Introduction

This document defines the Security Assertion Markup Language (SAML). The purpose of SAML is to facilitate the exchange of authentication and authorization information.

This document is an OASIS-Draft and is (for the most part) in conformance with relevant OASIS SSTC document standards.

Send overall comments on this document to: security-services@lists.oasis-open.org, though this document, as of this update, been most actively discussed on the security-use@lists.oasis-open.org list and comments to that list about this document are just find, too.

The OASIS Security Services Technical Committee (SSTC) web pages and document repository are available here:

http://www.oasis-open.org/committees/security/

Revision History

27 February 2001: The previous version of this document (draft-sstc-saml-01.doc) was issued.

2 March 2001: draft-sstc-saml-01.doc was reviewed by the OASIS Security Services Technical Committee.

11 April 2001: Changes agreed by the OASIS SSTC meeting were incorporated, as was new material from the following sources:

draft-sstc-use-domain-03.doc

draft-sstc-core-05.doc

draft-sstc-protocols-00.doc

draft-sstc-bindings-model-02.html

draft-sstc-glossary-00.doc

The Use Cases and Requirements section of draft-sstc-saml-01.doc has been removed from this document and incorporated into a separate document entitled draft-sstc-saml-reqs-00.doc. The issues list section of draft-sstc-saml-01.doc has been removed from this document and incorporated into a separate document entitled draft-sstc-saml-reqs-issues-00.doc.

Structure of the Document

Document Sections

This document is divided into the following major sections:

Architectural Model: describes the overall structure of SAML and how its pieces relate to one another and to other components of an information security system.

Core Assertions: defines the syntax and semantics of SAML security assertions.

Request/Response Protocols: defines the syntax of messages within which SAML security assertions are exchanged.

Bindings: defines how SAML messages and assertions are used in a variety of protocols.

Security Considerations: lists the security issues implementors and users of SAML need to be aware of.

Conformance: defines what it means for an implementation to conform to the SAML specification.

Glossary: defines the technical terms used in this specification.

References: lists other documents to which this specification's text refers.

Terminology

The key words "MUST", "REQUIRED", "SHOULD", "RECOMMENDED", and "MAY" in this document are to be interpreted as described in IETF RFC 2119 [RFC 2119].

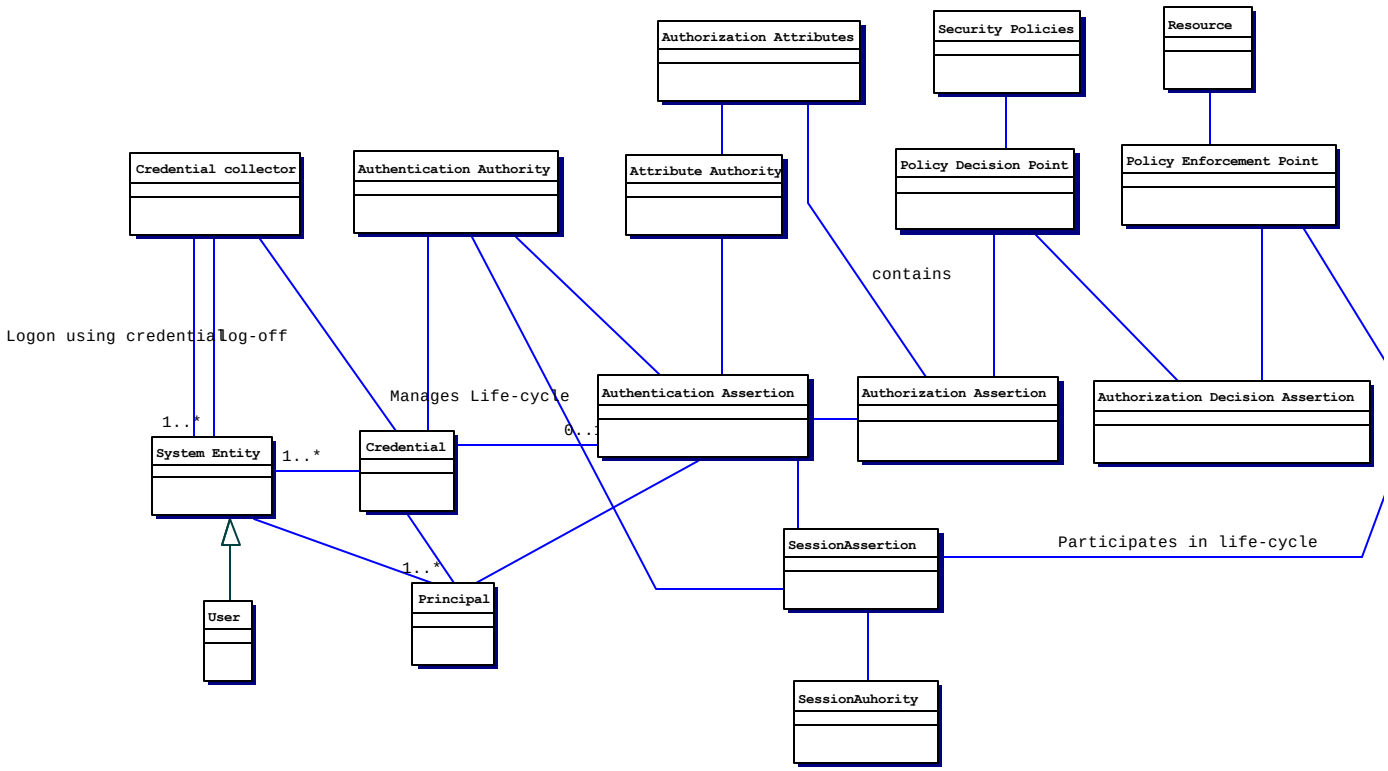
Architectural Model of the Specification

This domain model provides a description and categorization of the domain that SAML solves problems in. People, software, data, interactions, and behavior are described in the abstract, without binding the specification to a particular implementation. It provides a standardized or normalized description of concepts for the purposes of further discussion in requirements, use-cases, etc. It covers material out-of-scope for the specification in order to show the context that the specification solves problems in. It does not describe implementation information such as API details, Schema definitions and data representations.

A typical use-case for this document is: "We all agree what we mean by term x and how entity y creates it and entity z consumes it. Is x in scope or out of scope for SAML?". Another use case "We have created an OASIS TC committee on functionality A. A is the standardization of term x that is out of scope for SAML".

In the rational unified process, an artifact we are working on is the logical view, http://www.rational.com/products/whitepapers/350.jsp#RTFToC2.

Static Model



116

117 ***Glossary (abridged):***

118 **(General editor's note on this section: this has been retained in place because it captures**
119 **information about the use case subgroup's consensus. It needs to be reconciled with the**
120 **main glossary and removed here).**

121 Notation: Definitions that have been agreed upon by the use case subgroup are denoted(Conf)

122 **Assertion: TBD**

123 **Attribute Authority:** (Conf) A system entity that produces Attribute assertions, based upon
124 TBD inputs.

125 **Attribute Assertion:** An assertion about attributes of a principal.

126 **Authentication** – (from glossary with principal added) (Conf) Authentication is the process of
127 confirming an entity's asserted principal identity with a specified, or understood, level of
128 confidence. [7]

129 The process of verifying a principal identity claimed by or for a system entity. [12]

130 **Authentication Assertion:** Data vouching for the occurrence of an authentication of a principal
131 at a particular time using a particular method of authentication. Synonym(s): name assertion.

132 **Authentication Authority:** (Conf) A system entity that verifies credentials and produces
133 authentication assertions

134 **Authorization Attributes**: (Conf) Attributes about a principal which may be useful in an
135 authorization decision (group, role, title, contract code,...).

136 **Authorization Decision Assertions**: (from glossary) In concept an authorization assertion is a
137 statement of policy about a resource, such as:

138 the user "noodles" is granted "execute" privileges on the resource "/usr/bin/guitar."

139 **Authorization Assertion:** A data structure that contains Authentication Assertions and
140 Authorization attributes.

141 **Credential**: (Conf) Data that is transferred or presented to establish a claimed principal identity.

142 **Log-on:** The process of presenting credentials to an authentication authority for requesting
143 access to a resource

144 **Log-off:** The process of informing an authentication authority that previous credentials are no
145 longer valid for a User Session

Policy Decision Point: (from glossary, access control decision) The place where a decision is arrived at as a result of evaluating the requester's identity, the requested operation, and the requested resource in light of applicable security policy. (surprisingly enough, not explicitly defined in [10])

Policy Enforcement Point: (from glossary, access enforcement function) The place that is part of the access path between an initiator and a target on each access control request and enforces the decision made by the Access Decision Function [10].

Principal, or Principle Identity: (Conf) An instantiation of a system entity within the security domain.

Resource: (from glossary) Data contained in an information system (e.g. in the form of files, info in memory, etc); or a service provided by a system; or a system capability, such as processing power or communication bandwidth; or an item of system equipment (i.e., a system component--hardware, firmware, software, or documentation); or a facility that houses system operations and equipment. (definition from [1])

Security Domain: TBD

Security Policies: (from glossary) A set of rules and practices specifying the "who, what, when, why, where, and how" of access to system resources by entities (often, but not always, people).

