OASIS SSTC SAML Protocols Schema Discussion

draft-sstc-protocol-discussion-01.doc

28 July 2001

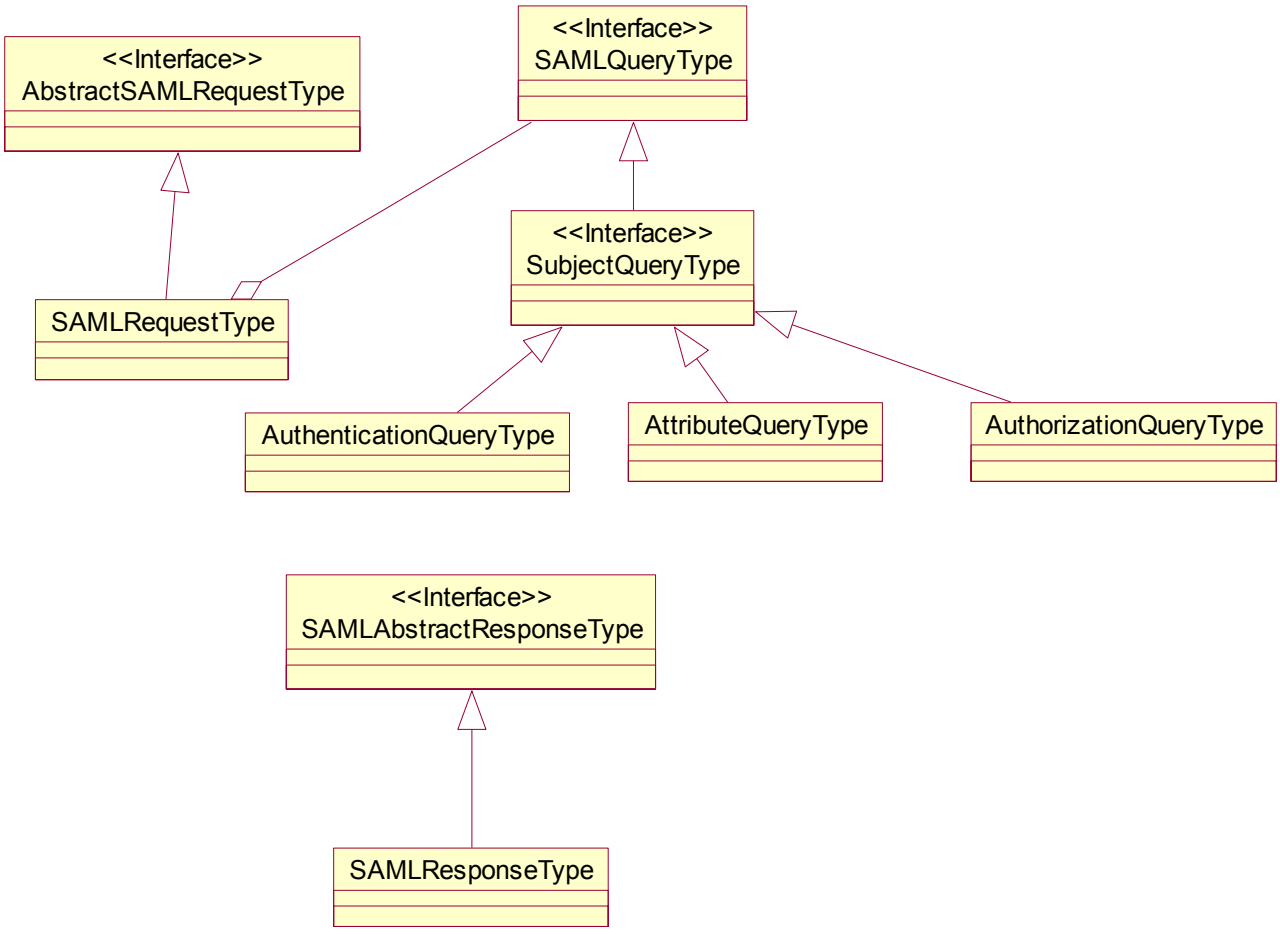
Authors:
Chris McLaren, Netegrity
Prateek Mishra, Netegrity

1 Overview

This document is the second document in a series of documents which provide a commentary on the SAML schema [draft-sstc-schema-assertion-12] and [draft-sstc-schema-protocols-12]. The first document [draft-sstc-core-discussion-01] describes the general principles governing the SAML schema design and comments on the assertion schema with examples.

