# Security Assertion Markup Language

# Revision History

| | |
|---|---|
| Version | Draft 1 |
| Date | 27 February 2001 |
| Editor | Bob Blakley |
| Comments | First draft<br>Includes input from the following subgroups:<br>Use Cases and Requirements<br>Request/Response Protocols<br>Bindings<br>Glossary<br>This version is the input to FTF #1 |

# Introduction

# Structure of the Document

## Architectural Model of the Specification

## Document Sections

# Conformance

## Normative and Non-Normative Material

## Levels of Conformance

# Use Cases and Requirements

## Purpose

This document describes the requirements and use cases for the Security Assertions Markup Language (SAML) derived by the Oasis Security Services Technical Committee.

## Introduction

This document provides an initial set of use cases and requirements for the Oasis Security Services Technical Committee's (TC's) ultimate product, SAML, an XML standard for exchanging authentication and authorization data between security systems.

### Notes on This Document

Requirements are specified as a list of goals and non-goals for the project. Use cases in this document are illustrated with UML (Unified Modelling Language) diagrams. A link to the UML home page is provided below. UML diagrams are analysis and design tools, and each diagram format can support multiple levels of abstraction. In this document a balance has been struck between using a standard diagram format for requirements elaboration, and maintaining a high level of abstraction.

The document uses UML-style use-case diagrams to illustrate high-level use cases. The following list is probably sufficient as a crash course in UML use-case diagrams:

- Stick figures represents actors or roles in a scenario. These can be human beings or

software systems.

- Ellipses represent use cases, i.e. actions or units of functionality in a system.
- Lines between actors and use cases indicate a participation of the actor in the use case. Note that no direction or payload of data flow is expressed by the lines between actors and use cases.

Use-case diagrams capture high-level functionality of a system or interaction without providing excessive implementation detail. The document uses UML sequence diagrams to illustrate detailed use case scenarios. For quick reference, a sequence diagram works as follows:

- Boxes at the top of the diagram represent an actor in the scenario.
- Arrows with a solid head represent a message sent from one actor to another. The arrow points from sender to receiver.
- Arrows with a line head represent the return value of a message. The arrow points from the receiver of the earlier message to the sender.
- A dotted line ("swim lane") running down the diagram from a box indicates that arrows whose endpoints (tail or head) is on the line apply to that actor.
- Intersections between arrows and dotted lines are meaningless.
- Vertical layout represents time. Messages (arrows) farther down on the page happen after messages higher on the page.
- Horizontal layout has no formal meaning. Since right-pointing arrows look better, actors that initiate a scenario tend to appear leftward of actors they send messages to.

Note that sequence diagrams are often used for more concrete design, and that actors and messages are often objects and object methods. They provide value for this document in that they give a clearly ordered message layout. The actors and messages in the sequence diagrams below are more properly roles in a scenario and actions associated with that scenario. Readers will probably be interested in the accompanying glossary and issues list.

# Requirements

The requirements describe the scope of the SAML standard.

## Goals

- [R-AuthN] SAML should define a data format for authentication assertions, including descriptions of authentication events. This includes time of authentication event and authentication protocol.

- [R-AuthZ] SAML should define a data format for authorization attributes. Authorization attributes ("authz attributes") are attributes of a principal that are used to make authorization decisions, e.g. an identifier, group or role membership, or other user profile information.

- [R-AuthZDecision] SAML should define a data format for recording authorization decisions.

- [R-UserSession] SAML shall support web user sessions.

- [R-Anonymity] SAML will allow assertions to be made about anonymous principals, where "anonymous" means that an assertion about a principal does not include an attribute uniquely identifying the principal (ex: user name, distinguished name, etc.).

- [R-Pseudonymity] SAML will allow assertions to be made about principals using pseudonyms for identifiers.

- [R-Message] SAML should define a message format and protocol for distributing SAML data.

- [R-PushMessage] SAML's messaging protocol should support "pushing" data assertions from an authoritative source to a receiver.

- [R-PullMessage] SAML's messaging protocol should support "pulling" data assertions from an authoritative source to a receiver.

- [R-Reference] SAML should define a data format for providing references to authentication and authorization assertions.

- [R-MultiDomain] SAML should enable communication between zones of security administration.

