SAML XRI Authentication Service Profiles

Draft 0.12, 5 June 2008

Document identifier:

draft-xri-saml-profiles-01

Location:

[tbd]

Editors:

Peter Davis, NeuStar, Inc.

Contributors:

[TBD]

Abstract:

This document specifies SAML V2.0 profiles when using XRI identifiers.

Status:

[draft prologue]

[IPR prologue]

Table of Contents

The table of contents is empty because none of the paragraph styles selected in the Document Inspector are used in the document.

1 Introduction

This specification describes additional profiles and Authentication Contexts of [SAML2], as prescribed by [SAML2Prof] and [SAML2AC]. These additional profiles enable SAML authorities and requestors to perform Authority discovery by means of XRI resolution (and defines an explicit service type for XRD documents described in [XRIRes] and further in [GSSspec]). A second profile defines mechanisms to obtain SAML V2 metadata for a SAML provider, in order to properly convey messages between SAML provider parties, and the user. New Authentication Contexts are introduced, which supplement existing contexts with visual element requirements on the Authentication Authority, to aid in principals verification of the providers identity.

2 Notation

The keywords "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this specification are to be interpreted in this specification and all of the SAML V2.0 specifications as described in IETF RFC 2119:

> *…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.

This specification uses the following typographical conventions in text: `<SAMLElement>`, `<ns:ForeignElement>`, `XMLAttribute`, **`Datatype`**, `OtherKeyword`. In some cases, angle brackets are used to indicate non-terminals, rather than XML elements; the intent will be clear from the context.

3 Application of XRIs with Security Assertion Markup Language (SAML)

This document specifies new profiles and extensions to the OASIS Security Services Technical Committee's Security Assertion Markup Language revision 2.0 [SAML2].

The new capabilities include:

- Authentication Service Provider Discovery (aka IDP Discovery)
- Metadata publication and retrieval
- Authentication context for visual verification of provider

A hypothetical (non-normative) example is provided in A, which takes a reader through some detailed steps and possible deployment considerations.

3.1 Identifiers for XRI aware systems

While this specification will make use of and references the XRI URI scheme, implementations following the profiles laid out here SHOULD take care to maintain support for other URI schemes. In addition, the various identifiers may be any of XRI, URI, IRI, or mixed. This specification makes use of XRI-specific capabilities to:

- Resolve Authentication service endpoints and providers
- Provider SAML metadata discovery
- Used as identifiers: providerID (and `<Issuer>)` of service provider

The identifiers used in these (and other SAML-based systems) SHOULD remain opaque to the application.

4 XRI Resolution for Authentication Service Discovery Profile

[SAML2prof] suggests a cookie based mechanism which enables service providers to inspect user agent requests, and determine if some transparent means of routing an authentication request is possible. Other authentication request mechanisms exist elsewhere which define their own discovery mechanisms. This XRI-based discovery profile is defined to provide a comparable set of capabilities by providing a more portable means of discovering authentication and attribute services, by way of an XRI.

This discovery profile does not specify explicitly a particular authentication service profile, nor does it preclude it's use for other XRI-aware authentication services, however it is STRONGLY RECOMMENDED that implementations which support this discovery profile also support the profile defined in Section 5.

Support of this profile requires implementers to support XRI resolution as defined in [XRIres] and XRI syntax as defined in [XRIsyntax].

4.1 Required Profile Registration Information

Identification: xri://+i-service*(+authn)*(+discovery)*($v*1.0)

Contact information: someone@oasis

Description: Given below.

4.2 Authentication Service Resolution Profile

Service resolution is broadly defined in [XRIres] as a means of discovering services available for an XRI authority. This profile defines specific processing requirements for services with a base service type xri://+i-service*(+authn)

In order for the service provider to determine possible authentication service endpoints for a principal wielding an i-name, service providers shall obtain the i-name (or i-number) via some mechanism, which is out of scope for this specification.

4.3 Handling Multiple Discovery Mechanisms

Principals may advertise, using various means, support for multiple authentication service discovery mechanisms, such as those defined in [SAML2Prof] and [Yadis]. Service Providers may determine which of the possible discovery mechanisms (and thus which SSO mechanisms) appropriate to fulfill their requirements of a given context. Principals SHOULD be informed, should multiple mechanisms be offered, about selection criteria used by the service provider, or otherwise allowed to select the preferred authentication for the present interaction.

4.4 Extensibility of Authentication Protocol Descriptors

XRI aware authentication services MUST define an authentication protocol identifier that shall be used as the <type> value of the XRD.

This identifier MUST be an XRI, and it MUST be rooted in the +authenticationService namespace xri://+i-service*(+authn).

Section 5 defines xri://+i-service*(+authn)*(+saml)*($v*1.0)

4.5 Resolution of the XRI

While resolving XRIs, Service Providers MUST resolve the fully qualified authority part of the XRI as defined in [XRIres] 'authority-part' ABNF production (section xxx.xxx). The resolution authority(s) MUST be contacted using [SSL3] or [TLS1], and resolving clients MUST verify the subject of the x509v3 certificate and ensure it corresponds to the authority with which it is contacting. If any portion of the resolution chain cannot be completed using TLS/SSL, the service provider SHOULD discard the i-name, and appropriately inform the principal that service resolution could not be completed. XRI Authorities MAY use client authentication mechanisms, including mutual TLS v1.0 to authenticate and/or authorize resolution requests.