2 Class Diagram

This section should contain a complete class diagram for draft-sstc-protocols-12. For now we have the following overview of a few key types types (interface represents an abstract type which cannot be directly instantiated):



3 General Architecture

3.1 Discussion

The SAML Protocols schema describes a request-response protocol for three different types of queries. Our architecture is based one request type, and one response type for all of these queries. The request type is capable of carrying many different types of queries all of which extend from one of two common abstract extension points. Three such concrete extensions are defined in the specification.

3.1.1 Inheritance Scheme

In order to ask a question in SAML one creates an instance of the SAMLRequest element. The SAMLRequest element's type definition is an extension of SAMLAbstractRequestType, which is provided as an extension point for potential future requirements.

On of possible content element in a SAMLRequest is a query. This query must be a descendant of the abstract SAMLQueryType. Three different such descendants are defined in the specification: AuthenticationQueryType, AttributeQueryType, and AuthorizationQueryType. Note that all of these are descended from the abstract base type SubjectQuery, which is in turn descended from the SAMLQueryType.

3.1.1.1 SAMLAbstractRequest and SAMLRequest

```
<xsd:complexType name="SAMLAbstractRequestType" abstract="true">
      <xsd:attribute name="RequestID" type="saml:IDType" use="required"/>
      <xsd:attribute name="Version" type="string" use="required"/>
</xsd:complexType>

<element name =”SAMLRequestType” type=”samlp:SAMLRequestType”>
<complexType name="SAMLRequestType">
      <complexContent>
            <extension base="samlp:SAMLAbstractRequestType">
                  <choice>
                        <element ref="saml:SAMLQuery"/>
                        <element ref="saml:AssertionID"
                         maxOccurs="unbounded"/>
                  </choice>
            </extension>
      </complexContent>
</complexType>
```

The SAMLAbstractRequest defines only two attributes, both of which are required. It is an abstract type that can not be instantiated and is only provided as an extension point for possible future uses.

RequestID: This is a unique identifier for the request (See also ISSUE:[CONS-02])

Version: This is the string that identifies the version of the SAML protocol being used (See also ISSUE:[CONS-05]).

The SAMLRequestType extends the abstract base by adding one of two elements to its body. The element <SAMLRequest> is made available as a container for SAMLRequestType objects.

The element found within <SAMLRequest> indicates the type of the question asked:

If an <AssertionID> is present in the body the question is interpreted as "Would you return the Assertion corresponding to this AssertionID?"

Otherwise some descendant of the abstract SAMLQueryType will be present, and will contain a more specific question. These specific questions are detailed in §4 through §6.

3.1.2 Example: SAMLRequest

```
<SAMLRequest RequestID="urn:SAMLID:17862301" Version="100">
  <AssertionID>urn:SAMLID:29100231XA</AssertionID>
</SAMLRequest>
```

In this example, the <SAMLRequest> element is requesting a SAML assertion with a certain <AssertionID>. If successful, the assertion will be returned within a <SAMLResponse> element.

3.1.3 SAMLQuery and SubjectQuery

```
<complexType name="SAMLQueryType" abstract="true"/>

<complexType name="SubjectQueryType" abstract="true">
      <complexContent>
            <extension base="samlp:SAMLQueryType">
                  <sequence>
                        <element ref="saml:Subject"/>
                  </sequence>
            </extension>
      </complexContent>
</complexType>
```

The SAMLQuery element is an abstract element that exists to provide both an extension point for possible future query types and could also act as the head element of a substitution group that covers all possible query types. The type has no defined characteristics.

