Notification Elements: `<NewSubject>`, `<ModifySubject>`, or `<RetireSubject>`.

This `<ChangeNotifyRequest>` message is a complex type based on `ChangeNotifyRequestType`, which extends `RequestAbstractType`.

The `<ChangeNotifyRequest>` element allows for one or more notification elements to allow multiple change notifications to be passed in a single request message. It includes the following attributes:

`expires` [optional]
: The time at which the notified changes expire. Default is never.

`protocol` [required]
: The URI of a protocol that MAY be used to act or implement a change as defined in section 2.5, or any other URIs pre-negotiated between service providers.

`endpoint` [optional]
: The URI of the Notifiers service endpoint associated with the protocol. When omitted, the endpoint is assumed to be the current endpoint of the request message issuer.

`issuerInitiated` [default=true]
: A flag indicating whether the issuer is to initiate the action operation.

`redirect_uri` [optional]
: An optional URI that can be used to redirect the browser to a new site following the completion of the action protocol step. For example, this option MAY be used in the front-channel to redirect the browser back to the Notifier after completion of a an operation at a Target service provider.

The following schema fragment defines the `<ChangeNotifyRequest>` protocol message:

```
<element name="ChangeNotifyRequest" type="samln:ChangeNotifyRequestType" />
<complexType name="ChangeNotifyRequestType">
```

```
      <complexContent>
            <extension base="samlp:RequestAbstractType">

                  <sequence>
                        <choice>
    <element name="NewSubject" type="samln:NewSubjectType" minOccurs="0"
       maxOccurs="unbounded" />
    <element name="ModifySubject" type="samln:ModifySubjectType"
       minOccurs="0" maxOccurs="unbounded" />
    <element name="RetireSubject" type="samln:ChangeSubjectType"
       minOccurs="0" maxOccurs="unbounded" />
                        </choice>
                  </sequence>
                  <attribute name="expires" type="dateTime" use="optional"/>
                  <attribute name="protocol" type="anyURI" use="required"/>
                  <attribute name="endpoint" type="anyURI" use="optional"/>
                  <attribute name="issuerInitiated" type="boolean"
default="true"/>
                  <attribute name="redirect_uri" type="anyURI"
use="optional"/>
            </extension>
      </complexContent>
</complexType>
```

2.7 Notification Elements

Notification elements are an extension of <ChangeSubjectType> which defines a common type for defining changes to a particular subject entity. Notification elements <NewSubject>, <ChangeSubject>, and <RetireSubject> define the basic transaction notifications that are available in a <ChangeNotifyRequest>.

```
   <complexType name="ChangeSubjectType">
      <sequence>
            <choice>
            <element ref="saml:BaseID"/>
            <element ref="saml:NameID"/>
            <element ref="saml:EncryptedID"/>
         </choice>
      </sequence>
      </complexType>
```

2.7.1 Notification Element <NewSubject>

The <NewSubject> element has the complex type <NewSubjectType> , an extension of <ChangeSubjectType> which requires that one or more identifier elements <saml:NameID>, <saml:BaseID>, or <saml:EncryptedID> elements be provided. In addition, the Issuer MAY also include a list of one or more <saml:Attribute> elements listing the attributes available for every identifier listed within the current <NewSubject> element.

The purpose of this element is to allow an Issuer to notify a Target server of principals that are “new” to the Issuer.

```
 <element name="NewSubject" type="samln:NewSubjectType" minOccurs="0"
       maxOccurs="unbounded" />
 <complexType name="NewSubjectType">
       <complexContent>
            <extension base="samln:ChangeSubjectType">
                  <sequence>
                        <element ref="saml:Attribute"
                              minOccurs="0" maxOccurs="unbounded" />
            </sequence>
      </extension>
    </complexContent>
</complexType>
```

2.7.2 Notification Element <ModifySubject>

The <ModifySubject> element has the complex type <ModifySubjectType> , an extension of <ChangeSubjectType> which requires that one or more SAML Identifier elements <saml:NameID>, <saml:BaseID>, or <saml:EncryptedID> elements be provided. In addition, the Issuer MAY include a list of one or more <saml:Attribute> elements listing the modified attributes for each identifier listed within the current <ModifySubject> element.

The purpose of this element is to allow an Issuer to notify a Target server of changes to a subject's attributes.

```
<element name="ModifySubject" type="samln:ModifySubjectType"
      minOccurs="0" maxOccurs="unbounded" />
<complexType name="ModifySubjectType">
      <complexContent>
            <extension base="samln:ChangeSubjectType">
                  <sequence>
                        <element ref="saml:Attribute" minOccurs="0"
                        maxOccurs="unbounded" />
                  </sequence>
            </extension>
      </complexContent>
</complexType>
```

2.7.3 Notification Element <RetireSubject>

The <RetireSubject> element is based on the complex type <ChangeSubjectType> and allows for one or more SAML Identifier elements to be specified.

The purpose of this element is to allow the issuer to notify the target server that the record is to be retired or de-provisioned. The exact function (e.g. deletion, disablement, suspension) of this action is typically defined in a Issuer/Target service level agreement.

```
<element name="RetireSubject" type="samln:ChangeSubjectType"
      minOccurs="0" maxOccurs="unbounded" />
```

2.8 Element <ChangeNotifyResponse>

The recipient of the <ChangeNotifyRequest> message MUST respond with a <ChangeNotifyResponse> message, which is of type <samln:ChangeNotifyResponseType>.

The <ChangeNotifyResponse> element allows for one or more OPTIONAL notification elements to allow acknowledgment to multiple change notifications to the Notifier by the Target. It includes the following attributes:

endpoint [optional]

The URI of a service endpoint for the Notify Target associated with the protocol. When omitted, the endpoint is assumed to be the current endpoint of the notify responder.

issuerInitiated [default=true]

A flag confirming whether the issuer is to initiate the action operation. The value of this attribute overrides the value provided in the <ChangeNotifyRequest>.

redirect_uri [optional]

An optional URI that can be used to redirect the browser to a new site following the completion of the action protocol specified in the <ChangeNotifyRequest>. For example, this option MAY be used in the front-channel to redirect the browser back to the Notifier after completion of a an operation at a Target service provider.

actionAfter [optional]

Specifies the time at which the initiator MAY begin the specified change action protocol step. Default is immediately.