Resolving clients MAY express a preference for trusted resolution as defined in [XRIres], but MUST support basic resolution. XRI Authorities that provide service endpoint resolution for authentication services SHOULD support the Trusted Resolution mechanisms defined in [XRIres].

Resolving clients MUST NOT avail themselves of look ahead and proxied resolution mechanisms defined in [XRIres] while handling authentication service resolution, in order to ensure the integrity of the resolution results from the authority.

Service providers MUST process the resulting Extensible Resource Descriptor (XRDS) XML document obtained after completed resolution as defined in [XRIres] by the Service Endpoint Selection protocols prescribed there.

Service providers MAY select any authentication service which they determine best suites their requirements regardless of priority ordering.

4.6 XRD Example

The XRDS that is returned through resolution provides details for each service: a service type, one or more network endpoint(s), and an identifier for the provider. Service type specifications MUST provide guidance in the interpretation of the elements in the `Service` portion of the XRD

An example XRD fragment follows:

```
…
<Query>*alice</Query>
…
<Service priority="1">
        <Type>
        xri://+i-service*(+authn)*(+saml)*($v*1.0)
        </Type>
        <URI priority="1">https://idp.example.biz/login</URI>
        <URI priority="2">https://idp2.example.biz/login</URI>
        <URI priority="100">http://idp3.example.biz/login</URI>
        <ProviderID>
              xri://@example*identityProvider/SAMLAuthority
        </ProviderID>
        [...]
</Service>
<service priority=”10”>
  <Type>urn:oasis:names:tc:SAML:2.0:profiles:SSO:browser</type>
  <ProviderID>
              http://authN.example.biz/SAML/2/sso
  </ProviderID>
</service>
<Service priority="20">
        <Type>http://openID.org/login</Type>
        <URI priority="1">https://login.example2.biz/login</URI>
        <ProviderID>http://openid.example2.biz/openid</ProviderID>
</Service>
<Service priority="30">
        <Type>http://lid.netmesh.org/lid</Type>
        <URI priority="1">https://lid.example3.biz/login</URI>
        <ProviderID>http://my.lidexample.biz/lid</ProviderID>
</Service>
```

[ed.note: this example should true-up with the lid and openID service type declarations]

Subsequent actions taken by the resolving client are Service/Type specific. Section 5 specifies one such subsequent interaction.

4.7 Security Considerations

The use of XRI's for authentication service discovery introduces a new potential correlation handle of the principal. Authentication service providers should carefully consider the risks associated with this shared identifier.

One suggested remedy is allow the principal to only supply the XRI of the authentication service provider (eg: @IdentityProvider), and not their personal i-name.

5 XRI SAML Browser SSO profile

This profile is derived from the SAML2 “Web Browser SSO Profile” as defined in [SAMLprof]. Some of the material from that profile will be repeated here for the convenience of the reader. In the case of any inconsistencies or ambiguity, the SAML profile specification shall prevail (except for the extensions of the profile itself).

5.1 Required Registration Information

Identification: xri://+i-service*(+authn)*(+saml)*($v*1.0)

Contact information: someone@oasis

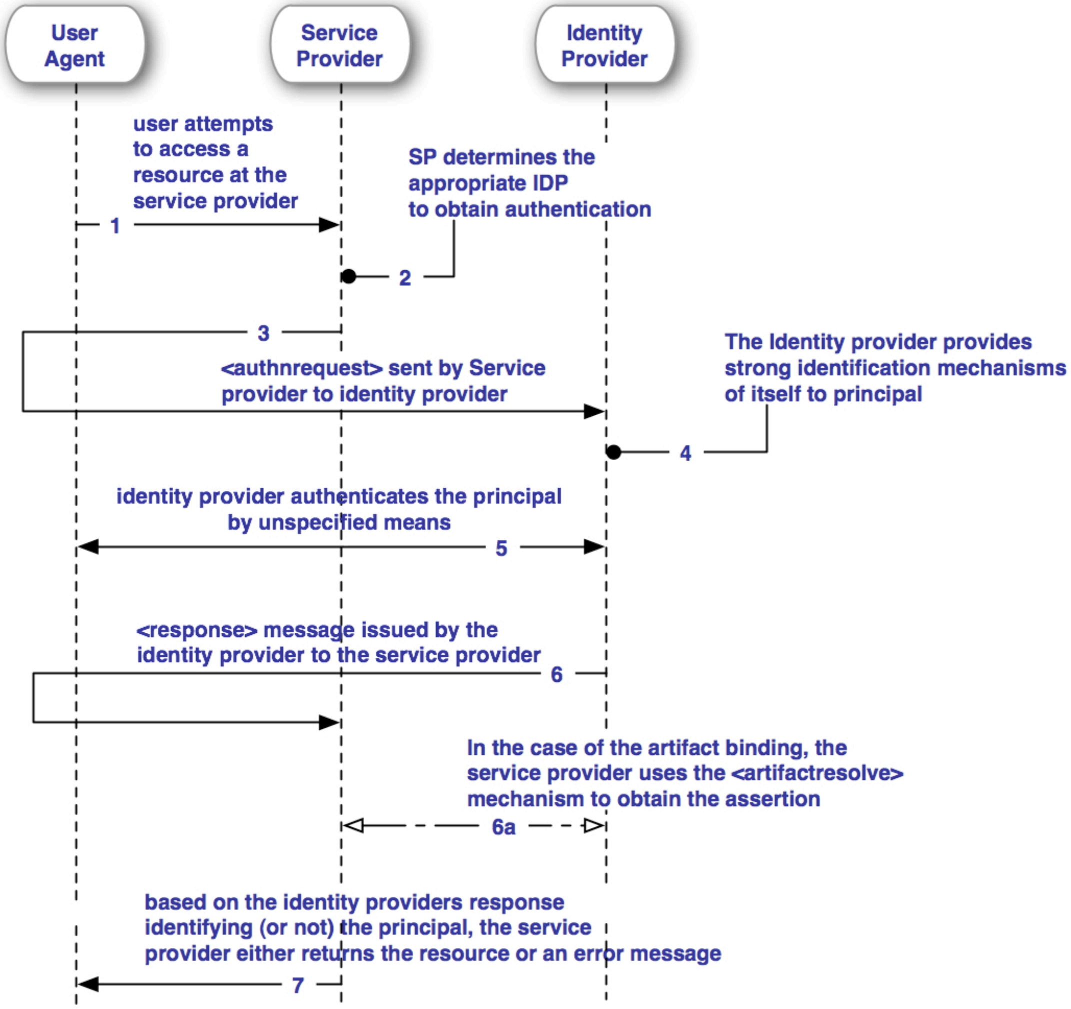
SAML Confirmation Method Identifiers: The SAML V2.0 "bearer" confirmation method identifier, urn:oasis:names:tc:SAML:2.0:cm:bearer, is used by this profile.