The SubjectQueryType is a second abstract type, derived from the SAMLQueryType with the addition of one and only one <Subject> element. For information on the <Subject> element see §4.1.3.1 of draft-sstc-core-XX-discussion.doc

The SubjectQueryType is the base class from which the three concrete query classes defined in this specification are derived.

3.1.3.1 Concrete Query Types

Three concrete query types are defined in this specification: an authentication query, an attribute query, and an authorization query. Depending on what sort of question a SAMLRequest is contains, one of these types is used as the SAMLQuery descendant within the request. The query types defined in this specification are detailed in §4 through §6.

3.2 Issues

3.2.1 ISSUE:[PRO-01] Return of expired assertions?

Does the specification make any normative statements about the expiry state of assertions returned in response to SAMLRequests? Is it a requirement that only unexpired assertions are returned, or is the client responsible for checking? (*Seems pretty clear that the client will have to check anyway at time-of-use, so forcing the responder to check before replying seems like extra processing.*)

3.2.2 ISSUE:[PRO-02] Assertion Lookup via Other Identifiers

In some instances (such as the web browser profile) it is necessary to lookup an assertion using an identifier other than the <AssertionID>. Typically, such an identifier is opaque and may have been created in some proprietary way by an asserting party. Do we need an additional element in SAMLRequestType to model this type of lookup?

4 Authentication Query

4.1 Discussion

```
<complexType name="AuthenticationQueryType">
      <complexContent>
            <extension base="samlp:SubjectQueryType">
                  <sequence>
                        <element ref="saml:AuthenticationCode"
                         minOccurs="0"/>
                  </sequence>
            </extension>
      </complexContent>
</complexType>
```

The AuthenticationQueryType asks the question “What authentication assertions are available for this Subject?”

An AuthenticationQuery contains all the elements and attributes of a SubjectQuery and extends them with the optional addition of an AuthenticationCode. The <AuthenticationCode> element is defined in §4.1.4 of draft-sstc-core-XX-discussion.doc. In the context of a query this code can be

used to as a filter for possible responses. This supports the query "What authentication assertions do you have for this Subject with the following AuthenticationCode?"

A SAML processor will return a certain number of AuthenticationAssertions in response to a SAMLQuery of type AuthenticationQueryType. The <Subject> and <AuthenticationCode> of the returned assertions must be identical to the subject (and optional <AuthenticationCode>) fields of the SAMLQuery. There is no implication that all such assertions must be returned.

4.1.1 Example: <SAMLQuery> with type AuthenticationQueryType

```
<SAMLQuery xsi:type="samlp:AuthenticationQueryType" >
      <Subject>
            <NameIdentifier>
                  <SecurityDomain>www.example.com</SecurityDomain>
                  <Name>cn=SomeUser,co=example,ou=sales</Name>
            </NameIdentifier>
      </Subject>
      <AuthenticationCode>X509v3</AuthenticationCode>
</AuthenticationQuery>
```

In this example we are asking for all authentication assertions for a subject with "cn=SomeUser, co=example, ou=sales" with <AuthenticationCode> set to X509v3.

When wrapped in a <SAMLRequest> element we have the complete query:

```
<SAMLRequest RequestID="{EE52CAF4-3768-4ebe-84D3-4D372C892A5D}"
 Version="http://www.oasis.org/tbs/1066-12-25/protocol/">
      <SAMLQuery xsi:type="samlp:AuthenticationQueryType" >
            <Subject>
                  <NameIdentifier>
                        <SecurityDomain>www.example.com</SecurityDomain>
                        <Name>SomeUser</Name>
                  </NameIdentifier>
            </Subject>
            <AuthenticationCode>X509v3</AuthenticationCode>
      </SAMLQuery>
</SAMLRequest>
```