`actionDeclined` [default=false]

Allows the Notify Target to indicate that the request has been successfully accepted but that no further action is required. This attribute is typically used in connection with `<RetireSubject>` notification elements.

The following schema fragment defines the `<ChangeNotifyResponse>` protocol message:

```
<element name="ChangeNotifyResponse" type="samln:ChangeNotifyResponseType" />
<complexType name="ChangeNotifyResponseType">
      <complexContent>
            <extension base="samlp:StatusResponseType">
                  <sequence>
                        <choice>
    <element name="NewSubject" type="samln:NewSubjectType" minOccurs="0"
      maxOccurs="unbounded" />
    <element name="ModifySubject" type="samln:ModifySubjectType"
      minOccurs="0" maxOccurs="unbounded" />
    <element name="RetireSubject" type="samln:ChangeSubjectType"
      minOccurs="0" maxOccurs="unbounded" />
                        </choice>
                  </sequence>
                  <attribute name="endpoint" type="anyURI" use="optional"/>
                  <attribute name="issuerInitiated" type="boolean"
default="true"/>
                  <attribute name="redirect_uri" type="anyURI"
use="optional"/>
                  <attribute name="actionAfter" type="dateTime"
use="optional"/>
                  <attribute name="actionDeclined" type="boolean"
default="false" use="optional"/>
            </extension>
      </complexContent>
</complexType>
```

2.9 Processing Rules

The Notify Issuer of the `<ChangeNotifyRequest>` message:

- MUST include at least one change notification element (`<NewSubject>`, `<ModifySubject>`, or `<RetireSubject>`);
- A notification element MAY include more than one SAML Identifier;
- A separate new notification element (e.g. `<ModifySubject>`) MUST be used for each differing set of attributes. Multiple subjects MAY be changed in ONE notification element provided the list of attributes remain the same;
- MUST indicate the protocol to be used to facilitate the changed by providing a `protocol` attribute value in the form of a URI;
- The Identifiers used within the change notification elements MUST be appropriate to the protocol URI defined in the `protocol` attribute;
- MAY include the attribute `expires` is present in the element `<ChangeNotifyRequest>`, the availability or validity of the changes contained will be deemed to have expired on the specified date/time. If the attribute is absent, the notification information is deemed not to expire;
- When using the `<RetireSubject>` change notifier element, the requestor MUST either sign the `<ChangeNotifyRequest>` message or use a binding-specific mechanism that ensures authenticity and integrity of the message.

The responding Notify Target of the `<ChangeNotifyRequest>` message:

- SHOULD respond with `<Status>` value of `urn:oasis:names:tc:SAML:2.0:status:notify:protocol` if the Notify Target is unable or does not wish to proceed with the protocol defined in the `<ChangeNotifyRequest>` message. After receiving such a status, the Notify Issuer MAY repeat the request with a new `protocol;`

- MAY include endpoint attribute which specifies the service endpoint for the Notify Target associated with the specified protocol;

- MAY include <NewSubject>, <ModifySubject>, <RetireSubject> sub-elements to indicate the processing action SHALL be restricted to only those NameID(s) specified in the notify sub-elements. If <NewSubject>, <ModifySubject>, <RetireSubject> sub-elements are not included, then the Notify Target is indicating that all changes will be process as per the original <ChangeNotifyRequest> message.

- MAY include <saml:Attribute> elements within the <NewSubject> or <ModifySubject> elements, to indicate the processing SHALL be restricted to the specified <saml:Attribute>s in a subsequent action. If <saml:Attribute> elements are not provided, the responder is indicating that the attributes specified in the <ChangeNotifyRequest> message SHALL be used;

- MAY include the attribute actionAfter to indicate to the Notify Issuer that action operations SHOULD begin on or after the date/time specified. If the attribute is absent, it is assumed that the responder intends action to begin immediately;

- MAY include the attribute actionDeclined to indicate to the Notify Issuer that no further action is required (e.g. as a result of receiving <ReturnSubject> notifications) and does not indicate an error condition;

- If the Notify Target does not recognize the <ChangeNotifyRequest>, the Notify Target MUST responds to the Notify Issuer with <ChangeNotifyResponse> with <status> of urn:oasis:names:tc:SAML:2.0:status:Responder.

3 Bindings

Mappings of the SAML Change Notify Protocol request-response message exchanges onto standard messaging or communications protocols follow the core SAML Protocol Bindings specifications (saml-bindings-2.0-os) [SAML2-Bind].

4 Profile

The Change Notify Protocol has one universal profile that can be used in both front-channel and back-channel modes and can be used in conjunction with other SAML Profiles such as the Web Browser SSO Profile [SAML2-Prof]. In front-channel mode, an "issuer site" (known as Issuer) MAY notify a "target site" (Target) of a new or changed, or retired subject profile related to the currently authenticated subject. In back-channel mode, a Notifier can notify a Target of several changes about subjects in "batch" mode. Finally, a mix mode is supported whereby an front-channel notification MAY be combined with a back-channel transfer of information (e.g. using SAML AttributeQuery). The Change Notify Protocol is used in conjunction with HTTP Redirect, and HTTP Post.

4.1 Required Information

This section describes all of the required information of a profile as defined in section 2.1 of the Profile the Profiles for the OASIS Security Assertion Markup Language (SAML) V2.0 [SAML2Prof].

Identification: urn:oasis:names:tc:SAML:2.0:profiles:notify

Contact Information: security-services-comment@lists.oasis-open.org

Description: See below.

4.2 Profile Overview

In the Change Notify profile, a `<ChangeNotifyRequest>` is issued by a SAML Requester (known as Notify Issuer) providing one or more changes impacting one or more subjects. The SAML Responder (known as Notify Target) signals its agreement to exchange information in a subsequent step, known as the action protocol step by responding with a `<ChangeNotifyResponse>` message. Following the protocol exchange, the requestor and responder begin an exchange of information using the protocol indicated in the original `<ChangeNotifyRequest>`.