Description: Given below.

Updates: Extends the SAML2 “Web Browser SSO Profile”

5.2 Profile Overview

The following diagram illustrates the basic messaging components of this profile..


User Agent
Service Provider
Identity Provider
user attempts to access a resource at the service provider
1
SP determines the appropriate IDP to obtain authentication
2
3
<authnrequest> sent by Service provider to identity provider
The Identity provider provides strong identification mechanisms of itself to principal
4
identity provider authenticates the principal by unspecified means
5
<response> message issued by the identity provider to the service provider
6
In the case of the artifact binding, the service provider uses the <artifactresolve> mechanism to obtain the assertion
6a
based on the identity providers response identifying (or not) the principal, the service provider either returns the resource or an error message
7

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 Authentication service
In step 2, the service provider obtains the location of an endpoint at an authentication service for the authentication request protocol that supports its preferred binding. The means by which this is accomplished is implementation-dependent. The service provider SHOULD use the XRI Resolution Authentication service Discovery Profile defined in section 3.1 and MAY use the SAML authentication service discovery profile described in Section 4.3.

3. <AuthnRequest> issued by Service Provider to Authentication service
In step 3, the service provider issues an <AuthnRequest> message to be delivered by the user agent to the authentication service. Either the HTTP Redirect, HTTP POST, or HTTP Artifact binding can be used to transfer the message to the authentication service through the user agent.

4. Principal identifier the Authentication service
The authenticity of the parties in these 'front-channel' message exchanges require some verification, therefore, the IDP MUST provide mechanisms for allowing the IDP to assert their identity to the principal.. the Identity Assurance Mechanisms MAY be employed for this, or other mechanisms must be used.

5. Authentication service identifies Principal
In step 4, the principal is identified by the authentication service 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.

6. Authentication service issues <Response> to Service Provider
In step 5, the authentication service issues a <Response> message to be delivered by the user agent to the service provider. Either the HTTP POST, or HTTP Artifact binding can 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. The HTTP Redirect binding MUST NOT be used, as the response will typically exceed the URL length permitted by most user agents.

6.a Service Provider resolves Artifact
the artifact provided by the IDP to the Service Provider is obtained using the artifact resolution profile

7. Service Provider grants or denies access to Principal
Having received the response from the authentication service, the service provider can respond to the principal's user agent with its own error, or can establish its own security context for the principal and return the requested resource. Note that an authentication service can initiate this profile at step 5 and issue a <Response> message to a service provider without the preceding steps.

5.3 Profile Description

This profile follows closely the Web Browser SSO Profile defined in [SAML2prof] with the following variations:

- The service provider shall always initiate the request/response sequence. Authentication service initiated messages defined in [SAML2prof] remain unaltered by this profile.
- This profile MAY be initiated with any Identity Provider Discovery mechanism, but it is expected that XRI resolution discovery mechanisms defined in section 4 of this specification shall be used to determine the appropriate authentication service for the principal.
- The service provider MUST include one of the authentication contexts defined in Section 6 of this specification [xris of profiles] in the `<RequestedAuthnContext>` element of the

request.

- The service provider MAY include a <Subject> element in the request that names the actual iName about which it wishes to receive an assertion. This element MUST NOT contain any <SubjectConfirmation> elements, and MUST be an XRI. If the authentication service does not recognize the principal as that identity, then it MUST respond with a <Response> message containing an error status and no assertions.
-
- The attribute `ProviderName` MUST be provided, and the Authentication service MUST display the value of this attribute to the principal, unless the service provider indicates that interaction is prohibited by the `<IsPassive> element in the authentication request.`

5.4 Protocol Bindings Requirements

This profile does not constrain implementations to a particular set of protocol bindings defined in [SAML2core].