5 Attribute Query

5.1 Discussion

```
<complexType name="AttributeQueryType">
      <complexContent>
            <extension base="samlp:SubjectQueryType">
                  <sequence>
                        <element ref="saml:Attribute" minOccurs="0"
maxOccurs="unbounded"/>
                        <element name="CompletenessSpecifier"
                         type="samlp:CompletenessSpecifierType"
```

```
                        default="All"/>
                </sequence>
            </extension>
        </complexContent>
</complexType>
```

The AttributeQueryType can be used to ask the questions:

- "Return all of the attributes for this Subject (that I am allowed to see)?" The response will be in the form of an Attribute assertion.
- "Return ALL of the following attributes for this Subject?" The response will be in the form an Attribute assertion. If the requestor is not allowed access to ALL the requested attributes he will get NONE of them.
- "Return me ANY of the following attributes for this Subject?" The response will be in the form an Attribute assertion. Any attributes which the requestor is not allowed access to will not be present in the returned assertion.

An AttributeQuery contains all the elements and attributes of a SubjectQuery and extends them with the addition of zero or more Attributes and exactly one CompletenessSpecifier.

The set of Attributes are used to specify the list of attributes in the query in the second and third forms of the question. If there are no Attributes provided the query is interpreted as being the first question above and the CompletenessSpecifier is ignored.

The CompletenessSpecifier is a two-value enumeration of "Any" or "All". The chosen value selects between the second and third questions listed above.

5.2 Issues

5.2.1 ISSUE:[PRO-03] ANY/ALL Qualifier

Are these qualifiers necessary? Can we remove them?

See also ISSUE:[F2F#3-20, 3-21].

5.2.2 ISSUE:[PRO-04] Returning Attribute Names Only

Do we need an additional query format where only the attribute names are returned? For example, the query where all the attribute names for a subject are returned.

See also ISSUE:[F2F#3-22, 23].

5.2.3 ISSUE:[PRO-05] ALL or ``Error''

Is the all or "error" semantics for the ALL qualifier appropriate? Should we just follow LDAP semantics for this type of query?

See also [ISSUE:F2F#3-24] and discussion [F3] from [f2f#3-minutes]

5.2.4 ISSUE:[PRO-06] Expressiveness of Query Language

The approach taken in this draft is to use a very simple query language for attributes, essentially consisting of only of attribute names. At the f2f#3, there were suggestions that a richer query language might be used, such as for example based on XPATH. Is there any need to further explore this direction for SAML 1.0?

See also [F2] in [f2f#3-minutes]

5.3 Example

```
<SAMLQuery xsi:type="samlp:AttributeQueryType">
      <Subject>
            <NameIdentifier>
                  <SecurityDomain>www.example.com</SecurityDomain>
                  <Name>SomeUser</Name>
            </NameIdentifier>
      </Subject>
      <CompletenessSpecifier>ANY</CompletenessSpecifier>
      <Attribute>
            <AttributeName>NetWorthSummary</AttributeName>
            <AttributeNamespace>
                  http://ns.finance-vocab.org/finance
            </AttributeNamespace>
      </Attribute>
      <Attribute>
            <AttributeName>CreditScore</AttributeName>
            <AttributeNamespace>
                   http://ns.finance-vocab.org/finance
            </AttributeNamespace>
      </Attribute>
</SAMLQuery>
```

This example is requesting two namespace-qualified attributes for the given Subject: a "NetWorthSummary" and a "CreditScore". Since it is requested with the ANY flag that means the response will contain values for zero, one or both of these attribute names.

6 Authorization Query

6.1 Discussion

```
<complexType name="AuthorizationQueryType">
      <complexContent>
            <extension base="samlp:SubjectQueryType">
                  <sequence>
                        <element ref="saml:Evidence" minOccurs="0"
                         maxOccurs="unbounded"/>
                        <element ref="saml:Object"/>
                  </sequence>
            </extension>
```

```
      </complexContent>
</complexType>
```

An Authorization Query is used to ask only one type of question: "Should action(s) Y on resource Z be allowed for subject S given evidence E?" The answer comes in the form of an Authorization Decision assertion. The action(s) and resource are optionally namespace-scoped and are contained with in the <Object> element of the query.