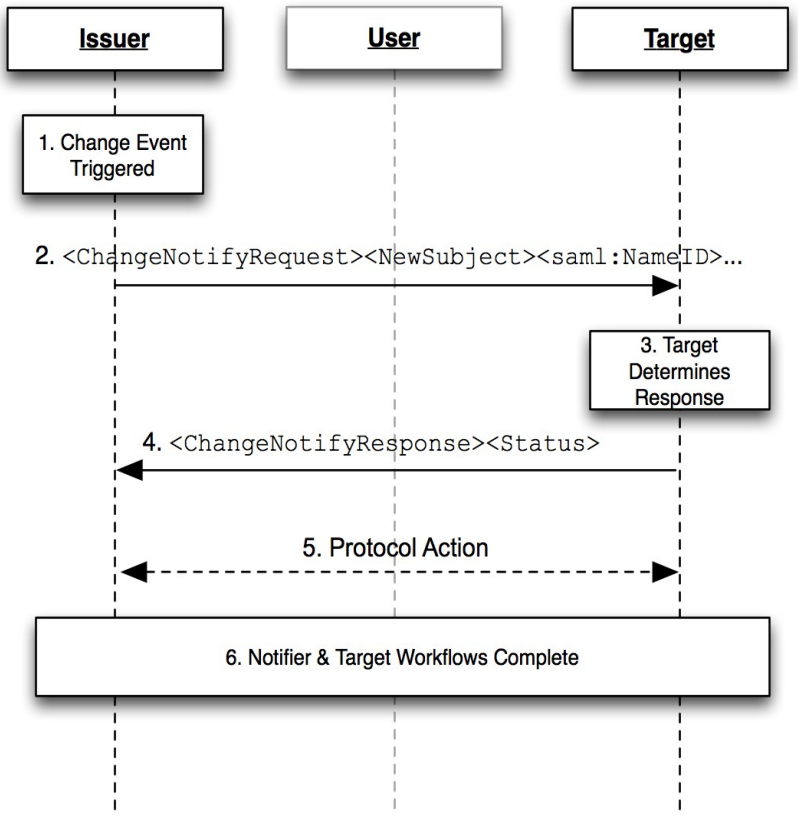


Figure 1: Base Change Notify Profile

The grayed-out user illustrates that the message exchange may pass through a user agent or may be a direct exchange between notification entities (Issuer and Target), depending on the binding used to implement the profile.

The following steps are described by the profile. Within each step, there MAY be variation on the actual message exchanges depending on the binding used for that step, and the subsequent protocol selected for transfer of information between Notification parties.

Change Notify protocol flow is intended to allow an Issuer and Target to coordinate updates to entities of common interest. Change Notify Protocol enables the Notifer to communicate changes that it believes to be of interest without having to know the state of data within the Target. On receiving a change notification, the Target is able to determine how to proceed and to place the change notification in a context that makes sense within its service "domain".

1. **Change Event Triggered**

   A workflow event triggers the Notify Issuer node to determine that there is a change of interest to a Notify Target server. An event can consist of one or more changes to one or more subjects.

2. **<ChangeNotifyRequest> issued by Notify Issuer**

   The Notify node, takes the set of changes and forms a request by including one or more change notify elements. As part of the request, the Notifier MUST indicate the protocol to be used in step 5, and which party is to initiate the step.

3. **Target Determines Response**

   The Target server receives the change notification and determines how to process the incoming change given its knowledge of the current state of potentially affected entities in its domain.

4. **Target Responds with <ChangeNotifyResponse>**

   The Target issues a response containing either no notifications, or listing only those notification elements and subject identifiers with which it wishes to proceed with. The Target also confirms when processing time is to begin. The Target MAY also indicate that no further processing is required by setting the attribute `actionDeclined`, or it MAY indicate a desire to change protocols by responding with a `<Status>` of `urn:oasis:names:tc:SAML:2.0:status:notify:protocol`

5. **Protocol Action**

   Based on the protocol URI supplied in the `<ChangeNotifyRequest>` and the value of the attribute issuerInitiated, the endpoints proceed to exchange information using an SAML 2 protocol, or by using another protocol. Note that the exact process for this exchange is out of scope for this specification.

6. **Notifier & Target Workflow Completion**

   Based on the selected protocol and the value of `redirect_uri` attribute, the endpoints complete their processing and for front-channel cases, the user-agent is redirected appropriately.

4.3 Front-Channel Examples

4.3.1 SP Initiated Change Using Web Browser SSO

This example demonstrates a web service provider transferring a signed on user context to an IDP for the purpose of provisioning a user to the IDP. In this situation, it assumed, though not guaranteed, that the SP is already familiar with the user, while the IDP likely does not have a pre-existing relationship with the user. The effect is to allow the SP to provide a "warm-introduction" of the user to the IDP.

The following figure illustrates an example of transferring a subject from a Service Provider acting as a Notify Issuer server to a Notify Target server (acting as an Identity Provider) using Web SSO to achieve the transfer of attributes and to maintain authentication state between the parties. The service provider issue a `<ChangeNotifyRequest>` notification request to the identify provider to add this user as a new subject. Once the Change Notify protocol followed by the action protocol step are completed, the service provider resumes the Web SSO authentication request, per the normal Web SSO Profile allowing the user to access a resource at the service provider using a SSO from the IDP.