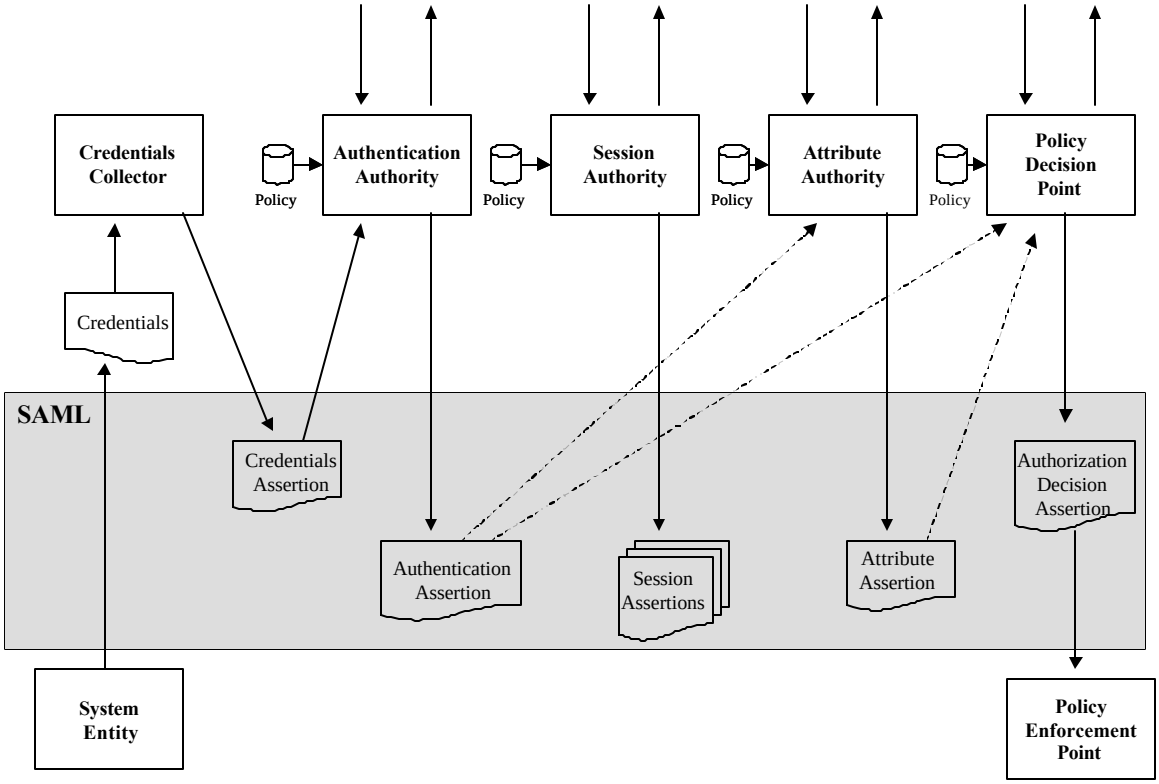
System Entity: (from glossary) (Conf) An active element of a system--e.g., an automated process, a subsystem, a person or group of persons--that incorporates a specific set of capabilities. (definition from [1])

Time Out: A step where an authorization assertion is deemed no longer viable. Subsequent resource requests from a user must proceed with log on.

User: (Conf) A human individual that makes use of resources for application purposes

User Session: A container for the authentication and attribute assertions that apply to a given system entity through the principals incarnated by that entity. The purpose is to maintain the relationship of the assertions to the initiating entity.

172 ***Producer Consumer model***



173

174 This diagram provides a view of the elements of the SAML problem space that is focused on the
175 architectural entities and their inputs and outputs. Its main purpose is to achieve a sufficient
176 commonality of understanding the meanings of the various terms used to allow productive
177 discussion. The names have been chosen either to be consistent with standard usage in the field
178 or suggestive of their purpose or action, in many cases their exact nature or contents are not fully
179 agreed upon. Although the diagram is intended to be neutral on the SAML design, the choice of
180 which elements to include and which to leave out anticipates likely elements of the design.

181 This diagram should **not** be interpreted as describe message flows or a single processing flow. It
182 merely attempts to describe which entities are capable of producing certain outputs and which
183 entities may make use of certain inputs. For example, all of the following are consistent with this
184 diagram:

185 • A PDP collects various assertions from their sources in order to make a policy decision

186 • An Attribute Assertion is returned to the System Entity that initiated the interaction
187 (lower left) who presents it as required

- A PDP makes a decision without the use of any assertions

All of the entities shown may be a part of distinct security domains, or some of them may be in the same domain. Typically there will only be two or three security domains involved. Common groupings include:

- Combined Authentication Authority and Attribute Authority
- Combined PEP and PDP
- All combined except for PEP

Many of the components can have multiple instances. For example, there can be multiple Attribute Authorities or multiple PDPs. This may introduce relationships not shown in the diagram, for example, a PDP might provide assertions to another PDP.

There is an asymmetry between input and output. The outputs that are standardized have the names shown, by definition. The entities may or may not use the inputs identified for any particular action. This is represented by the use of solid and dashed lines respectively.

The entities that have an associated policy store, are assumed to use that policy to modulate the outputs they produce. This policy store is assumed to be non-volatile and capable of being administered in some way. The unlabeled arrows at the top represent other inputs and outputs, not specified by SAML. For inputs these fall into two categories: 1) inputs which have the same semantics as SAML defined Assertions, but are in unspecified format and 2) items which are not specified by SAML at all. An example of #1 is an X.509 Attribute Certificate. An example of #2 is the current date and time.

The diagram anticipates the design of SAML by identifying only the security assertions that could be output by these entities. SAML will also have protocol messages to send and receive these assertions and will make use of existing communications protocols to transmit these assertions.

The central gray box labeled SAML indicates which assertions **may be** specified by SAML. In particular, the inclusion of Credentials Assertions and Sessions Assertions has not been settled.

The definitions of these items can be found elsewhere.

The following comments cover points that may not be completely evident.

The System Entity in the diagram is the one requesting some action that will ultimately be permitted or denied. As a preliminary step it may provide credentials to authenticate itself.

The Credentials are not merely limited to a password, but might involve a sequence of messages exchanges, for example in a Public Key authentication protocol.

The Credentials Collector is an entity that can front-end the authentication process and pass to

the Authentication Authority the information necessary for it to authenticate the System Entity. This is similar to the functionality provided by the RADIUS protocol.

The exact nature of Session Assertions has not been determined at this point. Therefore it is unknown what entities might consume them.

The Authorization Decision Assertion might simply provide a yes/no response, or it might provide specific information about why access is denied, or it might provide statements of policy.

The Policy Enforcement Point is defined to have no policy, but to act directly on the contents of the Authorization Decision Assertion.

Core Assertions

XML Assertion and Request Syntax

Each SAML protocol exchange consists of a request and response. The embedding of these requests and responses in specific protocols is described in detail in the section on Bindings.

The syntax of requests and responses are closely related and so both are described here.

Namepaces

For clarity, some examples of XML are not complete documents and namespace declarations may be omitted from XML fragments. In this document, certain namespace prefixes represent certain namespaces.

All SAML protocol elements are defined using XML schema **[XML-Schema1][XML-Schema2]**. For clarity unqualified elements in schema definitions are in the XML schema namespace:

```
xmlns="http://www.w3.org/2000/10/XMLSchema."
```

References to Security Assertion Markup Language schema defined herein use the prefix “s0” and are in the namespace:

```
xmlns:s0="http://www.oasis.org/tbs/1066-12-25/"
```

This namespace is also used for unqualified elements in message protocol examples.

The SAML schema specification uses some elements already defined in the XML Signature namespace. The “XML Signature namespace” is represented by the prefix `ds` and is declared as:

```
xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
```

The “XML Signature schema” is defined in **[XML-SIG-XSD]** and the `<ds:KeyInfo>` element (and all of its contents) are defined in **[XML-SIG]**§4.4.

SAML Assertion

SAML specifies several different types of assertion for different purposes, these are:

Authentication Assertion

Attribute Assertion

Decision Assertion

The different types of SAML assertion are encoded in a common XML package which at a minimum consists of:

Basic Information.
Each assertion MUST specify a unique identifier that serves as a name for the assertion. In addition an assertion MAY specify the date and time of issue and the time interval for which the assertion is valid.

Claims.
The claims made by the assertion. This document describes the use of assertions to make claims for Authorization and Key Delegation applications.

In addition an assertion MAY contain the following additional elements. An SAML client is not required to support processing of any element contained in an additional element **with the sole exception that an SAML client MUST reject any assertion containing a Conditions element that is not supported.**

Conditions.
The assertion status MAY be subject to conditions. The status of the assertion might be dependent on additional information from a validation service. The assertion may be dependent on other assertions being valid. The assertion may only be valid if the relying party is a member of a particular audience.

Advice.
Assertions MAY contain additional information as advice. The advice element MAY be used to specify the assertions that were used to make a policy decision.

The SAML assertion package is designed to facilitate reuse in other specifications. For this reason XML elements specific to the management of authentication and authorization data are expressed as claims. Possible additional applications of the assertion package format include management of embedded trust roots **[XTASS]** and authorization policy information **[XACML]**.

Element <Assertion>

The <Assertion> element is specified by the following schema:

```
<element name="Assertion">
   <complexType>
      <sequence>
                  <!-- Basic Information -->
         <element name="AssertionID" type="s0:AssertionID"/>
         <element name="Issuer" type="string"/>
         <element name="IssueInstant" type="DateTime”/>
         <element name="ValidityInterval" type="s0:ValidityInterval"/>

         <!-- Data -->
```

```
        <element name="Claims" type="s0:Claims"/>
        <element name="Conditions" type="s0:Conditions"/>
        <element name="Advice" type="s0:Advice"/>

    </sequence>
  </complexType>
</element>
```

SAML Request

SAML Assertions may be generated and exchanged using a variety of protocols. The bindings section of this document describes specific means of transporting SAML assertions using existing widely deployed protocols.

SAML aware clients may in addition use the request protocol defined by the `<SAMLQuery>` and `<SAMLQueryResponse>` elements described in this section.

Element <SAMLQuery>

The query specifies the principal and the resources for which access is requested by use of the claim element syntax. The information requested in the response is specified by means of the `<Respond>` element described in section 0.

The `<SAMLQuery>` element is defined by the following schema:

```
<element name="SAMLQuery">
   <complexType>
      <sequence>
                 <!-- Basic Information -->
        <element name="RequestID" type="s0:AssertionID"/>
        <element name="AssertionID" type="s0:AssertionID"/>
        <element name="ValidityInterval" type="s0:ValidityInterval"/>

        <!-- Data -->
        <element name="Query"       type="s0:Claims"/>
        <element name="Conditions"  type="s0:Conditions"/>
        <element name="Advice"      type="s0:Advice"/>

        <element name="Respond"     type="s0:Respond"/>
     </sequence>
  </complexType>
</element>
```

Element <RequestID>

The `RequestID` element defines a unique identifier for the assertion request. If an assertion query specifies a `RequestID` value the same value MUST be returned in the response unless a Respond element of `Static` is specified.

The <RequestID> element is defined by the following schema:

```
<element name="RequestID" type="string"/>
```

Element <Respond>

The <Respond> element in the request specifies one or more strings included in the request that specify data elements to be provided in the response.

The Service SHOULD return a requested data element if it is available. The Service MAY return additional data elements that were not requested. In particular, the service MAY return data elements specified in the request with the response.

Defined identifiers include:

Identifier	Description
Decision	Return the result of the Query (True/False).
Static	Specifies that the response may return any data element thus allowing the responder to return a static pre-signed assertion.
ValidityInterval	Return the ValidityInterval element
Conditions	Return the assertion conditions
Claims	Return the assertion claims
Advice	Return additional advice elements
XML Schema URI	If a URI is specified the response may contain Claims, Conditions and Advice elements specified by the corresponding XML schema.