- [R-SingleDomain] SAML should enable communication within a single zone of security administration.

- [R-Signature] SAML assertions and messages should be authenticatable.

- [R-Open] SAML should not be dependent on any particular security or user database format.

- [R-XML] SAML should be defined in XML.

- [R-Extensible] SAML should be easily extensible.

- [R-Bindings] SAML should allow SAML messages to be transported by standard Internet protocols. SAML should define bindings of the message protocol to at least the following

protocols:

- standard commercial browsers
- HTTP as a transport protocol
- MIME as a packaging protocol
- XML Protocol as a messaging protocol
- ebXML as a messaging protocol

### Non-Goals

- SAML will not propose any new cryptographic technologies or models for security; instead, the emphasis is on description and use of well-known security technologies utilizing a standard syntax (markup language) in the context of the Internet.
- Non-repudiation services and markup are outside the scope of SAML.
- Challenge-response authentication protocols are outside the scope of SAML.
- SAML does not provide for negotiation between authorities about trust between domains and realms or the inclusion of optional data. Trust negotiations must be made out-of-band.
- No provision is made for protecting SAML messages from interception by third parties. This is left up to the transport mechanism of choice between authorities.
- No specification is made for providing authorization policies through SAML.

# Use Cases And Scenarios

This section provides a set of high-level use cases for SAML and use case scenarios that illustrate the use case. They give a very abstract view of the intended use of the SAML format. Each use case has a short description, a use case diagram in UML format, and a list of the steps involved in the case. Note that, for each use case, the mechanics of how the actions are performed is not described. More detail provided in the detailed use case scenarios. Each of these high-level use cases has one or more specializations in the detailed use-case scenarios.

Each scenario contains a short description of the scenario, a UML sequence diagram illustrating the action in the scenario, a description of each step, and a list of requirements that are related to the scenario.

## Use Case 1: Single Sign-On

In this use case, a Web user authenticates with a Web site. The Web user then uses a secured resource at another Web site, without directly authenticating to that Web site.

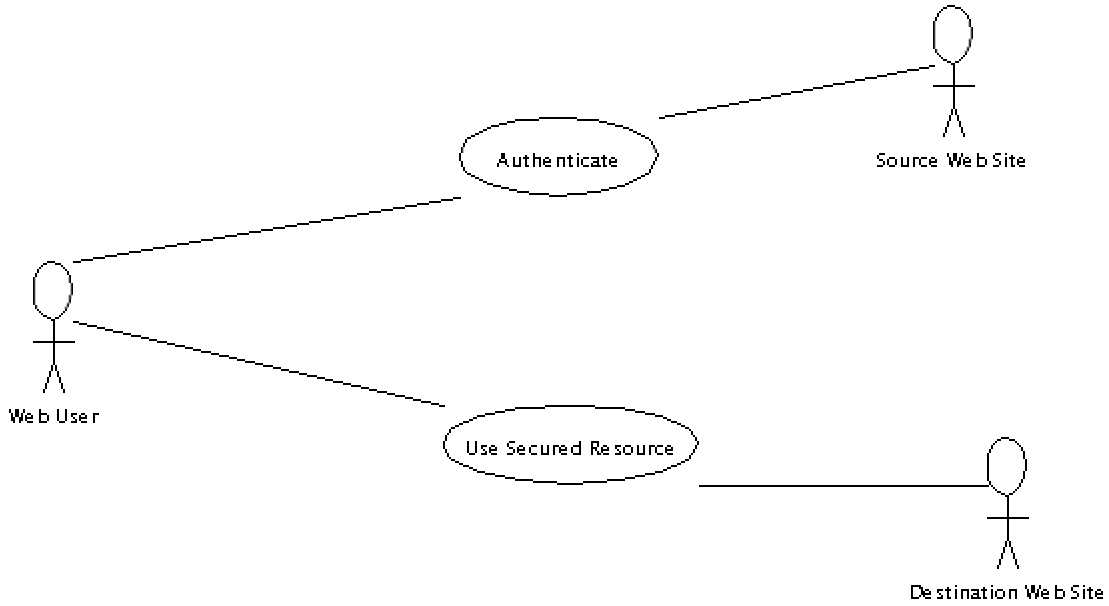



Fig 1. Single Sign-on.