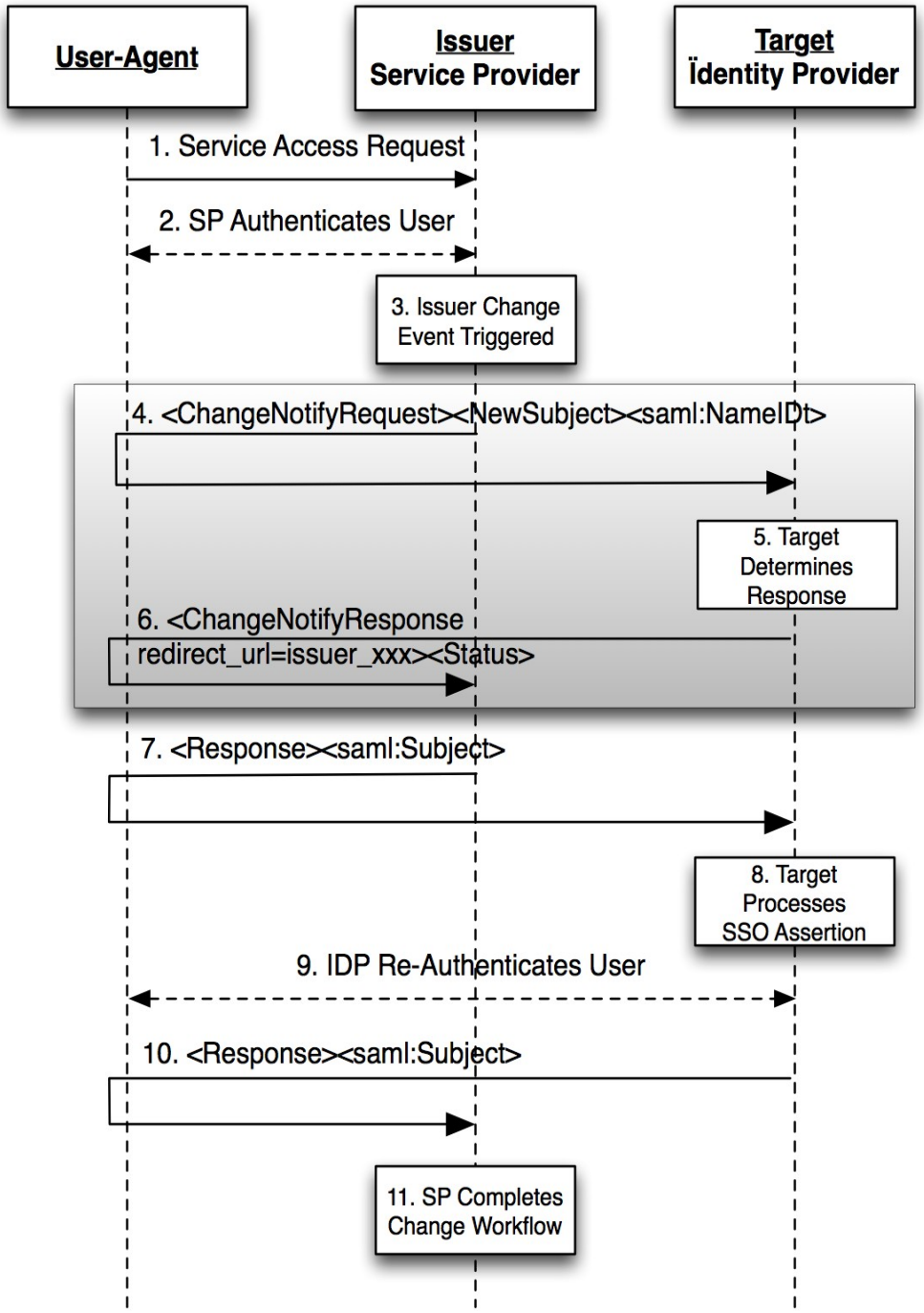


Figure 2: Change Notify Web SSO at Service Provider

1) The user makes request for a secure resource at the service provider without security context; possibly triggering a provisioning workflow.

2) If not already performed, the SP authenticates the user, via local or other federated means.

3) Notify Issuer (service provider) interprets a locally generated change event and determines a Target (identity provider) interested in potentially receiving the notification.

4) Notify Issuer (service provider) sends an `<ChangeNotifyRequest>` with `<NewSubject>` notification element (or the notification MAY also be a `<ModifySubject>` or `<RetireSubject>` element) including a `<saml:NameID>`. The Notifier, sets the attributes `issuerInitiated` to `true`, and the `protocol` attribute to:
`urn:oasis:names:tc:SAML:2.0:notify:protocol:SAML:FrontChannel`

Notify Target (IdP) processes the notification request and accepts the request.

5) In response, Notify Target send a `<ChangeNotifyResponse>`, to the Notify Issuer.

6) In response to the `protocol` and `issuerInitiated` attributes of the `<ChangeNotifyRequest>`, the Notify Issuer initiates the protocol step by issuing an unsolicited `<samlp:response>` to the Notify Target endpoint, thereby facilitating the new subject transfer and including the user's SSO context.

Note: Step 4-6 are the procedures from Change Notify Protocol.

7) The Notify Target processes the inbound SSO SAML Assertion and provisions the new subject as appropriate.

8) Optionally, the Notify Target MAY choose to re-authenticate the user within its own administrative domain.

9) The Notify Target uses the value of `redirect_uri` passed in the initial `<ChangeNotifyRequest>` to pass the user-agent back to the Notify Issuer, including a web SSO assertion from the Identity Provider.

10) The Notify Issuer is now able to proceed with any final event workflow requirements (e.g. local de-provisioning).

4.3.1.1 Mixed Front and Back Channel Variation

In a mixed channel variation, an front-channel notification is transmitted via the browser while SAML Assertion data is transferred in a back-channel. The intention here is to provide greater workflow flexibility between providers.

In the previous example, the `protocol` URI in step 4 is set to `urn:oasis:names:tc:SAML:2.0:notify:protocol:SAML:BackChannel`, while `issuerInitiated` is set to `false`. The effect would be to cause step 7 to be replaced with a back-channel SAML Attribute Query initiated by the Notify Target instead of an Unsolicited SAML Response from the Notify Issuer in step 7.

4.3.2 IDP Initiated Change Using Web Browser SSO

Figure 3 shows a user initially accessing an Identity Provider site action which triggers a change for a target Service Provider. This triggers the <ChangeNotifyRequest> to the Service Provider. Once the Change Notify with the Action Protocol procedures are completed, the Identity Provider sends unsolicited <response>, per the SAML Web SSO Profile [SAML2Prof]. Note that the grayed block area shows the Change Notification protocol portion of the overall exchange sequence.