The <Respond> element is defined by the following schema:

```
<element name="Respond" >
   <complexType>
      <sequence>
         <element name="string" type="string"
               minOccurs="0" maxOccurs="unbounded"/>
      </sequence>
   </complexType>
</element>
```

Element <SAMLQueryResponse>

The response to a <SAMLQuery> is a <SAMLQueryResponse> element. This returns the <RequestID> specified in the response together with a <Decision> element and/or an <Assertion> element. The information returned in the response is controlled by the <Respond> element of the request.

The <SAMLQueryResponse> element is defined by the following schema:

```
<element name="SAMLQueryResponse">
   <complexType>
      <sequence>
                  <!-- Basic Information -->
         <element name="RequestID"   type="s0:AssertionID"/>
         <element name="Decision"    type="s0:Decision"/>
         <element name="Assertion"   type="s0:Assertion"/>

      </sequence>
  </complexType>
</element>
```

Element <Decision>

The <Decision> element in the request specifies an authorization decision and has three possible values: Permit, Deny and Indeterminate.

The <Respond> element is defined by the following schema:

```
<simpleType name="Decision" base="string">
   <enumeration value="Permit"/>
   <enumeration value="Deny"/>
   <enumeration value="Indeterminate"/>
</simpleType>
```

Basic Information

Four basic information elements are defined; a unique identifier, the issuer, the time instant of issue, the validity interval and the assertion status.

Element <AssertionID>

Each assertion MUST specify exactly one unique assertion identifier. All identifiers are encoded as a Uniform Resource Identifier (URI) and are specified in full (use of relative identifiers is not permitted).

The URI is used as a *name* for the assertion and not as a *locator*. It is only necessary to ensure that no two assertions share the same identifier. Provision of a service to resolve an identifier into an assertion is not a requirement.

The `<AssertionID>` element is defined by the following schema:

```
<element name=“AssertionID” type="string"/>
```

Element `<Issuer>`

The `Issuer` element specifies the issuer of the assertion by means of a URI. It is defined by the following XML schema:

The `<Issuer>` element is defined by the following schema:

```
<element name="Issue" type="string"/>
```

Element `<IssueInstant>`

The time instant of issue.

The `<IssueInstant>` element is defined by the following schema:

```
<element name="IssueInstant" type="timeInstant"/>
```

Element `<ValidityInterval>`

The `<ValidityInterval>` structure specifies limits on the validity of the assertion. It contains the following elements:

Member	Type	Description
`NotBefore`	`DateTime`	Time instant at which the validity interval begins
`NotAfter`	`DateTime`	Time instant at which the validity interval has ended

The `DateTime` instant MUST fully specify the date.

The `NotBefore` and `NotAfter` elements are optional. If the value is either omitted or equal to the start of the epoch it is unspecified. If the `NotBefore` element is unspecified the assertion is valid from the start of the epoch (0000-01-01T00:00.00) until the `NotAfter` element. If the `NotAfter` element is unspecified the assertion is valid from the `NotBefore` element with no expiry. If neither element is specified the assertion is valid at any time.

In accordance with the XML Schemas Specification, all time instances are interpreted in Universal Coordinated Time unless they explicitly indicate a time zone.Implementations MUST NOT generate time instances that specify leap seconds.

For purposes of comparison, the time interval `NotBefore` to `NotAfter` begins at the earliest time instant compatible with the specification of `NotBefore` and *has ended* at the earliest time instant compatible with the specification of `NotAfter`

For example if the time interval specified is *day*`T12:03:02` to *day*`T12:05:12` the times `12:03:02.00` and `12:05:11.9999` are within the time interval. The time `12:05:12.0000` is outside the time interval.

The `<ValidityInterval>` element is defined by the following schema:

```
<complexType name="ValidityInterval">
   <sequence>
      <element name="NotBefore" type="timeInstant"/>
      <element name="NotAfter" type="timeInstant"/>
   </sequence>
</complexType>
```

Conditions

Assertion Conditions are contained in the `<Conditions>` element. SAML applications MAY define additional elements using an extension schema. If an application encounters an element contained within a `<Conditions>` element that is not understood the status of the Condition MUST be considered Indeterminate.

The following conditions are defined:

Identifier	Type	Description
`Audiences`	`URI []`	Specifies the set of audiences to which the assertion is addressed.

The `<Conditions>` element is defined by the following XML schema:

```
<element name="Conditions">
   <complexType>
      <sequence>
         <element name="Audiences" >
           <complexType >
              <sequence>
                 <element name="string" type=“string"
                          minOccurs="0" maxOccurs="unbounded"/>
               </sequence>
            </complexType>
         </element>
```

```
          <element name="ValidityDependsUpon" >
            <complexType >
               <sequence>
                   <element name="string" type="string"
                            minOccurs="0" maxOccurs="unbounded"/>
               </sequence>
             </complexType>
          </element>
      </sequence>
  </complexType>
</element>
```

Element <Audiences>

Assertions MAY be addressed to a specific audience. Although a party that is outside the audience specified is capable of drawing conclusions from an assertion, the issuer explicitly makes no representation as to accuracy or trustworthiness to such a party.

- Require users of an assertion to agree to specific terms (rule book, liability caps, relying party agreement)
- Prevent clients inadvertently relying on data that does not provide a sufficient warranty for a particular purpose
- Enable sale of per-transaction insurance services.

An audience is identified by a URI that identifies to a document that describes the terms and conditions of audience membership.

Each client is configured with a set of URIs that identify the audiences that the client is a member of, for example:

`http://cp.verisign.test/cps-2000`
Client accepts the VeriSign Certification Practices Statement

`http://rule.bizexchange.test/bizexchange_ruebook`
Client accepts the provisions of the *bizexchange* rule book.

An assertion MAY specify a set of audiences to which the assertion is addressed. If the set of audiences is the empty set there is no restriction and all audiences are addressed. Otherwise the client is not entitled to rely on the assertion unless it is addressed to one or more of the audiences that the client is a member of. For example:

`http://cp.verisign.test/cps-2000/part1`
Assertion is addressed to clients that accept the provisions of a specific part of the VeriSign CPS.

In this case the client accepts a superset of the audiences to which the assertion is addressed and may rely on the assertion.

The <Audiences> element is defined by the following XML schema:

```
<element name="Audiences" >
   <complexType >
      <sequence>
         <element name="string" type="string"
                  minOccurs="0" maxOccurs="unbounded"/>
      </sequence>
   </complexType>
</element>
```

Element <ValidityDependsOn>

The Validity of an assertion may be dependent on the validity of another assertion. For example an assertion stating that a Principal is authorized to access a resource might be dependent on another assertion specifying that the Principal has been granted a particular role.

The <ValidityDependsUpon> element specifies the <AssertionID> of one or more assertions on which the validity of the assertion depends. An assertion with a <ValidityDependsUpon> element MAY contain the assertion referenced as an <Advice> element but is not required to do so.

The <ValidityDependsUpon> element is defined by the following XML schema:

```
<element name="ValidityDependsUpon" >
   <complexType >
      <sequence>
         <element name="string" type="string"
                  minOccurs="0" maxOccurs="unbounded"/>
      </sequence>
   </complexType>
</element>
```

Claims

The <Claims> element contains one or more SAML assertion claims. At present only one type of claim is defined, the <Authority> element. Additional types of claims may be defined in future revisions of the SAML specification or by means of an extension schema.

In each case if more than one assertion claim element is specified the validity of each claim is asserted jointly and severally, that is the semantics of a single assertion containing two claims are identical to the semantics of two separate assertions each of which contain one of the claims.

The <Claims> element is defined by the following XML schema:

```
<element name="Claims">
   <complexType>
      <sequence>
         <element name="Authority" type="so:Authority"
               minOccurs="0" maxOccurs="unbounded"/>
     </sequence>
  </complexType>
</element>
```

Element <Authority>

The <Authority> element specifies a SAML authorization assertion. An <Authority> element specifies a subject, an object and an action and asserts that the principal identified by the subject is authorized to perform the specified action on the resource specified by the object.

The <Authority> element is defined by the following XML schema:

```
<element name="Authority">
   <complexType>
      <sequence>
                  <!-- Basic Information -->
         <element name=“Subject” type="s0:Subject"/>
         <element name="Object" type="s0:Object"/>
         <element name="Action" type="s0:Action"/>
     </sequence>
  </complexType>
</element>
```

Element <Subject>

The <Subject> element is defined by the following XML schema:

```
<element name="Subject">
   <complexType>
      <sequence>
         <element name=“Account” type="string"/>
         <element name=“Role” type="string"/>
         <element name=“KeyInfo” type="ds:KeyInfo"/>
     </sequence>
  </complexType>
</element>
```

Element <Object>

The <Object> element is defined by the following XML schema:

```
<element name="Object">
   <complexType>
      <sequence>
         <element name="Resource" type="string"
```

```
                minOccurs="0" maxOccurs="unbounded"/>
      </sequence>
   </complexType>
</element>
```

Element <Action>

The <Action> element is defined by the following XML schema:

```
<element name="Object">
   <complexType>
      <sequence>
         <element name="Resource" type="string"
                minOccurs="0" maxOccurs="unbounded"/>
      </sequence>
   </complexType>
</element>
```

Structured Entitlement

SAML applications MAY specify highly structured authority data in an <Authority> claim by means of an extension schema. The details of such schemas are outside the scope of SAML.

Advice

The Advice element is a general container for any additional information that does not affect the semantics or validity of the assertion itself.

Element <Advice>

The <Advice> element permits evidence supporting the assertion claims to be cited, either directly (through incorporating the claims) or indirectly (by reference to the supporting assertions.

The <Advice> element is defined by the following XML schema:

```
<element name="Advice">
   <complexType>
      <sequence>
         <element name="Assertion" type="Assertion"
                minOccurs="0" maxOccurs="unbounded"/>
     </sequence>
  </complexType>
</element>
```

[An alternative use for the Advice element that is exploited in XTASS 1.0a is for specifying reissue information. This is not employed in SAML but is the reason for the change of name since the last version in case people were wondering.]

Request/Response Protocols

The basic data objects of the SAML protocol model are "Assertions" and "References" (to Assertions). Assertions are of two different types: "authentication" and "attribute". The resulting four data objects, in their current versions, are represented in the SAML namespace. Syntax definitions for the various types of assertion can be found elsewhere.