Steps:

1. Web user authenticates to the source Web site.
2. Web user uses a secured resource at the destination Web site.

### Scenario 1-1: Single Sign-on, Pull Model

This scenario is an elaboration of the Single Sign-on use case. In this model, the destination Web site pulls authentication information from the source Web site based on references or tokens provided by the Web user.

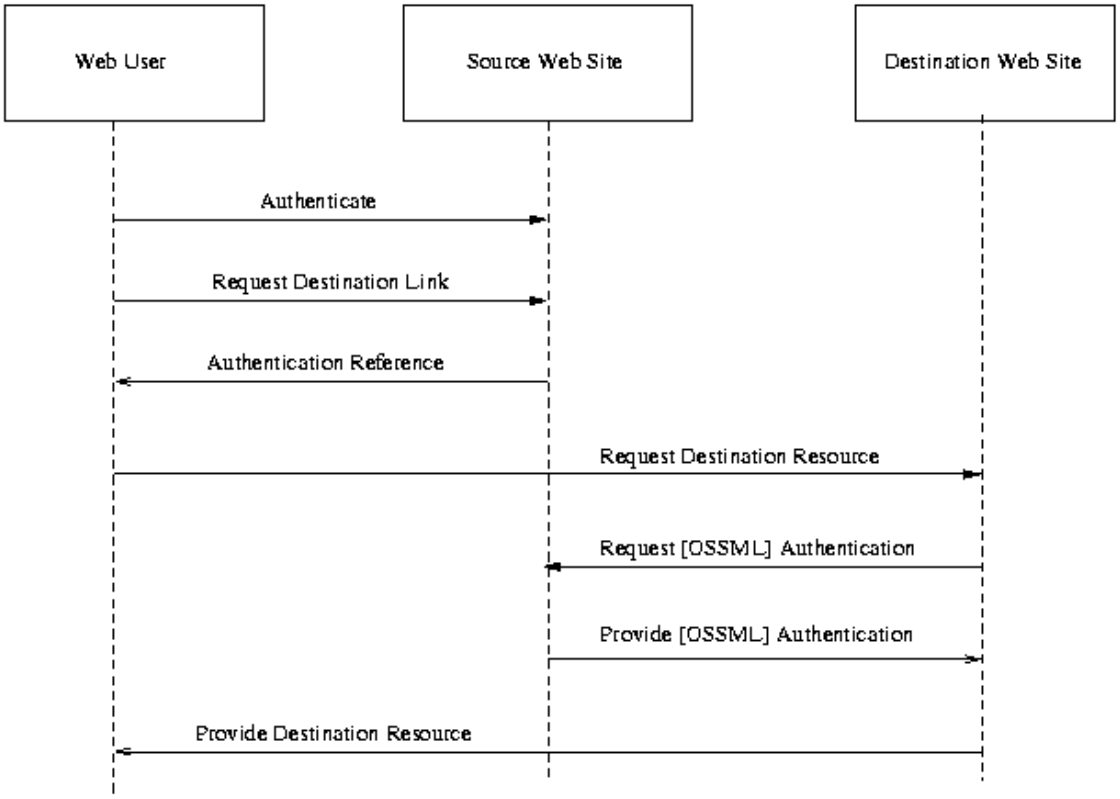



Fig 2. Single Sign-on, Pull Model.