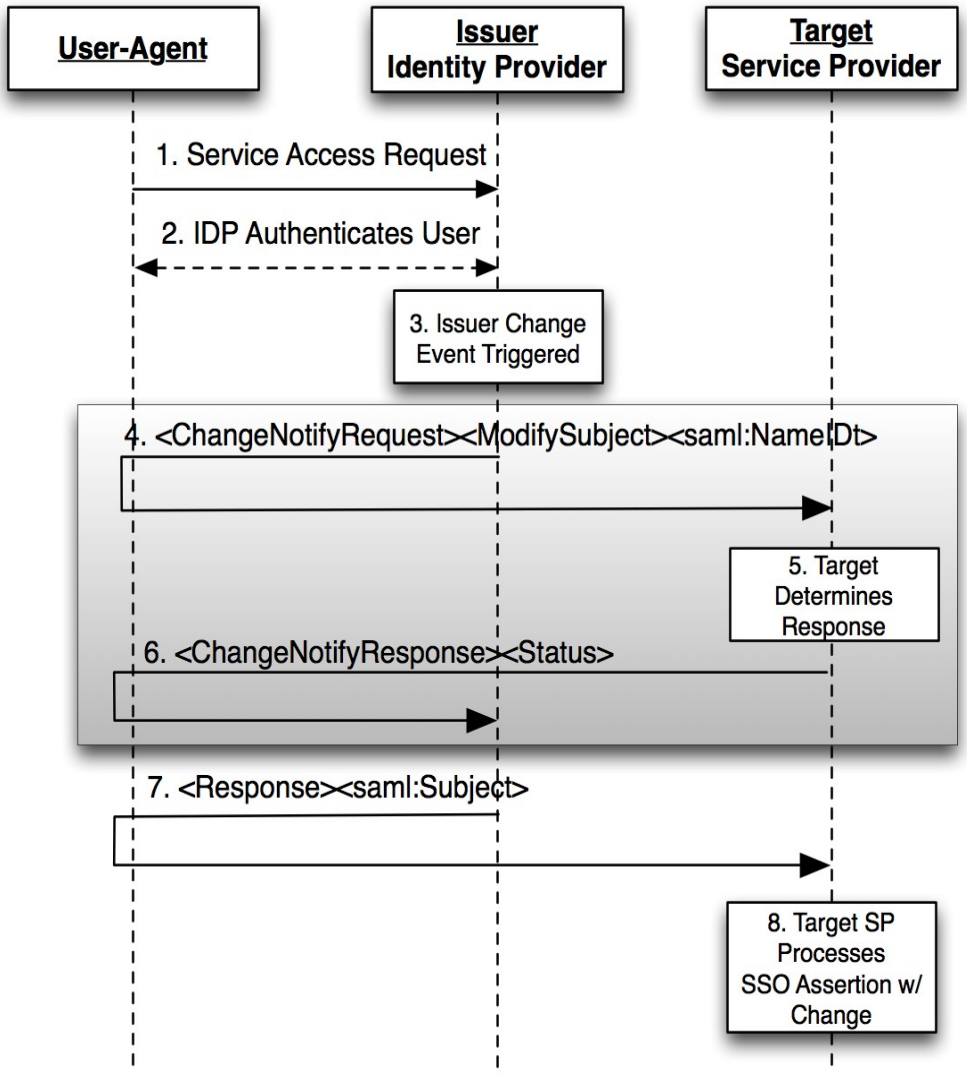


Figure 3: Web SSO Initiated Change at IdP

1) The user makes request for a secure resource at the Notify Issuer (Identity Provider) requiring authentication.

2) The Notify Issuer (Identity Provider) authenticates the user.

3) Notify Issuer (Identity Provider) determines a change notification is required along with an "Unsolicited" Web SSO Profile [SAML2Prof].

4) Notify Issuer (Identity Provider) sends an `<ChangeNotifyRequest>` with a `<ModifySubject>` notification element (which MAY also be a `<NewSubject>` or `<RetireSubject>` element) and SAML Name Identifier, the attribute `protocol` set to `urn:oasis:names:tc:SAML:2.0:notify:protocol:SAML:FrontChannel` with `issuerInitiated` set to `true` to the Notify Target (Service Provider), and a list of available SAML Attributes (except in the case of `<RetireSubject>` notification element).

5) Notify Target (Service Provider) process the request and accepts the notification request.

6) Notify Target sends an `<ChangeNotifyResponse>` to the Notify Issuer, with an accepted list of SAML Attributes.

7) According to the protocol attribute defined in the original `<ChangeNotifyRequest>`, the Notify Issuer completes the transaction by issuing an unsolicited SAML `<Response>` containing a SAML `<Subject>` to the Notify Target endpoint, including the accepted list of SAML `<Attribute>` value assertions.

8) Based on the SAML `<Response>` message, the service provider processes the SSO assertion containing the notified changes.

4.4 Back-Channel Change Notification to a SAML Subject

Figure 4 shows an update being propagated from a Notify Issuer to a Notify Target using a back-channel. They grey-box shows the Change Notify Protocol while the second box shows how the payload for each change MAY be exchanges using the SAML Assertion Query/Request profile [SAML2Prof].

For the purpose of this example, a Notify Issuer or Target MAY be any SAML endpoint such as a Service Provider or Identity Provider.