(Note: the decision assertion is eliminated, by allowing the PEP to request an attribute assertion (or reference thereto) that affirms the question to be decided (e.g. such-and-such a Principal occupies such-and-such a role, or such-and-such a Principal is permitted to perform such-and-such an action on such-and-such an object. If the PDP returns the requested assertion (or reference thereto), without modification, it has effectively answered "Yes" to the question).

The SAML protocol specification defines a Request/Response pair of messages by which the Requestor requests that the Responder issue an assertion of a specified type. If a suitable assertion already exists, then that assertion may be returned in response to the request, without the responder having to create a new one. Even for the case where the PEP requests that the PDP return a specified list of attributes for an identified Principal, the response is treated as an assertion whose authenticity is vouched for by the PDP.

This scope does not include the request by a Principal to a PEP for access to a resource. This aspect will be addressed directly by the "Bindings" working group.

The following entities in the protocol model may adopt the role of Requestor in the exchange: Principal, PEP, PDP and Authority. The following entities in the protocol model may adopt the role of Responder in the exchange: Authority and PDP. Table 1 shows typical applications of the messages.

610

Requestor	Responder	Typical application
Principal	Authority	The Authority returns an authentication or attribute assertion (or reference thereto) with the Principal as subject
Authority	PDP	The PDP returns an authentication or attribute assertion (or reference thereto) with a Principal designated by the Authority as subject
PEP	PDP	The PDP returns an attribute assertion (or reference thereto) with a Principal designated by the PEP as subject
PDP	Authority	The Authority returns an authentication or attribute assertion with a Principal designated by the PDP as subject

611 **Table 1 - Typical applications of the request/response messages**

612 The request is in the form of a "prototype" of the required assertion. Each attribute of the
613 required assertion is represented in the prototype by a "type"/"value" pair. The requestor may
614 omit the "value" field, if it does not know, or care, what value should be assigned to the
615 corresponding element in the resulting assertion. The responder may modify the requested
616 values. It may also omit requested elements and it may add additional elements. These actions
617 are reflected in the "status" element of the response.

618 In addition to the prototype assertion, the Requestor may supply some or all of the information
619 required by the Responder to prepare the requested assertion. The additional information may
620 take the form of:

621 • Assertions of any type,

622 • References to assertions of any type, and

623 • Information about the Principal (such as its posited name and authenticator).

624 (Note: XML schemas are used here to define the contents of the request and response messages.
625 However, it is not the intention that messages conformant with these schemas will actually form
626 the messages exchanged between parties in the SAML model. The precise contents of messages
627 will depend on the transport protocols to which they are bound, and it is the task of the
628 "Bindings" working group to define the precise message contents for each transport protocol.
629 The schemas defined here serve merely as guidance to the "Bindings" working group.)

630 There are two basic message types, the Request message and the corresponding Response
631 message.

Request Message

The Request message contains the following fields.

```
<element name = "RequestIdentifier" type = "string"/>
<element name = "PrototypeAssertionsList">
        <element name = "PrototypeAssertion" minOccurs = "0" maxOccurs = "unbounded" >
                <complexType>
                        <sequence>
                                <element name = "FieldType" type = "string"/>
                                <element name = "FieldValue" type = " … " minOccurs = "0"/>
                        </sequence>
                </complexType>
        </element>
</element>
<element name = "SupportingInformation" type = "SupportingInformation"/>
</element>
```

The FieldType string is the name of the element requested to be present in the assertion returned by the responder.

The FieldValue value is the value requested for that element.

(Note: an alternative way to handle this is to include a conformant assertion whose field values are set to some special value that indicates they are to be completed.)

```
<element name = "SupportingInformation">
        <complexType>
                <sequence>
                        <element name = "Reference"
                                type = "string"
                                minOccurs = "0" maxOccurs = "1" />
                        <element name = "Assertion"
                                type = "SamlAssertion"
                                minOccurs = "0" maxOccurs = "unbounded"/>
                        <element name = "Principal"
                                type = "Principal"
                                minOccurs = "0" maxOccurs = "1"/>
                </sequence>
        </complexType>
</element>
```

```
<element name = "Principal">
    <complexType>
        <sequence>
            <element name = "Name"
                type = "Name"
                minOccurs = "0" maxOccurs = "1" />
            <element name = "Authenticator"
                type = "Authenticator"
                minOccurs = "0" maxOccurs = "unbounded"/>
        </sequence>
    </complexType>
</element>
```

The "Authenticator" element is yet to be defined. However, it must be capable of accommodating a salted password digest, a cryptographic challenge/response pair or a document/signature pair.

Response Message

The Response message contains the following fields.

```
<element name = "RequestIdentifier" type = "string"/>
<element name = "AssertionsList">
    <element name = "Assertion" minOccurs = "0" maxOccurs = "unbounded">
        <complexType>
            <sequence>
                <element name = "Assertion"
                    type = "SamlAssertion"/>
                <element name = "Status"
                    type = "Status"/>
            </sequence>
        </complexType>
    </element>
</element>
</element>
```

Bindings

The purpose of this section is to (1) characterize the scope of work and deliverables for the bindings sub-committee, (2) identify relevant work items and open issues, (3) point to relevant references. It should provide a reasonably complete starting point for the efforts of the binding sub-committee.

Definitions/terminology

[JeffH: the below list isn't definitive. Many of the terms have found their way into [Glossary]. We need to decide whether we place particular terms in this doc as well as [Glossary], or just in [Glossary]. Also we will need to refine the terminology expressed here and in [Glossary] (the latter being an overall item for SSTC, not just this subcommittee).]

assertion (aka "security assertion"?)

authn - authentication

authz - authorization

business payload - [Chris F: how is this different or distinguished from "message payload" below? JeffH: good question. I pulled this term, and "message payload" from [S2ML] and we need to figure out semantically what was being referred to in that doc, and then name them appropriately (imho).]

message payload - [Chris F: how is this different or distinguished from "business payload" above? I pulled this term, and "business payload" from [S2ML] and we need to figure out semantically what was being referred to in that doc, and then name them appropriately (imho).]

originating site

package == assertions [+ entitlements] + payload ? - [Chris F: do we want to use the term "message" here? JeffH: I agree it's possible that we do (want to use "message" rather than "package") and should discuss it.]

payload

principal

receiving site

Relying party

root -- "root of the message" (from mime?)

scruitinize

security package - one or more s2ml documents combined into a single MIME entity.

security services

subject

web service

Scope

Other Oasis Security Services TC subcommittes (e.g. Core Assertions and Protocol) are producing a specification of security assertions and services.

The high-level goal of the Bindings subcommittee is to specify how..

(1) security assertions are embedded in or combined with other objects (e.g. files of various types), communicated from site to site over various protocols, and subsequently scrutinized, and,

(2) security services defined with SAML as message exchanges
(e.g., the Authz protocol utilized between PDP and PEP in [Use Case 2, Straw2])
are mapped into one or more standard messaging protocols such as SOAP/XP and BEEP.

(1) and (2) MUST be specified in sufficient detail to yield interoperability when independently implemented.

Deliverables

- General guidelines for *binding* security assertions to payloads in the context of a protocol. The intent here is to provide general guidelines that MUST or SHOULD be followed when embedding or combining security assertions with objects drawn from an arbitrary messaging protocol.
  `[JeffH:I'm wondering just how distinct this is from the third item below. Perhaps the intent of this item is more: embedding security assertions into other objects (independent of protocols)? cf. S2ML 4.4][Chris F: I see this as being distinct from the actual bindings as it provides the overall guidelines that SHALL or SHOULD be followed when defining a protocol binding]`
  These should include considerations of the case where the assertions are "secret" versus the case when they are "scoped". cf. [S2ML]

- A process framework for describing and registering proposed and future protocol bindings.

- Bindings for selected protocols.

Bindings MUST be specified in enough detail to satisfy the interoperability requirement. The intent here is that such bindings are "recommendations" of the Oasis SSTC; the groups responsible for developing those protocols will be responsible for defining normative bindings with SAML security asssertions. This is facilitated by providing a method for describing and registering bindings.

- Standard mapping to SOAP/XP and BEEP of all security services defined within SAML. The distinction between a protocol binding and service mapping would be that the latter carries SAML assertions (and other requred data elements as determined by the service schemas) as payload whereas the bindings carry assertions at a different level (e.g., the "headers" of SOAP/XP, ebXML etc).

  We would expect each security service (e.g., Section 3.1, S2ML) to be given a high-level description by other working groups within SAML. The effort in this sub-group would focus on considerations such as required headers, selection of encoding descriptions etc. such that interoperability can be achieved between providers and consumers of SAML security services, where both parties have selected a standard messaging framework such as SOAP/XP or BEEP.

Assertion Bindings

Assertion bindings will be provided for the following standard protocols:

(a) HTTP
In case of HTTP, there is a sub-case where the user is utilizing a standard off-the-shelf browser and information about SAML assertions must be conveyed from one site to another through the browser (i.e., there is no direct site-to-site interaction). In this case, we need to ensure that mechanisms for conveying assertions from one site to another be developed that are based on URLs and HTTP headers (e.g., cookies). Both of these entities are strongly size constrained. Representing assertions by some form of "small" fixed-size object is an important consideration here [Section 6.1, S2ML].

[Section 6.2, S2ML] provides some discussion of a HTTP binding which is not constrained by the use of web browsers.

(b) MIME [Section 6.3 S2ML]

(c) SMTP [Open Issue-2: Relationship to (b) above] `[JeffH: I seriously wonder if there are any viable use cases for a SMTP binding that aren't addressed by a definition of MIME packaging for security assertions?]`

`[Chris F: note that BEEP, HTTP and ebXML also leverage or are MIME aware. One could make the same argument for all of these ;-)]`

(d) ebXML

(e) SOAP/XP

(f) BEEP

Registration/Profiling Templates

[JeffH: the below text is extracted from [BEEP] and [SASL] as boilerplate/example text that will need substantial massaging -- but whose underlying concepts are applicable here.]

Registration of a profile for using SAML

The perspective here is from the specification of some other protocol (e.g., say, ebXML, cXML, OBI, etc.) that is incorporating SAML.