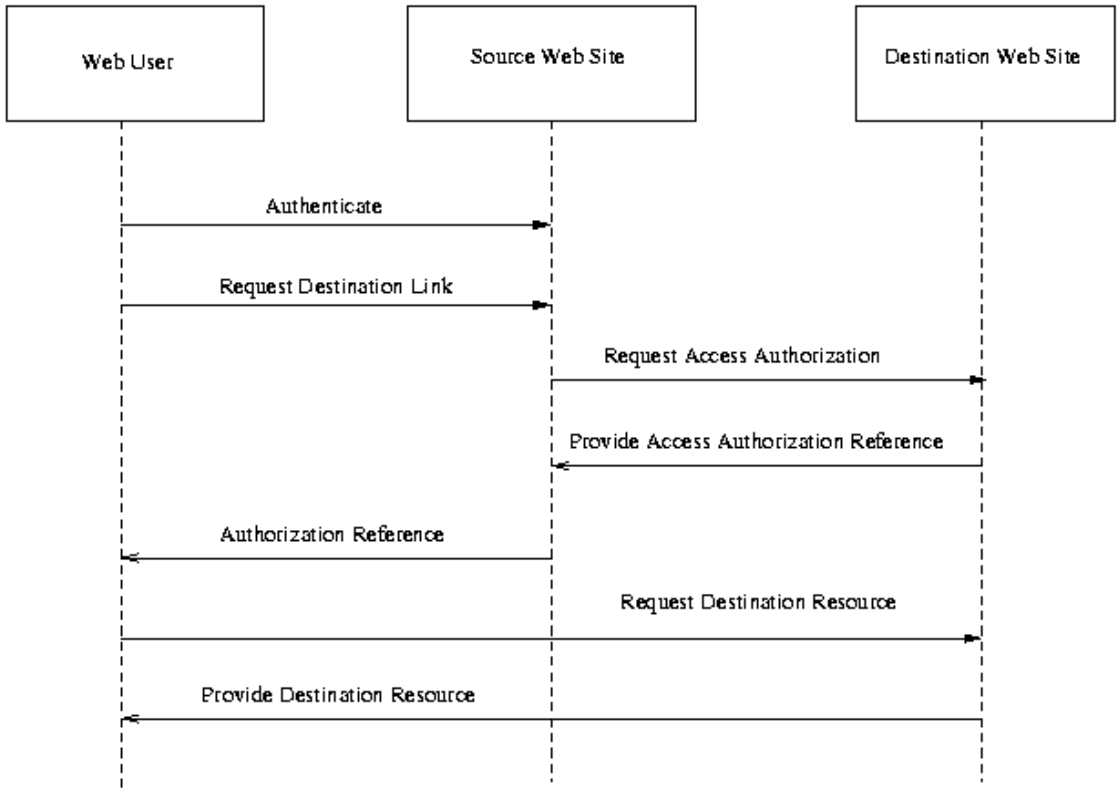
Steps:

1. Web user authenticates with source Web site.
2. Web user requests link to destination Web site.
3. Source Web site provides user with authentication reference (AKA "name assertion reference"), and redirects user to destination Web site.
4. Web user requests destination Web site resource, providing authentication reference.
5. Destination Web site requests authentication document ("name assertion") from source Web site, passing authentication reference.
6. Source Web site returns authentication document. This document includes authn event description and authz attributions about the Web user.
7. Destination Web site provides resource to Web user.

Associated requirements: [R-AuthN], [R-PullMessage], [R-MultiDomain], [R-Bindings] (standard commercial browsers), [R-Reference].

147 **Scenario 1-2: Single Sign-on, Push Model**

148 This scenario is a variation on the Single Sign-on use case. It's called the "push model" because
149 the source Web site pushes authentication information to the destination Web site.




150
151 Fig 3. Single Sign-on, Push Model.

152 Steps:

153 1. Web user authenticates with source Web site.

154 2. Web user requests link to destination Web site.

155 3. Source Web site sends requests for Web user to use destination resource from destination
156 Web site, pushing the authentication information (authentication assertion) for the user to
157 the destination site. This assertion includes authorization attributes.

158 4. Destination Web site returns an authz decision reference to Source Web site, recording
159 the decision to allow the user to access the resource.

160 5. Source Web site provides user with authz decision reference and redirects user to
161 destination Web site.

162 6. User requests destination resource from destination Web site, providing authz decision

reference.

7. Destination Web site provides resource to Web user.

Associated requirements: [R-AuthN], [R-AuthZ], [R-AuthZDecision], [R-PullMessage], [R-MultiDomain], [R-Bindings] (standard commercial browsers), [R-Reference].

**Scenario 1-3: Single Sign-on, Third-Party Security Service**

In this single sign-on scenario, a third-party security service provides authentication assertions for the user. Multiple destination sites can use the same authentication assertions to authenticate the Web user. Note that the first interaction, between the security service and the first destination site, uses the pull model as described above. The second interaction uses the push model. Either of the interactions could use a different single sign-on model.