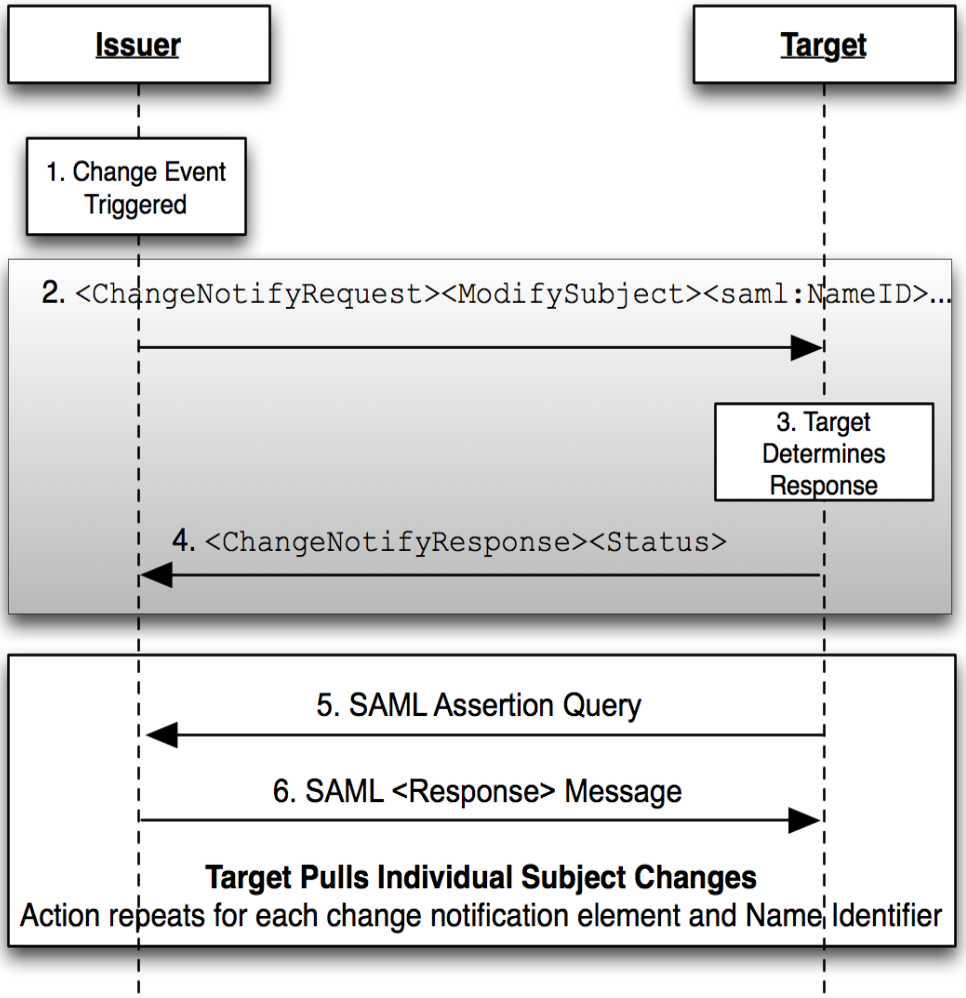


Figure 4: Back-Channel Change Using SAML Assertion Query

1) The Notify Issuer (Identity Provider) determines a change has occurred that SHOULD be shared with a particular target.

2) Notify Issuer sends an `<ChangeNotifyRequest>` with one or more notification elements (`<ModifySubject>` is shown) along with one or more SAML Name Identifiers, the attribute `protocol` set to `urn:oasis:names:tc:SAML:2.0:notify:protocol:SAML:BackChannel` with `issuerInitiated` set to `false` to the Notify Target. For each notification elements, a list of available SAML Attributes (except in the case of `<RetireSubject>` notification element).

3) Notify Target processes the request and accepts the notification request.

4) Notify Target sends an `<ChangeNotifyResponse>` to the Notify Issuer, with an accepted list of SAML Attributes.

5) According to the protocol attribute defined in the original `<ChangeNotifyRequest>`, the Notify Target completes the action phase of the notification by issuing SAML Assertion Queries according to the SAML

Assertion Query Profile [SAML2Prof]. A new query is issued for each <NewSubject> or <ModifySubject> element and name identifier received in the change notify request.

6) As per the SAML Assertion Query/Response Profile, the Notify Issuer responds to each request and returns a SAML <Response> completing the transfer of subject changes described in the original <ChangeNotifyRequest>.

4.5 Profile Description

4.5.1 Change Event Triggers Notifications

An event occurs, either triggered directly by a user, workflow, or backend process, that causes a Notify Issuer to determine there is a change of interest to a particular Notify Target.

4.5.2 <ChangeNotifyRequest> issued to Notify Target

To initiate the profile, the Notify Issuer issues a <ChangeNotifyRequest> message to a target service provider known as a Notify Target. Metadata (as in [SAML2Meta]) MAY be used to determine the location of this endpoint and the bindings supported by the responding provider.

Synchronous Binding (Back-Channel)

The Notify Issuer MAY use a synchronous binding, such as the SOAP binding [SAML2Bind], to send the request directly to the Notify Target provider. The requestor MUST authenticate itself to the other provider, either by signing the <ChangeNotifyRequest> or using any other binding-supported mechanism.

Asynchronous Binding (Front-Channel)

Alternatively, the Notify Issuer MAY (if the principal's user agent is present) use an asynchronous binding, such as the HTTP Redirect, or POST [SAML2Bind] to send the request to the other provider through the user agent.

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 <ChangeNotifyRequest> message MUST be signed.

Each of these bindings provide a RelayState mechanism that the Notify Issuer MAY use to associate the subsequent exchanges with the original request. The Notify Issuer SHOULD reveal as little information as possible in the RelayState value unless the use of profile does not require such privacy measures.

The Notify Issuer server sends a <ChangeNotifyRequest>, and MUST include the attribute protocol specifying the protocol to be used for the action step. The attribute issuerInitiated is defaulted to true. If a different service will issue the action in 4.1.3.4, the Issuer SHALL include the endpoint of the server issuing the SSO assertion.

In the case of <NewSubject>, or <ModifySubject>, the <ChangeNotifyRequest> MUST include one of the notification type elements: <NewSubject>, or <ModifySubject>. Within the notification type element is contained one identifier element <saml:NameID>, <saml:BaseID>, or <saml:EncryptedID>. If the notification element is <NewSubject> or <ModifySubject> transaction, it MAY include one or more SAML Attribute names. No data is transferred.