An AuthorizationQuery contains all the elements and attributes of a SubjectQuery and extends them with the addition of exactly one Object element and a collection zero or more assertions or assertionIDs to be used as evidence.

The Evidence and Object elements are defined in draft-sstc-core-XX-discussion.doc; Evidence in §4.1.6 and Object in §4.1.6.1.

6.2 Issues

6.3 Example

```
<SAMLQuery xsi:type="samlp:AuthorizationQueryType">
      <Subject>
            <NameIdentifier>
                  <SecurityDomain>us-staff</SecurityDomain>
                  <Name>cn=SomeUser,ou=finance,co=example</Name>
            </NameIdentifier>
      </Subject>
      <Object>
            <Resource>
                  http://www.example.com/confidential/agree.html
            </Resource>
            <Action>GET</Action>
            <Action>POST</Action>
            <Namespace>urn:samlaction:HTTP</Namespace>
      </Object>
      <Evidence>
            <AssertionID>{EE52CAF4-3452-4ebe-84D3-4D372C892A5D}</AssertionID>
      </Evidence>
</SAMLQuery>
```

In this example, the query concerns whether subject "cn=SomeUser,…" has the right to GET AND POST on a certain URL. The actions GET, POST are taken from the namespace urn:samlaction:HTTP.

7 SAMLResponse and SAMLAbstractResponseType

7.1 Discussion

```
<complexType name="SAMLAbstractResponseType" abstract="true">
      <attribute name="ResponseID" type="saml:IDType" use="required"/>
      <attribute name="InResponseTo" type="saml:IDType" use="required"/>
      <attribute name="Version" type="string" use="required"/>
</complexType>
<element name=”SAMLResponse” type=”SAMLResponseType”/>
<complexType name="SAMLResponseType">
      <complexContent>
            <extension base="samlp:SAMLAbstractResponseType">
                  <sequence>
                        <element ref="saml:Assertion" minOccurs="0"
                         maxOccurs="unbounded"/>
                  </sequence>
                  <attribute name="StatusCode" type="samlp:StatusCodeType"
                   use="required"/>
            </extension>
      </complexContent>
</complexType>
```

All types of requests in SAML are met with a common response, the <SAMLResponse>, which has type SAMLResponseType. SAMLResponseType is an extension of the abstract base type SAMLAbstractResponseType, which is provided as an extension point for other possible future needs with respect to responses.

7.1.1 SAMLAbstractResponseType

The base type defines only the three required attributes of a response: the ResponseID, the InResponseTo identifier, and the version string.

RequestID: This is the identifier for the response. See also [ISSUE:CONS-02]

InResponseTo: This attribute holds the identifier of the request which this response is answering. Naturally whatever formatting rules, if any, that are arrived at for the request identifiers will also apply to the type of this attribute.

Version: This is the string that identifies the version of the SAML protocol being used by the responder. See also [ISSUE:CONS-05].

7.1.2 SAMLResponseType

The SAMLResponseType extends the SAMLAbstractResponseType with the addition of a required Status Code attribute and a series of zero or more assertions that represent the answers to the original request.

StatusCode: This attribute holds a string that describes the result of the original request, as expressed in this response. The contents of this attribute are restricted to members of an enumeration defined in the type "StatusCodeType" (see below).

```
<simpleType name="StatusCodeType">
      <restriction base="string">
            <enumeration value="Success"/>
            <enumeration value="Failure"/>
            <enumeration value="Error"/>
            <enumeration value="Unknown"/>
      </restriction>
</simpleType>
```

The assertions carried in the response represent the answers to the original question. The responding entity is responsible for what assertions are provided in response to any question.