5.5 Metadata Considerations

This profile requires the use of the additional authentication context defined in section 7.1 below. Thus the authentication MUST demonstrate support for this authentication context in its metadata instance document.

Implementations supporting this profile must indicate this in their metadata instance document.

[SAML2Meta] defines possible URI oriented mechanisms for Metadata retrieval and consumption. This specification supplies a means by which an XRI aware implementations may identify and obtain metadata for the authentication service participants.

Providers are free to obtain and derive metadata by any other means, including as defined by [SAML2Meta] as well as out of band mechanisms.

There are two variants for establishing the provider for which metadata is required, one flow for the service provider, and the other for the authentication service provider. The variation occurs only in the manner by which the provider obtains the (providerID) of the other party involved in the authentication protocol sequence.

5.6 Obtaining the <providerID>

5.6.1 Service Provider Initiated Metadata Discovery

In the service provider initiated sequence, the SP performs XRI-based authentication service resolution as in section 4.2, which includes the providerID of the authentication service via the XRDS returned in that resolution step (the <providerID> element of the service).

5.6.2 Authentication Service Initiated Metadata Discovery

If the authentication service, being the recipient of an authentication request by an SP, derives the (providerID) of the request from the request message itself via the [need the true element name here]. Otherwise, the remaining steps proceed in an identical fashion.

Figure 1 complete example message flow incorporates these steps by way of example.

5.7 Resolving XRI-based Provider Metadata

Upon obtaining either an i-name or i-number of a provider, whose metadata is required to process the SAML message, the metadata relying party resolves that XRI. Implementations SHOULD make use of trusted resolution, as defined in [XRIres].

The resolver then processes the resulting XRDS, and selects the service element whose type is:
`xri://+i-service*(+metadata)*(+saml)*($v*2.0)`

The <URI> element presented in this resolution step MUST be the URI endpoint by which the relying party may obtain the metadata instance document for the provider identified by the XRI initially resolved.

5.8 Instance Caching Considerations

There are several caching instruction sets to manage within these resolution steps:

- caching of the XRDS document
- caching of the metadata document
- HTTP(S) caching directives during retrieval of the XRDS document

Implementations MUST adhere to the strictest (that is, earliest) of all these datetime values provided for caching directives. Thus, the earliest time MUST ALWAYS preside over any other cache directive.

5.9 Security Considerations

No new security considerations are introduced by this service.

As this profile allows unintroduced parties to potentially engage in SAML exchanges, SAML entities should carefully consider honoring requests or responses from others with whom they have not previously interacted.

6 SAML Authentication Context Extension

[SAML2AC] provides identifiers of authentication mechanisms that can be conveyed in both SAML requests and responses. This specification adds an additional parameter to the context that obligates the authentication authority to provide visual clues to the principal that the principal has reached the present provider they have interacted with in the past (aka: a known authentication authority).

Personalized Multifactor challenge adds additional 'personal' questions to the authentication mechanism, in addition to the mechanisms that are defined in [SAML2AC].

The Visual Provider Verification mechanism adds personalized visual (usually graphical) elements to the resource located at `SingleSignOnServiceURL`.

6.1 Visual Provider Verification of the SAML Authority

Class Identifier: xri://+i-service*(+authn)*(+context)*(+vvAuthority)*($v*1.0)
In order to mitigate phishing attacks upon authentication services' principals, authentication service providers can take several actions with respect to authenticating themselves to the user.

Authentication services MAY manufacture principal specific provider authentication mechanisms, which provide the principal with visual clues of the authenticity of the page and provider that rendered the login page to the principal.

Authentication services MUST provide the principal with at least one mechanism that allows that principal to customize the page presented for consumption of authentication mechanisms, such as a login page to collect username and password, when this context is enumerated in the service providers request. Two mechanisms are provided in this specification.

The following authentication contexts defined in [SAML2AC] SHOULD make use of this context, and MAY choose to use this context under other circumstances:

- `urn:oasis:names:tc:SAML:2.0:ac:classes:Password`
- `urn:oasis:names:tc:SAML:2.0:ac:classes:PasswordProtectedTransport`
- `urn:oasis:names:tc:SAML:2.0:ac:classes:TimeSyncToken`

These mechanisms require cross-session persistence, in order to properly recognize the user agent. One such persistence mechanism is the use of HTTP Cookies. Other mechanisms may be employed as well.

In the event that a user agent is 'unrecognized' in this sense, Authentication services MUST NOT present a resource that allows the collection of security tokens for the cited contexts above. Authentication services MUST instruct the principal manually navigate to the Authentication service to authenticate.

Authentication services MUST retain the authentication request supplied by the Service Provider, and be able to correlate a manual request for the authentication page with the original <Request>.

6.2 Identity Provider Implementation Guidance

This mechanism provides a means for the principal, while administering their account with the Authentication service, to select questions or other visual clues either from a predefined list or freeform, and provide corresponding answers. When using this mechanism, Authentication services MUST supplement the requested authentication mechanism with user specified questions or other personalization supplied by the principal, and must correlated the principal answers to said questions. These mechanisms should be selected to the subscriber at enrollment time, and be at the discretion of the principal.