In the case of <RetireSubject>, the <ChangeNotifyRequest> MUST include one of the notification type elements: <RetireSubject>, MUST includes one identifier element <saml:NameID>, <saml:BaseID>, or <saml:EncryptedID>, MUST one or more SAML Attribute names and MUST NOT include attribute data.

4.5.2.1 Notify Target Determines Action

The Notify Target service provider, on receiving the <ChangeNotifyRequest> determines the internal action it wishes to take regarding the request. The Target evaluates the notification and the protocol attribute included in the request and prepares the server to handle any subsequent action protocol step. This MAY include queuing and/or recording of transaction information such as Subject Identifiers transferred in the <ChangeNotifyRequest> message.

4.5.2.2 Notify Target Responds With <ChangeNotifyResponse>

The Notify Target, the recipient, MUST process the <ChangeNotifyRequest> as defined in section 2.9 Processing Rules. After processing the message or upon encountering an error, the Notify Target MUST issue a <ChangeNotifyResponse> containing an appropriate status code to the requesting provider (Notify Issuer) to complete the protocol exchange.

Synchronous Bindings (Back-Channel)

If the Notify Issuer used a synchronous binding, such as the SOAP binding [SAML2Bind], the response is returned directly to complete the synchronous communication. The responder MUST authenticate itself to the requesting provider, either by signing the <ChangeNotifyResponse> or using any other binding-supported mechanism.

Asynchronous Bindings (Front-Channel)

If the Notify Issuer used an asynchronous binding, such as the HTTP Redirect, or POST bindings [SAML2Bind], then the <ChangeNotifyResponse> is returned through the user agent to the Notify Issuer's endpoint. Metadata (as in [SAML2Meta]) MAY be used to determine the location of the endpoint and the bindings supported by the requesting provider (Notify Issuer). Any binding supported by both entities MAY be used.

If the HTTP Redirect or POST binding is used, then the <ChangeNotifyResponse> message is delivered to the Notify Issuer (requesting provider) in this step.

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 <ChangeNotifyResponse> message MUST be signed.

The exact format of this HTTP response and the subsequent HTTP request to the assertion consumer service is defined by the SAML binding used. Profile-specific rules on the contents of the <ChangeNotifyResponse> are included in Section 2.8 and Section 2.9.

In the case of <NewSubject>, or <ModifySubject>, the <ChangeNotifyResponse> MAY includes a different endpoint to receive the action protocol response by specifying it in the endpoint attribute.

If the Notify Target wishes to take no action due to error, the Target MUST issue a status response of urn:oasis:names:tc:SAML:2.0:status:Responder to indicate an error condition. If the Notify Target wishes to indicate a non-error status result but that no further action is necessary, the responder SHOULD include the attribute actionDeclined with a value of true.

4.5.2.3 Protocol Action

After successful exchange of a <ChangeNotifyRequest> followed by a <ChangeNotifyResponse>, the end points SHALL execute an exchange of information using the appropriate protocol and endpoints negotiated in the message exchange and per the processing rules of section 2.9.

The protocol used is defined by the attribute protocol and the entity initiating the exchange is determined by the attribute issuerInitiated. The protocol action step MAY be delayed until the date specified by the attribute actionAfter, or MAY be declined entirely if the responder sets the attribute actionDeclined to true.

The protocol used to transfer information SHOULD have security measures equivalent to or superior to those specified in this binding to protect the confidentiality and message integrity of data transferred.

5 Conformance

Conformance Notify Issuers and Notify Targets SHOULD implement the Change Notify profile using the HTTP Post, and HTTP redirect bindings.

Informational: Where appropriate, Notify Issuers and Notify Targets SHOULD have agreements in place to define how action protocols will be implemented and used.

A service provider wishing to issue ChangeNotifyRequests, MUST support the protocols necessary to facilitate configured action protocol. An service provider using SAML as an action protocol MUST support SAML Attribute Authority and SAML Authentication Authority functionality for the purpose of fulfilling SAML action steps as described in the profile.

A Notify Issuer can claim to support Change Notify Protocol if it can issue `<ChangeNotifyRequest>`s, respond to `<ChangeNotifyResponse>`s, and can support the use of at least ONE action protocol to facilitate transfer of change data to the Notify Target's designated protocol endpoint.

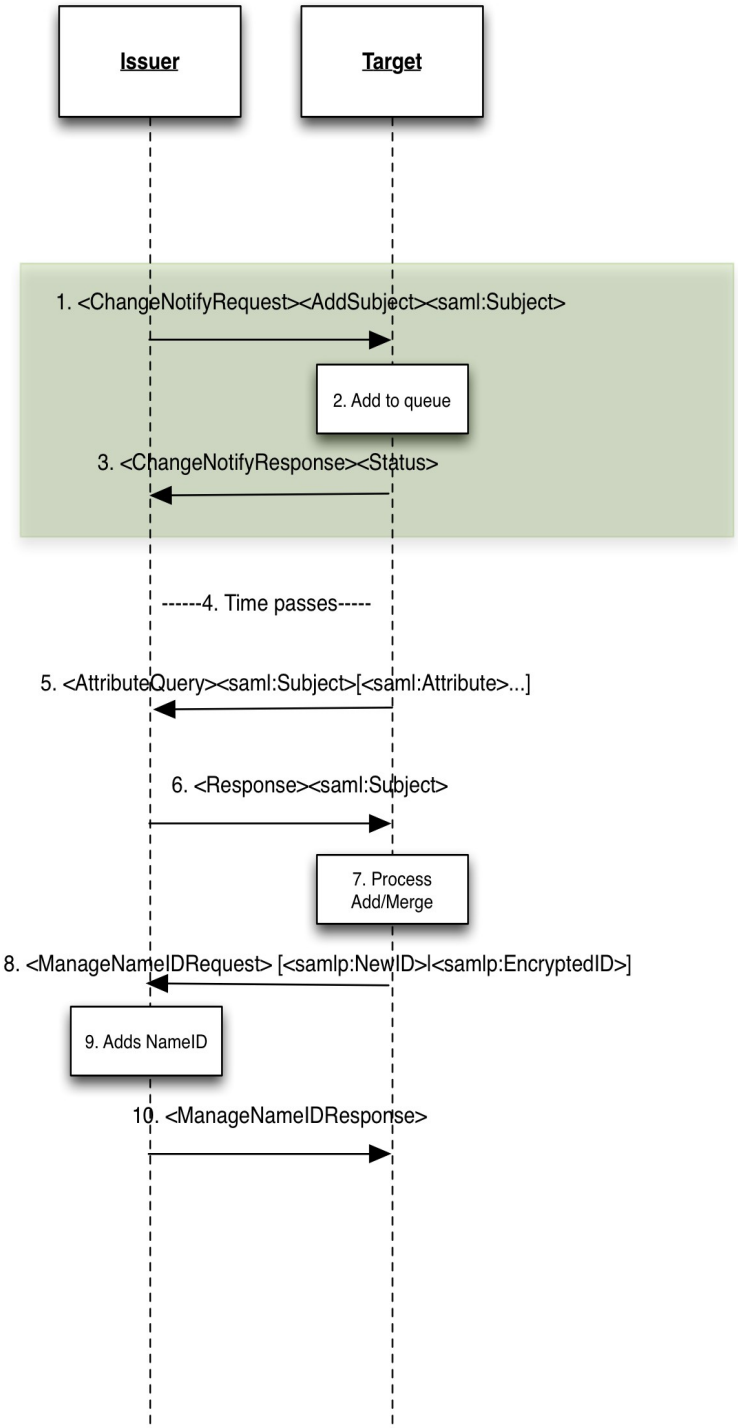
A Notify Target can claim to support Change Notify Protocol if it can respond to `<ChangeNotifyRequest>s`, issue `<ChangeNotifyResponse>s`, and can support the use of at least ONE action protocol to support the transfer of change data from the Notify Issuer's designated protocol endpoint.