From [BEEP]:

```
   5. Registration Templates

   5.1 Profile Registration Template

      When a profile is registered, the following information is
   supplied:

      Profile Identification: specify a URI[10] that authoritatively
         identifies this profile.

      Message Exchanged during Channel Creation: specify the datatypes
         that may be exchanged during channel creation.

      Messages starting one-to-one exchanges: specify the datatypes that
         may be present when an exchange starts.

      Messages in positive replies: specify the datatypes that may be
         present in a positive reply.

      Messages in negative replies: specify the datatypes that may be
         present in a negative reply.

      Messages in one-to-many exchanges: specify the datatypes that may be
         present in a one-to-many exchange.

      Message Syntax: specify the syntax of the datatypes exchanged by the
         profile.

      Message Semantics: specify the semantics of the datatypes exchanged
         by the profile.

      Contact Information: specify the postal and electronic contact
         information for the author of the profile.
```

```
   5.2 Feature Registration Template

      When a feature for the channel management profile is registered, the
      following information is supplied:

      Feature Identification: specify a string that identifies this
         feature. Unless the feature is registered with the IANA, the
         feature's identification must start with "x-".

      Feature Semantics: specify the semantics of the feature.

      Contact Information: specify the postal and electronic contact
         information for the author of the feature.
```

From [SASL]:

```
   4.    Profiling requirements

      In order to use this specification, a protocol definition must
   supply
      the following information:

      1. A service name, to be selected from the IANA registry of
   "service"
         elements for the GSSAPI host-based service name form [RFC 2078].

      2. A definition of the command to initiate the authentication
         protocol exchange.  This command must have as a parameter the
         mechanism name being selected by the client.

         The command SHOULD have an optional parameter giving an initial
         response.  This optional parameter allows the client to avoid a
         round trip when using a mechanism which is defined to have the
         client send data first.  When this initial response is sent by
   the
         client and the selected mechanism is defined to have the server
         start with an initial challenge, the command fails.  See section
         5.1 of this document for further information.

      3. A definition of the method by which the authentication protocol
         exchange is carried out, including how the challenges and
         responses are encoded, how the server indicates completion or
         failure of the exchange, how the client aborts an exchange, and
         how the exchange method interacts with any line length limits in
         the protocol.

      4. Identification of the octet where any negotiated security layer
         starts to take effect, in both directions.

      5. A specification of how the authorization identity passed from the
         client to the server is to be interpreted.
```

Registration of SAML Mechanisms

The perspective here is from the specification of some mechanism (e.g., say, some authorization mechanism) that one "plugs into" SAML. For example, the manner in which one may define and register SASL mechanisms. [JeffH: as I recall, whether or not SAML will provide for "plugin" of mechanisms (of whatever sort) into itself proper was a notion that was vigorously debated on a con-call or two. The spirit of including this subsection is therefore for present completeness' sake.]

From [SASL]:

```
   6.    Registration procedures

      Registration of a SASL mechanism is done by filling in the template
      in section 6.4 and sending it in to iana@isi.edu.  IANA has the right
      to reject obviously bogus registrations, but will perform no review
      of clams made in the registration form.

      There is no naming convention for SASL mechanisms; any name that
      conforms to the syntax of a SASL mechanism name can be registered.

      While the registration procedures do not require it, authors of SASL
      mechanisms are encouraged to seek community review and comment
      whenever that is feasible.  Authors may seek community review by
      posting a specification of their proposed mechanism as an internet-
      draft.  SASL mechanisms intended for widespread use should be
      standardized through the normal IETF process, when appropriate.

   6.1.  Comments on SASL mechanism registrations

      Comments on registered SASL mechanisms should first be sent to the
      "owner" of the mechanism.  Submitters of comments may, after a
      reasonable attempt to contact the owner, request IANA to attach their
      comment to the SASL mechanism registration itself.  If IANA approves
      of this the comment will be made accessible in conjunction with the
      SASL mechanism registration itself.

   6.2.  Location of Registered SASL Mechanism List

      SASL mechanism registrations will be posted in the anonymous FTP
      directory "ftp://ftp.isi.edu/in-notes/iana/assignments/sasl-
      mechanisms/" and all registered SASL mechanisms will be listed in the
      periodically issued "Assigned Numbers" RFC [currently STD 2, RFC
      1700].  The SASL mechanism description and other supporting material
      may also be published as an Informational RFC by sending it to "rfc-
      editor@isi.edu" (please follow the instructions to RFC authors [RFC
      2223]).


   6.3.  Change Control
```

Once a SASL mechanism registration has been published by IANA, the author may request a change to its definition. The change request follows the same procedure as the registration request.

The owner of a SASL mechanism may pass responsibility for the SASL mechanism to another person or agency by informing IANA; this can be done without discussion or review.

The IESG may reassign responsibility for a SASL mechanism. The most common case of this will be to enable changes to be made to mechanisms where the author of the registration has died, moved out of contact or is otherwise unable to make changes that are important to the community.

SASL mechanism registrations may not be deleted; mechanisms which are no longer believed appropriate for use can be declared OBSOLETE by a change to their "intended use" field; such SASL mechanisms will be clearly marked in the lists published by IANA.

The IESG is considered to be the owner of all SASL mechanisms which are on the IETF standards track.

6.4. Registration Template

To: iana@iana.org
Subject: Registration of SASL mechanism X

SASL mechanism name:

Security considerations:

Published specification (optional, recommended):

Person & email address to contact for further information:

Intended usage:

(One of COMMON, LIMITED USE or OBSOLETE)

Author/Change controller:

(Any other information that the author deems interesting may be added below this line.)

Security Assertion-based Authn & Authz Services

[Section 7, AuthXML] gives some examples of mapping a security service into SOAP messages over HTTP.

Security Considerations

(General Editor's note: this section does not yet have any content)

Conformance

(General Editor's note: this section does not yet have any content)

Glossary

This glossary comprises an overall glossary for the OASIS Security Services Technical Committee (SSTC) and its subgroups. Individual SSTC documents and/or subgroup documents may either reference this document and/or "import" select subsets of terms.

The sources for the terms and definitions herein are referenced in Appendix A. **(General editor's note: the references in the appendix are in a format which I could not get Word to interpret, and given the limited time available, I did not have time to re-type these. I'd appreciate it if we'd choose one (simple, text) reference style and separate references out into their own draft with a specified editor – Bob B.)** Please refer to those sources for definitions of terms not explicitly defined here. Where possible and convenient, hypertext links directly to definitions within the aforementioned sources are included. Some definitions are quoted directly from the sources, some are modified to fit the context of the OASIS SSTC (aka SAML) effort.

Style of use by other SAML documents

Other SAML documents may either or both (a) include copies of definitions herein (define by value), (b) refer to this document and the applicable definitions (define by reference). In the case of (a), editors of those documents should work with the glossary editor in order to normalize the value(s) of the definitions.

Notation

Definitions that need to be added (i.e. the entry is presently blank), decisions made about, or otherwise enhanced are marked with a **?**.

Definition senses and/or options - i.e. we need to decide which one(s) to base our usage on -- are denoted by "(a)", "(b)", and so on.

Definitions that've been specifically agreed to by the Use Case & Requirements (security-use@oasis-open.org) subgroup are denoted by reference to "**[Error! Bookmark not defined.]**".

Entries with a definition of "? (xxx)" means that at least the document editor suspects we need to condsider defining this term, and we haven't yet discussed it and/or no-one's taken a stab at defining it and/or we might actually not need to define it.

Editorial comments are highlighted in yellow as in this sentence. Some may also have comments attached at the end of the document.

Notes

Clarifications & Musings

It will arguably be reasonable to refer to a system implementing & using SAML as a "A", "AA", or "AAA" service - which one depending upon the functionality of the version of SAML being used, what the SSTC decides the functionality of the (potentially) various versions of SAML turn out to be, and so on. Looking ahead, may want to coin a phrase such as "a SAML-based AAA service", and think about contracting that phrase into a shorter term.

Candidates for removal

These are term that the editor thought more folks than just himself ought to think about removing.

AAA Server - synonymous with a PDP?

Access Control Factors - synonymous with access control information?

Actor - synonymous with principal?

Authc - synonymous with authn?

Clearance - specific to Multilevel Security (MLS)

Label - specific to Multilevel Security (MLS)

Policy Decision - essentially synonymous with Access Control Decision.

Receiving Site - synonymous with Relying party.

1065 ***Terms and Definitions***

AA or AAA	"Authentication and Authorization", or "Authentication, Authorization, and Accounting (or Auditing)" - each of the "A"s being a *general class* of security mechanism. These mechanisms are key building blocks for implementing security architectures and security services.
ACI	See Access Control Information.
ADF	See Access Control Decision Function.
ADI	See Access Control Decision Information.
AEF	See Access Control Enforcement Function.
AP	See Asserting Party.
AAA Administrative Component	An AAA system component whose users are typically administrators and whose function is mangement of various aspects of a AAA system deployment.
AAA Service	A network service providing AAA or AA functionality. AAA services typically implement portions of security policies, and are implemented by security mechanisms. AAA services are essentially a subset of security services, but the terms are sometimes informally used synonymously.
AAA Server	A system entity that is also an AAA system component whose function is to make policy decisions on behalf of requesters. It accepts and answers queries via some network protocol (TBD). It may or may not rely on information stored in a (external) repository, e.g. in a directory service, or a RDBMS, etc. [**Error! Bookmark not defined.**]
AAA System	A set of AAA system components delivering a AAA service.
AAA System Component	? A system entity that is one of the identifiable components of embodiments of AAA systems.

AAA System Deployment	An instance of a deployed AAA system. An AAA System Deployment is typically hosted within, and delivers security services to, a given administrative domain. It also may be utilized to provide such services to other administrative domains.
Access	The ability and means to communicate with, or otherwise interact with, a system entity in order to manipulate, and/or use, and/or gain knowledge of, some (or all) of a system entity's system resources. [Error! Bookmark not defined.]
Access Control	1. Protection of system resources against unauthorized access; a process by which use of system resources is regulated according to a security policy and is permitted by only authorized system entities (users, programs, processes, or other systems) according to that policy. [Error! Bookmark not defined.] 2. The prevention of unauthorized access of a resource, including the prevention of use of a resource in an unauthorized manner. [Error! Bookmark not defined.]
Access Control Decision	? The decision arrived at as a result of evaluating the requester's identity, the requested operation, and the requested resource in light of applicable security policy. (surprisingly enough, not explicitly defined in [Error! Bookmark not defined.])
Access Control Decision Function	A specialized function that makes access control decisions by applying access control policy rules to an access request, access control decision information (of initiators, targets, access requests, or that retained from prior decisions), and the context in which the access request is made [Error! Bookmark not defined.].