Principals must be clearly informed as to the purposes this personalization serves, in order to ensure they can identify its mis-use. If an IDP can identify a principal without direct user interaction (for example, a browser cookie), it MUST authenticate itself to the user by displaying the personalized login page containing the user's visual verification artifact(s).

If a service cannot identify the principal without user interaction, it MUST NOT display the personalized login page. Instead it MUST display an instruction page directing the user how to safely login.

Because this instruction page is expected to be a primary point of attack for phishing sites attempting to trick a user into entering their credentials, the following design criteria for instruction pages are STRONGLY RECOMMENDED.

7 References

[tbd]

saml2

saml2prof

saml2ac

SAML2Meta

xrires

gssspec

yadis

Identifiers

Discovery: xri://+i-service*(+authn)*(+discovery)*($v*1.0)

SAML XRI Authn: xri://+i-service*(+authn)*(+saml)*($v*1.0)

SAML MD: `xri://+i-service*(+metadata)*(+saml)*($v*2.0)`

`vvauthN:` xri://+i-service*(+authn)*(+context)*(+vvAuthority)*($v*1.0)

Appendix AExamples (non normative)

Provides an overview of the entire set of messages for an example deployment of these profiles utilizing the profiles laid out in this specification:

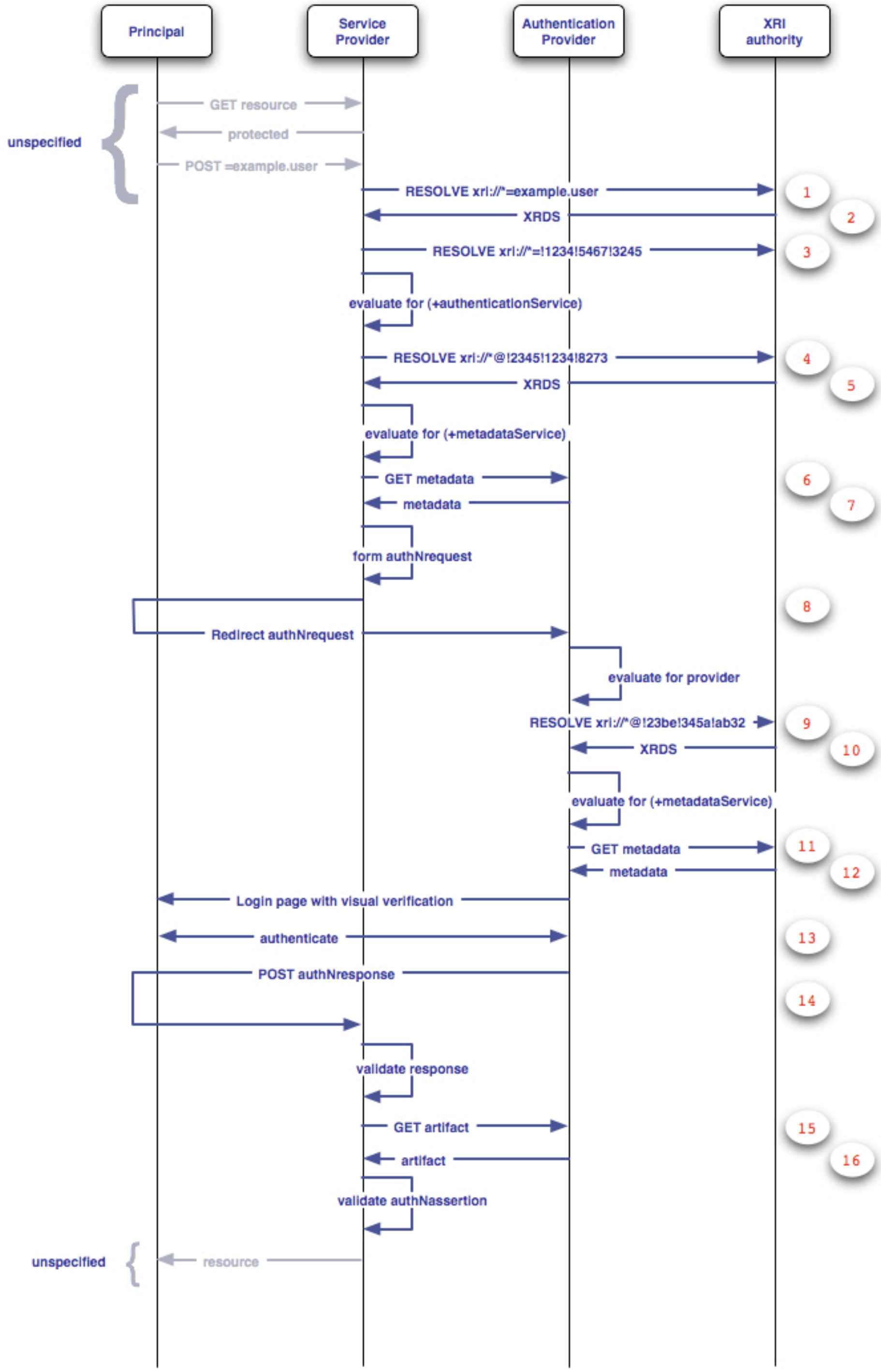


Figure 1 complete example message flow

.1Message Examples

The following messages correspond to the numbered messages of Figure 1 complete example message flow above.

```
1.RESOLVE xri://*=example.user
```