A Notify Issuer and Notify Target claiming to support Change Notify Protocol in the front-channel MUST also be able to support the Web SSO Profile [SAML2Prof] bi-directionally.

Appendix A. Use Cases

An issuer notifies a target of new information that is available. The target MAY then request the data via either an AttributeQuery or an AuthnRequest in the case of the browser profile.

A.1. Offline/Backchannel Mode*:



1. The issuer notifies the target of some updated information regarding a particular subject. In this case an add subject indicates that the issuer believes this subject is new to the target (which may or may not be true).

The assertion only includes the issuers nameidentifier. The issuer can indicate multiple requests in the same message. The issuer MAY indicate what attributes are available in the message.

2. The target receives the request and either adds it to its queue processing (immediate or delayed). The target MAY also choose to ignore the request, but MUST acknowledge the receipt of the request (step 3).

3. The target acknowledges the request. The target MAY indicate OK, or indicate declined. A response of OK does not oblige the target to do anything further.

4. The target MAY optionally delay processing (the process is asynchronous)

5. The target issues an attributeQuery for each nameidentifier supplied by the issuer. If no attributes are named, the attributes provided SHALL be the ones indicated in step 1, or all attributes as per the normal AttributeQuery processing. OR, if arranged by prior agreement, the target MAY use a different protocol to effect transfer (e.g SPML, OpenID, etc).

6. Issuer responds with the attributes requested.

7. The target MAY optionally update the issuer with its local name identifier depending on the relationship between issuer and target.

Note: for the purpose of this profile, issuer or target end-points can refer to either SP or IDP. E.g. An SP notifying an IDP of a new user transfer, or an IDP notifying an SP of a new user (e.g. Employee in an enterprise IDP).

A.2. Browser/Synchronous Profile

In the synchronous mode, information transfer is accomplished via browser SSO. This MAY be useful in cases where SSO transfer of context is desirable.

1. The issuer notifies the target of some updated information regarding a particular subject. In this case an `<NewSubject>` indicates that the issuer believes this subject is new to the target (which may or may not be true). The assertion only includes the issuer's name identifier. The issuer can indicate multiple requests in the same message. The issuer MAY indicate what attributes are available in the message.

2. The target receives the request and determines what it wants to do (e.g. process as add, modify, or ignore). The target MAY also choose to ignore the request, but MUST acknowledge the receipt of the request by issuing a `<ChangeNotifyResponse>` response.

Appendix B. Acknowledgments

The editor would like to acknowledge the contributions of the OASIS Security Services (SAML) Technical Committee, whose voting members at the time of publication were:

- Rob Philpott, EMC Corporation
- Bob Morgan, Internet2
- Scott Cantor, Internet2
- Nathan Klingenstein, Internet2
- Chad La Joie, Internet2
- Thomas Hardjono, M.I.T.
- Frederick Hirsch, Nokia Corporation
- Thinh Nguyenphu, Nokia Siemens Networks GmbH & Co. KG
- Ari Kermaier, Oracle Corporation
- Hal Lockhart, Oracle Corporation
- Emily Xu, Oracle Corporation
- Anil Saldhana, Red Hat
- David Staggs, Veterans Health Administration

The editor would also like to acknowledge the contribution of an earlier draft from NSN entitled: “SAML V2.0Attributes Management Protocol Version 1.0 Working Draft 06 November 2009“, upon which this document attempts to incorporate supporting requirements from.

Appendix C. Revision History

Document ID	Date	Committer	Comment
sstc-saml2-notify-protocol-01	07/19/10	Phil Hunt Thinh Nguyenphu	Initial draft
sstc-saml2-notify-protocol-02	09/17/10	Phil Hunt Thinh Nguyenphu	Editorial clean ups, saml:Subject changed to NameID etc
sstc-saml2-notify-protocol-03	10/01/10	Thinh Nguyenphu Phil Hunt	Updates to Profiles adding two overview flows
sstc-saml2-notify-protocol-04	10/21/10	Phil Hunt Thinh Nguyenphu	Removed ActionProtocol Element Completed profiles
sstc-saml2-notify-protocol-v1.0-wd05	5 May 2011	Thinh Nguyenphu	Editorial cleanup based on 30 days public review comments from Chapman Martin