7.2 Issues

7.2.1 ISSUE:[PRO-07 Status Code]

Are the status codes listed for StatusCodeType sufficient? If not how do we want to define a bigger list: keep it open with well-known values, use someone else's list, define an extension system, etc.

See also ISSUE:[F2F#3-33, 34].

7.3 Examples

The example below shows an example of a successful response to the AttributeQuery shown in Section 5.3.

```
<SAMLResponse ResponseID="{EF52CBF4-3452-4ebe-84D3-4D372C892A5D}"
 InResponseTo= "{EE52CAF4-3768-4ebe-84D3-4D372C892A5D}"
 Version="0100"
 StatusCode="Success">
      <Assertion xsi:type="saml:AttributeAssertionType"
        version="http://www.oasis.org/tbs/1066-12-25/"
        AssertionID="{EE52CAF4-3452-4ebe-84D3-4D372C892A5D}"
        Issuer="www.example.com"
        IssueInstant="2001-05-31T13:20:00-05:00">
            <Conditions NotBefore="2001-05-31T13:20:00-05:00"
              NotOnOrAfter="2001-05-31T13:25:00-05:00" />
            <Subject>
                  <NameIdentifier>
                        <SecurityDomain>www.example.com</SecurityDomain>
                        <Name>SomeUser</Name>
                  </NameIdentifier>
            </Subject>
            <Attribute>
                  <AttributeName>NetWorthSummary</AttributeName>
                  <AttributeNamespace>
```

```
                        http://ns.finance-vocab.org/finance
                  </AttributeNamespace>
                  <AttributeValue>
                        <CreditSummary>
                              <HistoryScore>Excellent</HistoryScore>
                              <CurrentAssets>Loaded</CurrentAsserts>
                        </CreditSummary>
                  </AttributeValue>
            </Attribute>
      </Assertion>
</SAMLResponse>
```

The example below shows an example of a successful response to the AuthorizationQuery shown in Section 6.3.

```
<SAMLResponse ResponseID="{EF52CBF4-3452-4ebe-84D3-4D372C892A5D}"
 InResponseTo= "{EE52CAF4-3768-4ebe-84D3-4D372C892A5D}"
 Version="0100"
 StatusCode="Success">
      <Assertion xsi:type="saml:AuthorizationDecisionAssertionType"
        version="http://www.oasis.org/tbs/1066-12-25/"
        AssertionID="{EE52CAF4-3452-4ebe-84D3-4D372C892A5D}"
        Issuer="www.example.com"
        IssueInstant="2001-05-31T13:20:00-05:00">
            <Conditions NotBefore="2001-05-31T13:20:00-05:00"
              NotOnOrAfter="2001-05-31T13:25:00-05:00" />
      <Subject>
            <NameIdentifier>
                  <SecurityDomain>us-staff</SecurityDomain>
                  <Name>cn=SomeUser,ou=finance,co=example</Name>
            </NameIdentifier>
      </Subject>
      <Object>
            <Resource>
                  http://www.example.com/confidential/agree.html
            </Resource>
            <Action>GET</Action>
            <Action>POST</Action>
            <Namespace>urn:samlaction:HTTP</Namespace>
      </Object>
      <Evidence>
            <AssertionID>{EE52CAF4-3452-4ebe-84D3-4D372C892A5D}</AssertionID>
      </Evidence>
      <Answer>Permit</Answer>
      </Assertion>
</SAMLResponse>
```

8 References

[f2f#3-minutes] http://www.oasis-open.org/committees/security/minutes/SSTC-F2F-3-Minutes-01.txt

[draft-sstc-core-discussion-01] http://www.oasis-open.org/committees/security/docs/draft-sstc-core-discussion-01.pdf

[draft-sstc-schema-assertion-12] http://www.oasis-open.org/committees/security/docs/draft-sstc-schema-assertion-12.xsd

[draft-sstc-schema-protocols-12] http://www.oasis-open.org/committees/security/docs/draft-sstc-schema-protocol-12.xsd