2.

```
GET /=example.user HTTP/1.1
Host: authority.example.biz
If-Modified-Since: Fri, 24 March 2006 19:43:32 GMT
Accept: application/xrid+xml
 (other headers)
```

2. XRDS response from XRI authority

```
200 OK HTTP/1.1
Content-Type: application/
Expires: Fri, 7 Nov
<other HTTP headers>

<XRIDescriptors xmlns=”…”>
<XRIDescriptor xmlns=”…”>
<Resolved>*example.user</
<Authority>
<URI>http://xri.example.com/</URI>
</Authority>
<Service>

</Service>
</XRIDescriptor>
</XRIDescriptors>
```

3. RESOLVE i-Number: xri://*=!1234!5467!3245

4. RESOLVE i-Number xri://*@!1234!8273

5. XRDS Response

```
<Service priority="1">
        <Type> xri://+i-service*(+metadata)*(+saml)*($v*2.0)</Type>
        <URI priority="1">https://idp.example.biz/samlmetadata</URI>
        <URI priority="2">https://idp2.example.biz/samlmetadata</URI>
        <ProviderID>
              xri://@example*identityProvider/SAMLAuthority
        </ProviderID>
        [...]
</Service>
```

6. GET Metadata for identity provider

```
 GET /samlmetadata HTTP/1.1
 Host:idp.example.biz
 If-Modified-Since: Fri, 24 March 2006 19:43:32 GMT
[other headers]
```

7. Metadata response

```
HTTP/1.1 200 OK
Date: 21 Jan 2006 07:00:49 GMT
Content-Type: text/html; charset=iso-8859-1

HTTP/1.1 200 OK
<?xml version="1.0" encoding="UTF-8"?>
<EntityDescriptor xmlns="urn:oasis:names:tc:SAML:2.0:metadata"
     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
entityID="xri://@example*identityProvider/SAMLAuthority">
    <Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
       <!—signature info omitted -->
    </Signature>

    <IDPSSODescriptor WantAuthnRequestsSigned="true"
        validUntil="2006-05-01T00:00:00Z"
        protocolSupportEnumeration="urn:oasis:names:tc:SAML:
2.0:protocol xri://+i-service*(+authn)*(+saml)*($v*1.0)">
        <ArtifactResolutionService index="1"
            Binding="urn:oasis:names:tc:SAML:
2.0:bindings:SOAP"
            Location="https://example.biz/SAML/artifact"/>
        <SingleLogoutService
            Binding="urn:oasis:names:tc:SAML:
2.0:bindings:HTTP-POST"
            Location="https://example.biz/SAML/logout"/>
        <SingleLogoutService
            Binding="urn:oasis:names:tc:SAML:
2.0:bindings:Redirect"
            Location="https://example.biz/SAML/logout"/>
        <SingleSignOnService
            Binding="urn:oasis:names:tc:SAML:
2.0:bindings:HTTP-POST"
            Location="https://example.biz/SAML/authNrequest"/>
        <SingleSignOnService
            Binding="urn:oasis:names:tc:SAML:
2.0:bindings:Redirect"
            Location="https://example.biz/SAML/authNrequest"/>

    </IDPSSODescriptor>
    <Organization>

        <OrganizationName xml:lang="en-US">Example i-Broker</
OrganizationName>
        <OrganizationDisplayName xml:lang="en-US">Example i-
Broker, Inc.</OrganizationDisplayName>
        <OrganizationURL xml:lang="en-US">http://example.biz/
</OrganizationURL>
    </Organization>

</EntityDescriptor>
```

8. Redirect authNrequest to Authentication service

The original xml message:

```
<?xml version="1.0" encoding="UTF-8"?>
<AuthnRequest xmlns="urn:oasis:names:tc:SAML:2.0:protocol"
     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
     xsi:schemaLocation="urn:oasis:names:tc:SAML:2.0:protocol
    file:/Users/peterdavis/projects/oasis/sstc/saml-2.0-os/
saml-schema-protocol-2.0.xsd"
    AssertionConsumerServiceIndex="1"
    IsPassive="false"
    ProtocolBinding="urn:oasis:names:tc:SAML:
2.0:bindings:HTTP-Artifact"
    ProviderName="Example Service Provider"
ID="_867342873297843297843297823 4"
    Version="2.0" IssueInstant="2006-03-16T00:00:00Z">
    <Issuer xmlns="urn:oasis:names:tc:SAML:
2.0:assertion">xri://*@!23be!345a!ab32</Issuer>
    <Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
       <!–signature here -->
    </Signature>
    <!-- Optional Subject identification -->
    <Subject xmlns="urn:oasis:names:tc:SAML:2.0:assertion">
        <NameID SPNameQualifier="xri://*@example.sp">xri://
*=example.user</NameID>
    </Subject>
    <NameIDPolicy AllowCreate="true"/>

    <RequestedAuthnContext>
        <AuthnContextClassRef>
        xri://+i-
service*(+authn)*(+context)*(+vvAuthority)*($v*1.0)
        </AuthnContextClassRef>
        <AuthnContextClassRef xmlns="urn:oasis:names:tc:SAML:
2.0:assertion">urn:oasis:names:tc:SAML:
2.0:ac:classes:Password</AuthnContextClassRef>
    </RequestedAuthnContext>
</AuthnRequest>
```