Access Control Decision Information	The portion (possibly all) of the Access Control Information made available to the Access Decision Function in making a particular access control decision [Error! Bookmark not defined.].
Access Control Enforcement Function	A specialized function that is part of the access path between an initiator and a target on each access request and enforces the decision made by the Access Control Decision Function [Error! Bookmark not defined.].
Access Control Information	Any information used for access control purposes, including contextual information [Error! Bookmark not defined.].
Access Control Factors	A request, when being processed by a server, may be associated with a wide variety of security-related *factors* (e.g. section 4.2 of [Error! Bookmark not defined.]). The server uses these factors to determine whether and how to process the request. These are called *access control factors* (ACFs). They might include source IP address, encryption strength, the type of operation being requested, time of day, etc. Some factors may be specific to the request itself, others may be associated with the connection via which the request is transmitted, others (e.g. time of day) may be "environmental". [Error! Bookmark not defined.]
Access Control Policy	The set of rules that define the conditions under which an access may take place [Error! Bookmark not defined.].
Access Control Policy Rules	? Security policy rules concerning the provision of the access control service [Error! Bookmark not defined.].
Access Path	? (haven't been able to find a concise def for this with a modicum of looking)
Access Permissions	? (xxx)
Access Privileges	? (xxx)

Access Rights	? (xxx)
Access Request	The operations and operands that form part of an attempted access of a system resource. An access request may be communicated between parties via a request. [Error! Bookmark not defined.]
Active Role	? A role that an actor has donned when performing some operation, e.g. accessing a resource.
Actor	? From [Error! Bookmark not defined.]: A computational entity [i.e. system entity] utilizing security services. Examples of actors include application servers, application programs, security services (?), transport and message-level interceptors etc. Perhaps actor is effectively synonymous with system entity.
Administrative Domain	An environment or context that is defined by some combination of administrative policies, Internet Domain Name registration(s), civil legal entity(ies) (e.g. individual(s), corporation(s), or other formally organized entity(ies)), plus a collection of hosts, network devices and the interconnecting networks (and possibly other traits), plus (often various) network services and applications running upon them. An Administrative Domain may contain or define one or more security domains. An administrative domain may encompass a single site or multiple sites. The traits defining an Administrative Domain may, and in many cases will, evolve over time. Administrative Domains may interact and enter into agreements for providing and/or consuming services across Administrative Domain boundaries.

Administrator	A person who installs, maintains, and/or makes use of the resources of a AAA System Deployment for system management and/or user management and/or content management purposes (as opposed to application purposes. See also End User). An administrator is typically affiliated with a particular administrative domain and *may* be affiliated with more than one administrative domain. See also deployer.
Anonymity	The quality or state of being anonymous.
Anonymous	The condition of having a name [or identity] that is unknown or concealed. **[Error! Bookmark not defined.]**
Application Server	A software system run on a host that provides an execution environment for higher-level applications, for example business-oriented apps.
Assertion	(a) A piece of data constituting a declaration of identity or authorizations. See also: credential. ? (b) "Data that is transferred to establish the claimed identity of an entity." **[Error! Bookmark not defined.]**
Asserting Party	? An issuer of assertions.
Attack	An assault on system security that derives from an intelligent threat, i.e., an intelligent act that is a deliberate attempt (especially in the sense of a method or technique) to evade security services and violate the security policy of a system. **[Error! Bookmark not defined.]**

Attribute	A distinct characteristic of an object. An object's attributes are said to describe the object. Objects' attributes are often specified in terms of their physical traits, such as size, shape, weight, and color, address, phone number, etc., for real-world objects. Objects in cyberspace might have attributes describing size, type of encoding, network address, etc. Which attributes of an object are salient is decided by the beholder. Attributes are of various types, and are often represented by an attribute name along with one or more attribute values. See also Attribute Value Assertion, entry. [Error! Bookmark not defined.] [Error! Bookmark not defined.]
Attribute Authority	? (a) A system entity that produces Attribute assertions, based upon TBD inputs. [Error! Bookmark not defined.] (b) An authority which assigns privileges by issuing attribute certificates. [Error! Bookmark not defined.]
Attribute Assertion	? An assertion about attributes of a principal.
Attribute Name	The human-palatable name associated with a particular attribute type.
Attribute List	A data structure consisting of lists of attribute value assertions (aka name-value pairs). [Error! Bookmark not defined.]
Attribute Type	An attribute type typically governs whether an attribute is single- or multi-valued, the syntax to which the values must conform, the kinds of matching which can be performed on values of that attribute, and other functions. [Error! Bookmark not defined.]
Attribute Value	An attribute value is one or more pieces of data, encoded according to the syntax of the attribute's type. [Error! Bookmark not defined.]

Attribute Value Assertion	An Attribute Value Assertion is an assertion with the general abstract form of "attribute type IS attribute value". [Error! Bookmark not defined.]
Audit	Independent review and examination of records and activities to determine compliance with established usage policies and to detect possible inadequacies in product technical security policies of their enforcement. [Error! Bookmark not defined.]
Audit Identity	An identity attribute containing an identity used only for accountability purposes. [Error! Bookmark not defined.]
Authc	See Authentication
Authn	See Authentication
Authz	See Authorization
Authenticate	? (a) To verify (i.e., establish the truth of) an identity claimed by or for a system entity. [Error! Bookmark not defined.] [Error! Bookmark not defined.] (b) "to authenticate" - the act of presenting one's credentials in order to become authenticated.
Authentication	? (a) Authentication is the process of confirming a system entity's asserted principal identity with a specified, or understood, level of confidence. [Error! Bookmark not defined.] [Error! Bookmark not defined.] (b) The process of verifying a principal identity claimed by or for a system entity. [Error! Bookmark not defined.] [Error! Bookmark not defined.]
Authentication Assertion	Data vouching for the occurrence of an authentication of a principal at a particular time using a particular authentication mechanism. Synonym(s): name assertion.

Authentication Authority	A system entity that verifies credentials and produces authentication assertions. [Error! Bookmark not defined.]
Authentication Mechanism	? *Examples*.. • Simple username & password. • Kerberos • Client-side (and server-side) authn via the TLS/SSL "handshake protocol" during TLS/SSL session establishment. • Any SASL mechanism. JeffH hasn't yet found a concise and referenceable def for this term.
Authority	An identified computer-based entity implementing a security service (e.g. creation of assertions, credentials, PACs, and so on). [Error! Bookmark not defined.]
Authorization	? The process of determining which types of activities are permitted. Usually, authorization is in the context of authentication. Once you have authenticated an entity, the entity may be authorized different types of access or activity. [Error! Bookmark not defined.] <rough>The "act of authorization" is when an AEF acts upon information received from an ADF.</rough> The (act of) granting of access rights to a subject (for example, a user, or program). [Error! Bookmark not defined.]

Authorization Assertion	? In concept, an authorization assertion is a statement of policy about a resource, such as: The user "noodles" is granted "execute" privileges on the resource "/usr/bin/guitar." Should this be Authorization Decision?
Authorization Attribute	Attributes about a principal which may be useful in an authorization decision (group, role, title, contract code,...). [Error! Bookmark not defined.]
Authorization Data	A data structure that contains Authentication Assertions and Authorization attributes.
Authorization Identity	? An authorization identity is one kind of access control factor. It is the name of the user or other entity that requests that operations be performed. Access control policies are often expressed in terms of authorization identities; e.g., entity X can perform operation Y on resource Z. [Error! Bookmark not defined.] The transmitted authorization identity may be different than the identity in the client's authentication credentials. This permits agents such as proxy servers to authenticate using their own credentials, yet request the access privileges of the identity for which they are proxying. [Error! Bookmark not defined.]
Authorized	A system entity or actor is "authorized" if it is granted a right or a permission or a capability to access a system resource. See also authorization.
Capability	A token that gives its holder the right to access a system resource. Possession of the token is accepted by the access control mechanism as proof that the holder has been authorized to access the resource named or indicated by the token. [Error! Bookmark not defined.]

Clearance	Initiator-bound ACI that can be compared with security labels of targets ⟦Error! Bookmark not defined.⟧.
Client	A system entity that requests and uses a service provided by another system entity, called a "server". ⟦Error! Bookmark not defined.⟧
Context	? See Contextual Information. (we may actually want to use a much more general, commonplace definition of context - i.e. what we mean when we're waving our hands and saying something like "that all depends upon the context". This because contextual information is defined narrowly.
Contextual Information	Information about or derived from the context in which an access request is made (e.g. time of day). ⟦Error! Bookmark not defined.⟧. Effectively synonymous with access control factors.
Control Attribute	? Attributes, associated with a security object that, when matched against the privilege attributes of a security subject, are used to grant or deny access to the security object. ⟦Error! Bookmark not defined.⟧
Credential	? (a) Data that is transferred or presented to establish either a claimed identity or the authorizations of a system entity. (See also: assertion, authentication information, capability, ticket.) ⟦Error! Bookmark not defined.⟧ (b) Data that is transferred to establish a claimed principal identity. ⟦Error! Bookmark not defined.⟧ ⟦Error! Bookmark not defined.⟧ --- We need to decide between (a) and (b).

Decision	The response of an Access Control Decision Function to a decision request [Error! Bookmark not defined.], using terminology from [Error! Bookmark not defined.]. See also access control decision.
Decision Request	The message an Access Control Enforcement Function sends to an Access Decision Function to ask it whether a particular access request should be granted or denied [Error! Bookmark not defined.] , using terminology from [Error! Bookmark not defined.].
Deployer	An administrator in the act of, and/or (sometimes) primarily responsible for deploying a particular system or systems in an administrative domain's network infrastructure. This may involve configuring the system or systems to interact with systems of other administrative domains.
Deployment Time	The time at which a system is actually configured, tested, and/or put to use, as opposed to its being in the vendor's development pipeline or in transit between the vendor and a customer. See also site-specific.
DMZ	"DMZ" is from the military term for an area between two opponents where fighting is prevented. See also [Error! Bookmark not defined.] and DMZ network.
DMZ network	DMZ network is a commonly-used, equivalent term for (see also) perimeter network.
End User	An entity, usually a human individual, that makes use of resources for application purposes (as opposed to system management purposes. See Administrator).
End User's Computer	A host that an end user makes use of for general computational, application, and communication purposes.

End User Profile	Various attributes and attribute values, mapped to a given end user. User attributes are stored in the profile, e.g. identifier(s), name(s), contact information, organizational information, computing infrastructure information, etc. Profiles are often implemented as directory entries.
End User System	Typically the combination of: an End User, plus the End User's computer, plus the browser running on that computer. End User system is (often? sometimes?) used, in place of the terms "client" or "user" because there are often many components that act as clients of other components, and which may not be directly and/or actively controlled by a user.
Entitlement	? (a) A data structure containing Access Control Decision Information and/or access control policy rule information in a form usable by applications to, for example, customize their behavior based on access control policy or to make access control decisions in their own code [Error! Bookmark not defined.] , using terminology from [Error! Bookmark not defined.]. (b) a digitally signed XML assertion consisting of a "portable" package of authorization data created by an issuing authority concerning an authenticated subject. [Error! Bookmark not defined.]
Entity	See System Entity.
EU System	See End User System.
EUS	See End User System.
External Network(s)	Networks outside one's administrative domain and (in typical usage of the term) with which one's networks are connected.