The resulting base64 encoded request for inclusion in a hidden HTML Form control applied to the browser post profile:

```
HTTP/1.1 200 OK
Date: 21 Jan 2006 07:00:49 GMT
Content-Type: text/html; charset=iso-8859-1
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
<body onload="document.forms[0].submit()">
<noscript>
<p>
<strong>Note:</strong> Since your browser does not support JavaScript,
you must press the Continue button once to proceed.
</p>
</noscript>
```

```
<form action="https://example.biz/SAML/authNrequest"
method="post">
<div>
<input type="hidden" name="RelayState"
value="0043bfc1bc45110dae17004005b13a2b"/>
<input type="hidden" name="SAMLRequest"
value="
PD94bWwgdmVyc2lvbj0iMS4wIiBlbmNvZGluZz0iVVRGLTgiPz4KPEF1dGhuUmVxdWVzdCB4bWxu
cz0idXJuOm9hc2lzOm5hbWVzOnRjOlNBTUw6Mi4wOnByb3RvY29sIgogICAgIHhtbG5zOnhzaT0i
aHR0cDovL3d3dy53My5vcmcvMjAwMS9YTUxTY2hlbWEtaW5zdGFuY2UiCiAgICAgeHNpOnNjaGVt
YUxvY2F0aW9uPSJ1cm46b2FzaXM6bmFtZXM6dGM6U0FNTDoyLjA6cHJvdG9jb2wgCiAgICBmaWxl
Oi9Vc2Vycy9wZXRlcmRhdmlzL3Byb2plY3RzL29hc2lzL3NzdGMvc2FtbC0yLjAtb3Mvc2FtbC1z
Y2hlbWEtcHJvdG9jb2wtMi4wLnhzZCIgCiAgICBJRD0iZm9vYmFyIiAKICAgIFZlcnNpb249IjIu
MCIgSXNzdWVJbnN0YW50PSIyMDA2LTAzLTE2VDAwOjAwOjAwWiI+CiAgICA8SXNzdWVyIHhtbG5z
PSJ1cm46b2FzaXM6bmFtZXM6dGM6U0FNTDoyLjA6YXNzZXJ0aW9uIj54cmk6Ly8qQCEyM2JlITM0
NWEhYWIzMjwvSXNzdWVyPgogICAgPFNpZ25hdHVyZSB4bWxucz0iaHR0cDovL3d3dy53My5vcmcv
MjAwMC8wOS94bWxkc2lnIyI+CiAgICAgICAgPFNpZ25lZEluZm8+CiAgICAgICAgICAgIDxDYW5v
bmljYWxpemF0aW9uTWV0aG9kIEFsZ29yaXRobT0iIj48L0Nhbm9uaWNhbGl6YXRpb25NZXRob2Q+
CiAgICAgICAgICAgIDxTaWduYXR1cmVNZXRob2QgQWxnb3JpdGhtPSIiPjwvU2lnbmF0dXJlTWV0
aG9kPgogICAgICAgICAgICA8UmVmZXJlbmNlPgogICAgICAgICAgICAgICAgPERpZ2VzdE1ldGhv
ZCBBbGdvcml0aG09IiI+PC9EaWdlc3RNZXRob2Q+CiAgICAgICAgICAgICAgICA8RGlnZXN0VmFs
dWU+PC9EaWdlc3RWYWx1ZT4KICAgICAgICAgICAgPC9SZWZlcmVuY2U+CiAgICAgICAgPC9TaWdu
ZWRJbmZvPgogICAgICAgIDxTaWduYXR1cmVWYWx1ZT48L1NpZ25hdHVyZVZhbHVlPgogICAgPC9T
aWduYXR1cmU+CiAgICA8IS0tIE9wdGlvbmFsIFN1YmplY3QgaWRlbnRpZmljYXRpb24gLS0+CiAg
ICA8U3ViamVjdCB4bWxucz0idXJuOm9hc2lzOm5hbWVzOnRjOlNBTUw6Mi4wOmFzc2VydGlvbiI+
CiAgICAgICAgPE5hbWVJRCBTUE5hbWVRdWFsaWZpZXI9InhyaTovLypAZXhhbXBsZS5zcCI+eHJp
Oi8vKj1leGFtcGxlLnVzZXI8L05hbWVJRD4KICAgIDwvU3ViamVjdD4KICAgIDxOYW1lSURQb2xp
Y3kgQWxsb3dDcmVhdGU9InRydWUiLz4KICAgIDxSZXF1ZXN0ZWRBdXRobkNvbnRleHQ+CiAgICAg
ICAgPEF1dGhuQ29udGV4dENsYXNzUmVmIHhtbG5zPSJ1cm46b2FzaXM6bmFtZXM6dGM6U0FNTDoy
LjA6YXNzZXJ0aW9uIj54cmk6Ly9AeGRpLm9yZyooK2F1dGhlbnRpY2F0aW9uU2VydmljZS9pU1NP
L1NBTUwvYWMvY2xhc3Nlcy9WaXN1YWxBdXRoKTwvQXV0aG5Db250ZXh0Q2xhc3NSZWY+CiAgICA8
L1JlcXVlc3RlZEF1dGhuQ29udGV4dD4KPC9BdXRoblJlcXVlc3Q+Cgo=”/>
</div>
<noscript>
<div>
<input type="submit" value="Continue"/>
</div>
</noscript>
</form>
</body>
</html>
```

9. RESOLVE xri://*@!123be!345a!ab32

```
 GET /?@!123be!345a!ab32
Host: resolver.example.biz
```

10. XRDS response

```
<Service priority="1">
        <Type>
        xri://+i-service*(+metadata)*(+saml)*($v*2.0)
        </Type>
        <URI priority="1">https://sp.example.org/samlmetadata</URI>
        <URI priority="2">https://sp1.example.org/samlmetadata</URI>
        <ProviderID>
              xri://@exampleOrg*identityProvider/SAMLEntity
        </ProviderID>
        [...]
</Service>
```

11. GET metadata for SP

```
GET /samlmetadata HTTP/1.1
Host:sp.example.org
If-Modified-Since: Fri, 24 March 2006 19:43:32 GMT
[other headers]
```

12. metadata for SP response

```
<?xml version="1.0" encoding="UTF-8"?>
<EntityDescriptor xmlns="urn:oasis:names:tc:SAML:2.0:metadata"
     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
     xsi:schemaLocation="urn:oasis:names:tc:SAML:2.0:metadata
file:/Users/peterdavis/projects/oasis/sstc/saml-2.0-os/saml-
schema-metadata-2.0.xsd" entityID="">
    <Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
   […]
    </Signature>
    <SPSSODescriptor
        WantAssertionsSigned="true"
        validUntil="2006-05-01T00:00:00Z"
        protocolSupportEnumeration="urn:oasis:names:tc:SAML:
2.0:protocol xri://+i-service*(+authn)*(+saml)*($v*1.0)">
        <SingleLogoutService
            Binding="urn:oasis:names:tc:SAML:
2.0:bindings:HTTP-POST"
            Location="https://sp.example.org/SSO/POST/Logout"/
>
        <NameIDFormat>urn:oasis:names:tc:SAML:2.0:nameid-
format:transient</NameIDFormat>
        <AssertionConsumerService index="2"
            Binding="urn:oasis:names:tc:SAML:
2.0:bindings:HTTP-POST"
            Location="https://sp.example.org/SSO/POST/
consumer"/>
        <AssertionConsumerService index="1" isDefault="true"
            Binding="urn:oasis:names:tc:SAML:
2.0:bindings:HTTP-Artifact"
            Location="https://sp.example.org/SSO/Artifact"/>
    </SPSSODescriptor>

    <Organization>

        <OrganizationName xml:lang="en-US">Example Service
Provider</OrganizationName>
        <OrganizationDisplayName xml:lang="en-US">Example
Service Provider</OrganizationDisplayName>
        <OrganizationDisplayName xml:lang="jp">例サービス提供者</
OrganizationDisplayName>
        <OrganizationDisplayName xml:lang="it">Fornitore Di
Servizio Di Esempio</OrganizationDisplayName>
        <OrganizationURL xml:lang="en-US">http://
sp.example.org/</OrganizationURL>
    </Organization>

</EntityDescriptor>
```

13. authentication occurs
14. Redirect authNresponse

```
 HTTP/1.1 302 Object Moved
 Date: 21 Jan 2004 07:00:49 GMT
 Location:
 https://ServiceProvider.com/SAML/SLO/Browser?
 SAMLart=AAQAADWNEw5VT47wcO4z
 X%2FiEzMmFQvGknDfws2ZtqSGdkNSbsW1cmVR0bzU%
3D&RelayState=0043bfc1bc45110dae17004005b13a2b
```

15. GET artifact

```
 POST /SAML/Artifact/Resolve HTTP/1.1
 Host: IdentityProvider.com
 Content-Type: text/xml
 Content-Length: nnn
 SOAPAction: http://www.oasis-open.org/committees/security

 <SOAP-ENV:Envelope
 xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
 <SOAP-ENV:Body>
 <samlp:ArtifactResolve
 xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
 xmlns="urn:oasis:names:tc:SAML:2.0:assertion"
 ID="_6c3a4f8b9c2d" Version="2.0"
 IssueInstant="2006-01-21T19:00:49Z">
 <Issuer> xri://*@!23be!345a!ab32</Issuer>
 <Artifact>
 AAQAADWNEw5VT47wcO4zX/iEzMmFQvGknDfws2ZtqSGdkNSbsW1cmVR0bzU=
 </Artifact>
 </samlp:ArtifactResolve>
</SOAP-ENV:Body>
```

16. Artifact response

```
 HTTP/1.1 200 OK
 Date: 21 Jan 2006 07:00:49 GMT
 Content-Type: text/xml
 Content-Length: nnnn

 <SOAP-ENV:Envelope
 xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
 <SOAP-ENV:Body>
 <samlp:ArtifactResponse
 xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
 xmlns="urn:oasis:names:tc:SAML:2.0:assertion"
 ID="_FQvGknDfws2Z" Version="2.0"
 InResponseTo="_6c3a4f8b9c2d"
 IssueInstant="2006-01-21T19:00:49Z">
 <Issuer> xri://@example*identityProvider/SAMLAuthority</Issuer>
 <samlp:Status>
 <samlp:StatusCode
 Value="urn:oasis:names:tc:SAML:2.0:status:Success"/>
 </samlp:Status>


 </samlp:ArtifactResponse>
 </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```