Extranet	The part of a company or organization's computer network which is available to outside users, for example, information services for customers and/or suppliers. [Error! Bookmark not defined.] See also extranet in [Error! Bookmark not defined.].
Firewall	A firewall is a device that gives an administrative domain a means to control how their internal network(s) interact with external networks.
Firewall boundary	A commonly-used term referring to a security perimeter that is largely defined by the presence of one or more firewalls.
Host	A computer that is attached to a communication subnetwork or internetwork and can use services provided by the network to exchange data with other attached systems. A host is distinguished from other similarly connected and addressable devices on the network, e.g. routers, in that it doesn't forward Internet Protocol packets that are not addressed to it. A host may be either an end user's computer or a server. [Error! Bookmark not defined.]
Identity	A representation (e.g. a string) uniquely mapped to a system entity (e.g. an end user, an administrator, a host, or some process, or some network device).
Initiator	An entity (e.g. human user or computer-based entity) that *attempts to access* other entities [Error! Bookmark not defined.].
Intermediary	? An entity which, after receiving an access request from an initiator, issues another access request on that initiator's behalf [Error! Bookmark not defined.]. This is a narrow definition of intermediary and is essentially the same a "proxy". We need to carefully think about our use of this term and carefully define it and associated terms.
Internal Network	See Intranet.

Intranet	A local area network which may or may not be connected to the Internet, but which has some similar functions. Some organizations set up World Wide Web servers on their own internal networks so employees have access to the organization's web documents. [Error! Bookmark not defined.] See also intranet in [Error! Bookmark not defined.].
Issuer	? A system entity that issues stuff, e.g. an issuer of assertions. [Error! Bookmark not defined.]
Label	A marking that is bound to a protected resource and that names or designates the security-relevant attributes of that resource (derived from [Error! Bookmark not defined.]).
Network-based security	The notion of controlling network access and usage, and consequently protecting hosts from attack, via network routing configuration and filtering, the use of firewalls and similar devices, or some combination thereof. See also [Error! Bookmark not defined.].
Network Device or Network Element	For the purposes of this document, one of routers, bridges, repeaters, hubs, switches, etc.
Network Service	Work performed (or offered) by a server over a network. This may mean simply serving simple requests for data to be sent or stored (as with web servers); or it may be more complex work, such as that of print servers, distributed file servers, X Windows servers, AAA servers, or application servers. (definition largely from [Error! Bookmark not defined.])
Network Topology	A configuration of network devices and hosts, and their interconnections.
Operation	The action that an initiator's access request asks to have performed on a protected resource [Error! Bookmark not defined.].

Origin Server	The server on which a given resource resides or is to be created. [Error! Bookmark not defined.]
Origin Site, Originating Site	? The site where the origin server resides.
PAC	See Privilege Attribute Certificate.
PDP	See Policy Decision Point.
PEP	See Policy Enforcement Point.
Package	= assertions [+ entitlements] + payload ?
Party	? An actor or actors (principal or principals) participating in some process or communication, such as accessing a resource. See also: access request, system entity, user.
Passive Role	? A role that a resource effectively dons when it is the *object* of some operation.
Payload	The essential data that is being carried within a packet or other transmission unit. The payload does not include the "overhead" data required to get the packet to its destination. Note that what constitutes the payload may depend on the point-of-view. To a communications layer that needs some of the overhead data to do its job, the payload is sometimes considered to include the part of the overhead data that this layer handles. However, in more general usage, the payload is the bits that get delivered to the end user (or whatever entity) at the destination. [Error! Bookmark not defined.]
Perimeter Network	A network between external networks and internal networks whose explicit role is to facilitate creation and management of additional layer(s) of security (as compared to not having a perimeter network). Also sometimes called a DMZ network. See also [Error! Bookmark not defined.].

Perimeter Security	Network-based security applied at the perimeter of one's security domain. See also [Error! Bookmark not defined.].
Policy, Policies	? Concisely, a policy is a mapping of user credentials with authority to act [Error! Bookmark not defined.]. Policies are often essentially access control *lists*. [Error! Bookmark not defined.]
Policy Decision	? essentially synonymous with Access Control Decision.
Policy Decision Point	? (a) A [system] entity that makes policy decisions for itself or for other system entities that request such decisions. [Error! Bookmark not defined.] (b) Synonymous with Access Control Decision Function. [Error! Bookmark not defined.] (c) Synonymous with AAA Server. --- JeffH feels that (a) and (b) are essentially equivalent and we need to decide whether.. 1. we use (a) "as is", or, 2. we use (b) "as is" (this would mean moving the def for Access Control Decision Function to this location), or, 3. we use (c) "as is", or, 4. we blend the three definitions together Selecting any of the above options involves deleting the entries for Access Control Decision Function and AAA Server from this doc, and updating all definitions using those terms to use the new terms.

Policy Enforcement Point	? (a) A [system] entity that [requests and subsequently] enforces policy decisions. [Error! Bookmark not defined.] (b) Synonymous with Access Control Enforcement Function. [Error! Bookmark not defined.] --- JeffH feels that (a) and (b) are essentially equivalent and we need to decide whether.. 1. we use (a) "as is", or, 2. we use (b) "as is" (this would mean moving the def for Access Control Enforcement Function to this location), or, 3. we blend the two definitions together. Selecting any of the above options involves deleting the entry for Access Control Enforcement Function itself from this doc, and updating all definitions using those terms to use the new terms.
Principal Principal Identity	? (a) AAA Service clients are sometimes called *principals* in order to distinguish them from clients of other services, and perhaps their own clients, if they are themselves servers. Note that a AAA service principal may be any form of system entity. [Error! Bookmark not defined.] (b) An instantiation of a system entity within the security domain. [Error! Bookmark not defined.] (c) An entity whose identity can be authenticated. [Error! Bookmark not defined.]
Privilege Attribute	An attribute associated with an initiator that, when matched against control attributes of a protected resource is used to grant or deny access to that protected resource (derived from ECMA TR/46 definition). [Error! Bookmark not defined.]

Privilege Attribute Certificate	A data structure containing privilege attributes. May be signed by the authority which generated it [Error! Bookmark not defined.].
Protected Resource	A target, access to which is restricted by an access control policy [Error! Bookmark not defined.].
Protected Web Resources	Web resources whose availability to requesters is being managed, i.e. protected, via some access control mechanism.
Proxy	(a) An entity authorized to act for another; (b) authority or power to act for another ; (c) a document giving such authority; [Error! Bookmark not defined.]
Proxy Server	A computer process that relays a protocol between client and server computer systems, by appearing to the client to be the server and appearing to the server to be the client. [Error! Bookmark not defined.]
Pull	? (xxx)
Push	? (xxx)
RP	See Relying Party.
Receiving Site	? A site that receives, interprets, and acts according to security assertions. Essentially synonymous to relying party.
Relying Party	? One who is making a decision contingent upon information or advice from another entity. E.g. an entity that is *relying* upon various security assertions about some other party(ies), made by yet another party(ies).
Resource	? Synonymous in this document for System Resource. JeffH feel's that we need to decide whether we use the term "resource" or "system resource" in this and other SAML docs. We need to choose one and use it consistently.

Request	? What clients make to servers. (need to enhance this ;)
Requester	As in "service requester", or "requester of resources". A system entity that is utilizing a protocol to request services from a service. Essentially functionally equivalent to the term client, but often used rather than "client" because many system entities simultaneously and/or serially act as both clients and servers.
Risk	(a) In the computer system and networking sense: An *expectation of loss* expressed as the probability that a particular threat (or set of threats) will exploit a particular vulnerability (or set of vulnerabilities) with a particular harmful result(s). **[Error! Bookmark not defined.]** (b) In general, the level of risk in a given context is inversely proportional to the level of trust the relationships within the context are accorded. **[Error! Bookmark not defined.]** (c) More generally: possibility of loss or injury. **[Error! Bookmark not defined.]**
Risk Analysis	Risk analysis involves determining what you need to protect, what you need to protect it from, and how to protect it. It is the process of examining all of your risks, then ranking those risks by level of severity. For example, see the Risk Assessment section of Chapter 2 in **[Error! Bookmark not defined.]**.
Role	? Dictionaries define a *role* as "a character or part played by a performer" or "a function or position." Principals *don* various types of roles serially and/or simultaneously, e.g. active roles and passive roles. The notion of an Administrator is often an example of a role.
Scrutinize	To examine or observe with great care; inspect critically. **[Error! Bookmark not defined.]**

Security	Security refers to a collection of safeguards that ensure the confidentiality of information, protect the system(s) or network(s) used to process it, and control access to it (them). Security typically encompasses the concepts/topics/themes of *secrecy*, *confidentiality*, *integrity*, and *availability*.It is intended to ensure that a system resists potentially correlated attacks. [Error! Bookmark not defined.]
Security Architecture	A plan and set of principles for an administrative domain and its security domains that describe (a) the security services that a system is required to provide to meet the needs of its users, (b) the system elements required to implement the services, and (c) the performance levels required in the elements to deal with the threat environment. A complete system security architecture addresses administrative security, communication security, computer security, emanations security, personnel security, and physical security. It prescribes security policies for each. A complete security architecture needs to deal with both intentional, intelligent threats and accidental kinds of threats. A security architecture should explicitly evolve over time as an integral part of its administrative domain's evolution. [Error! Bookmark not defined.]
Security Assertion	? An assertion that is typically scrutinized in the context of a security policy.
Security Domain	An environment or context that is defined by security policies, *security models*, and a security architecture, including a set of system resources and set of system entities that are authorized to access the resources. An administrative domain may contain one or more security domains. The traits defining a given security domain typically evolve over time. [Error! Bookmark not defined.]

Security Mechanism	The logic or algorithm that implements a particular security-enforcing or security-relevant function in hardware and software. [Error! Bookmark not defined.]
Security Object	A system entity in a passive role to which a security policy applies. [Error! Bookmark not defined.]
Security Package	? one or more security assertions or credentials combined into a single overall, for example, MIME-encoded data structure, or package.
Security Perimeter	The boundary of a security domain. [Error! Bookmark not defined.]
Security Policy	A set of rules and practices specifying the "who, what, when, why, where, and how" of access to system resources by system entities (often, but not always, involving or acting on behalf of *people*). Significant portions of security policies are implemented via security services. Security policies are components of security architectures. [Error! Bookmark not defined.]
Security Requirements	The types and levels of protection necessary for equipment, data, information, applications, and facilities to meet security policy [given the results of a risk analysis]. [Error! Bookmark not defined.]
Security Service	A processing or communication service that is provided by a system to give a specific kind of protection to system resources, where said resources may reside with said system or reside with other systems. E.g. an authentication service, a PKI-based document attribution & authentication service. Security Service describes a superset of AAA services. Security services typically implement portions of security policies, and are implemented via security mechanisms. [Error! Bookmark not defined.]

Security Subject	An entity in an active role to which a security policy applies. [Error! Bookmark not defined.]
Server	A process or set of processes running on a host that provide a network service. See also Server Host. [Error! Bookmark not defined.]
Server Host	A host on which a network service is being run. For example, the host upon which a web server is being run is one kind of a server host, referred to in this glossary as a web server host. Hosts regarded as server hosts are *typically not used simultaneously* as end users' computers, *but may be*.
Service	See Network Service.
Site	A term commonly used to refer to an administrative domain in geographical and/or DNS name sense. Thus *site* may refer to a particular geographical and/or topological subportion of an administrative domain, or, a site may contain multiple administrative domains, as may be the case at an ASP site.
Site-specific	A thing or a thing's deployment configuration that is tailored on a site-by-site basis. For example, *how* a site configures and performs load balancing of incoming HTTP requests to web server hosts *is site-specific*. From a vendor's perspective, site-specific decisions are usually made at deployment time.
SSL/TCP/IP	A shorthand notation denoting a protocol stack consisting of the SSL session layer running over the TCP/IP layers. An application layer protocol, e.g. LDAP or HTTP, is typically run on top of the SSL layer (which in turn is running on top of TCP/IP), and uses that layer (SSL) for end-to-end connection security.

Subject	? An identifiable entity. See also security subject. We will likely be describing a subject in terms of a principal, e.g. a subject of a PK certificate identifies the principal the certificate binds the PK to.
System	(a) A specific IT installation, with a particular purpose and operational environment. (b) An assembly of computer and/or communications hardware, software, and firmware configured for the purpose of classifying, sorting, calculating, computing, summarizing, transmitting, receiving, storing, and retrieving data, with the purpose of supporting users. (c) IT products assembled together - either directly or with additional computer hardware, software, and/or firmware - configured to perform a particular function within a particular operational environment. [Error! Bookmark not defined.] by way of [Error! Bookmark not defined.]
System Entity	An active element of a system--e.g., an automated process or set of processes, a subsystem, a person or group of persons--that incorporates a specific set of capabilities. [Error! Bookmark not defined.] [Error! Bookmark not defined.] JeffH wonders if we shouldn't use a phrase other than "specific set of capabilities here because the latter might be confused with capabilities in the access control mechanism sense rather than generic capabilities something like a system entity might have or embody.

System Resource	? (a) Data contained in an information system (e.g. in the form of files, information in memory, etc); or a service provided by a system; or a system capability, such as processing power or communication bandwidth; or an item of system equipment (i.e., a system component--hardware, firmware, software, or documentation); or a facility that houses system operations and equipment. [Error! Bookmark not defined.] (b) Anything used or consumed while performing a function. [Error! Bookmark not defined.] (c) Data contained in a system entity (e.g. in the form of files, information in memory, etc); or a service provided by a system entity; --- JeffH feels that (a) and (b) are essentially equivalent and we need to decide whether.. 1. we use (a) "as is", or, 2. we use (b) "as is", or, 3. we create another definition, perhaps based upon (a) &| (b), e.g. (c), and use that.
Target	? (a) An entity to which access may be attempted [Error! Bookmark not defined.]. (b) A resource an entity attempts to access. JeffH suspects sense (b) is the one we should use.

Threat	A potential for violation of security, which exists when there is a circumstance, capability, action, or event that could breach security and cause harm. That is, a threat is a possible danger that might exploit a vulnerability. A threat can be either "intentional" (i.e., intelligent; e.g., an individual cracker or a criminal organization) or "accidental" (e.g., the possibility of a computer malfunctioning, or the possibility of an "act of God" such as an earthquake, a fire, or a tornado). [Error! Bookmark not defined.] See especially [Error! Bookmark not defined.].
TCP or TCP/IP	See Transmission Control Protocol.
Ticket	? Aka a token. Specific example: Kerberos Tickets. See [RFC1510]. A ticket *may* be considered a credential.
Token	? See ticket.
Unauthorized	The opposite of a system entity or requester being authorized.
URL	See Uniform Resource Locator.
User	(a) A corporeal human making use of network services and/or application(s) inhabiting a given administrative domain(s), *as a means* rather than as an end. (based on "user" from [Error! Bookmark not defined.]). See also Administrator, End User. (b) A human individual that makes use of resources for application purposes [Error! Bookmark not defined.] --- JeffH feels that (a) and (b) are essentially equivalent and we need to decide whether.. 1. we use (a) "as is", or, 2. we use (b) "as is", or, 3. we blend the two definitions together.

User Profile or User's Profile	See End User Profile.
User Session	A "container" for the authentication and attribute assertions that apply to a given system entity through the principals incarnated by that entity. The purpose is to maintain the relationship of the assertions to the initiating entity. [Error! Bookmark not defined.]
Uniform Resource Locator	Defined as "a compact string representation for a resource available via the Internet." See [Error! Bookmark not defined.].
Vulnerability	A flaw or weakness in a system's design, implementation, or operation and management that could be exploited to violate the system's security policy. [Error! Bookmark not defined.]
Web-based Service	A network service where requesters are typically web browsers being wielded by end-users, and where the content delivered to the end-users' browsers via the web servers is the network service's primary end-user interface.
Web Browser	A software application used to locate and display web pages.
Web Resource	Any object (e.g. a file (e.g. a web page), a program, or any other system resource) that is being made available to requesters via a web server. Also known as "web-accessible resource". The implication here is that one may make reference to, and access, a web resource via a URL.
Web Server	A server process running on a server host and answering HTTP requests (at least),and often also several other protocols (e.g. FTP, Gopher). See also HTTP Server in [Error! Bookmark not defined.]. A web server is typically used to implement a web-based service.

Web Server Host	A host running a web server that is in turn providing some or all of the web resources accessible via the web server.
Web Service	See Web-based service.
Web Site	A web site is a site and/or administrative domain providing at least HTTP- (and often FTP-based) network services (sometimes called web services) to some set of users, with perhaps additional services offered based on yet other protocols such as LDAP . The distinguishing characteristic of a web site is that its users may make use of URLs to make reference to, and also to access, the web site's services and web resources.

1066

References

[AuthXML] AuthXML: A Specification for Authentication Information in XML. http://www.oasis-open.org/committees/security/docs/draft-authxml-v2.pdf

[BEEP] The Blocks Extensible Exchange Protocol Core http://www.normos.org/ietf/draft/draft-ietf-beep-framework-11.txt

[Glossary] OASIS Security Services TC: Glossary. http://www.oasis-open.org/committees/security/docs/draft-sstc-hodges-glossary-02.html

[Kerberos] *TBS*

[PKCS1] Kaliski, B., *PKCS #1: RSA Encryption Version 2.*0, RSA Laboratories, also IETF RFC 2437, October 1998.

[RFC-2104] Krawczyk, H., Bellare, M. and R. Canetti, *HMAC: Keyed Hashing for Message Authentication*, IETF RFC 2104, February 1997.

[RFC 2119] "Key words for use in RFCs to Indicate Requirement Levels", RFC 2119

[S2ML] S2ML: Security Services Markup Language, Version 0.8a, January 8, 2001. http://www.oasis-open.org/committees/security/docs/draft-s2ml-v08a.pdf

[SAML-USE] *TBS*

[SASL] Simple Authentication and Security Layer (SASL) http://www.ietf.org/rfc/rfc2222.txt

[Shib] Shiboleth Overview and Requirements http://middleware.internet2.edu/shibboleth/docs/draft-internet2-shibboleth-requirements-00.htmlhttp://middleware.internet2.edu/shibboleth/docs/draft-internet2-shibboleth-requirements-00.html

[SOAP] D. Box, D Ehnebuske, G. Kakivaya, A. Layman, N. Mendelsohn, H. Frystyk Nielsen, S Thatte, D. Winer. *Simple Object Access Protocol (SOAP) 1.1*, W3C Note 08 May 2000, http://www.w3.org/TR/SOAP

[Straw2] Oasis Security Services Use Cases And Requirements, Straw Man Draft 2, 9 Feb 2001 http://unique.outlook.net/~evan/a2mluc/usecases-strawman-2.html http://middleware.internet2.edu/shibboleth/docs/draft-internet2-shibboleth-requirements-00.html

[WSSL] E. Christensen, F. Curbera, G. Meredith, S. Weerawarana, *Web Services Description Language (WSDL) 1.0* September 25, 2000, http://msdn.microsoft.com/xml/general/wsdl.asp

[XACML] *TBS*

[XML-SIG] D. Eastlake, J. R., D. Solo, M. Bartel, J. Boyer , B. Fox , E. Simon. *XML-Signature Syntax and Processing*, World Wide Web Consortium. http://www.w3.org/TR/xmldsig-core/

[XML-SIG-XSD] XML Signature Schema available from http://www.w3.org/TR/2000/CR-xmldsig-core-20001031/xmldsig-core-schema.xsd.

[XML-Enc] *XML Encryption Specification*, In development.

[XML-Schema1] H. S. Thompson, D. Beech, M. Maloney, N. Mendelsohn. *XML Schema Part 1: Structures*, W3C Working Draft 22 September 2000, http://www.w3.org/TR/2000/WD-xmlschema-1-20000922/, latest draft at http://www.w3.org/TR/xmlschema-1/

[XML-Schema2] P. V. Biron, A. Malhotra, *XML Schema Part 2: Datatypes*; W3C Working Draft 22 September 2000, http://www.w3.org/TR/2000/WD-xmlschema-2-20000922/, latest draft at http://www.w3.org/TR/xmlschema-2/

[XTASS] P. Hallam-Baker, *XML Trust Axiom Service Specification 1.0*, VeriSign Inc. January 2001. http://www.xmltrustcenter.org/