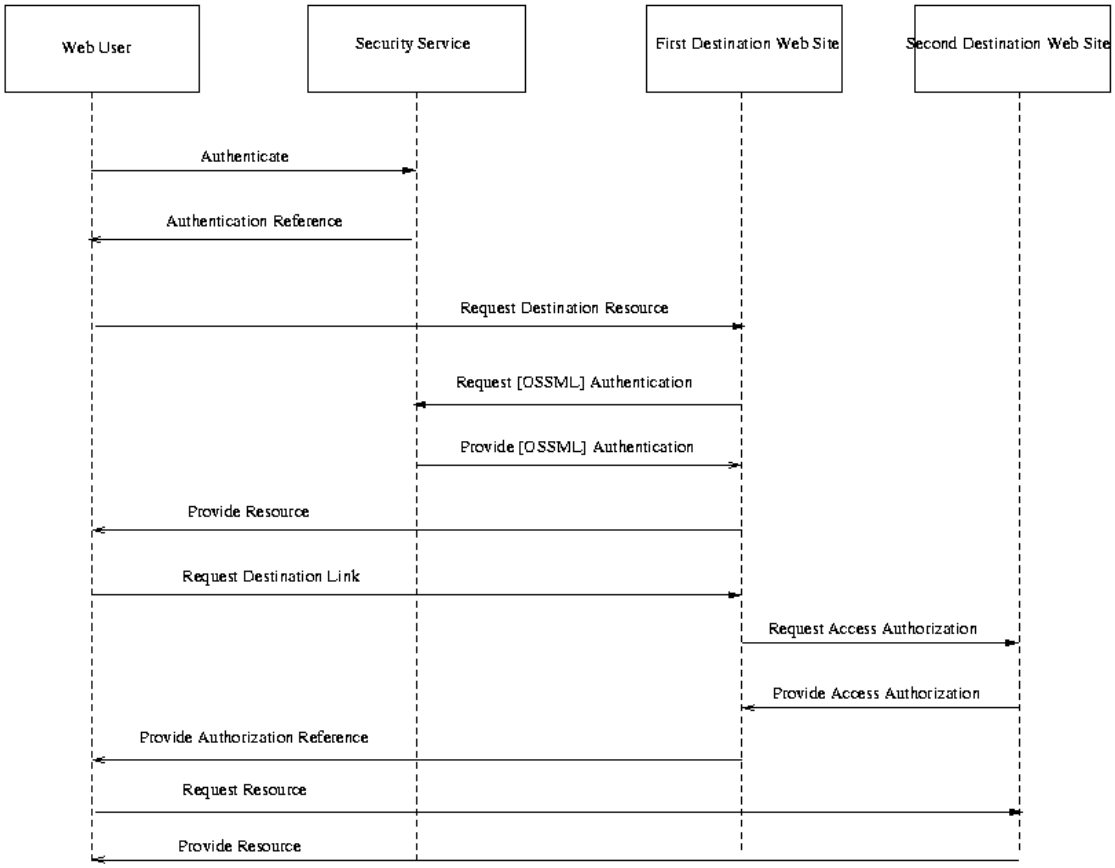



Fig. 4. Single Sign-on, Third-Party Security Service

Steps:

1. Web user authenticates with security service.

2. Security service returns SAML authentication reference to Web user.

3. Web user requests resource from first destination Web site, providing authentication reference.
4. First destination Web site requests authentication document from security service, passing the Web user's authentication reference.
5. Security service provides authentication document to first destination Web site, including authorization attributes and authn event description.
6. First destination Web site provides resource to Web user.
7. Web user requests link to second destination Web site from first destination Web site.
8. First destination Web site requests access authorization from second destination Web site, providing third-party security service authentication document for user.
9. Second destination Web site provides access authorization, returning an authz decision reference.
10. First destination Web site provides authz decision reference to Web user.
11. Web user requests resource from second destination Web site, providing authz decision reference.
12. Second destination Web site provides resource.

Associated requirements: [R-AuthN], [R-AuthZDecision], [R-AuthZ], [R-PullMessage], [R-MultiDomain], [R-Bindings] (standard commercial browsers), [R-Reference].

**Scenario 1-3: Single Sign-on, User Session**

In this single sign-on scenario, a Web user is logs into a Web site and thus instigates a user session. This session is maintained as the user navigates to other Web sites.

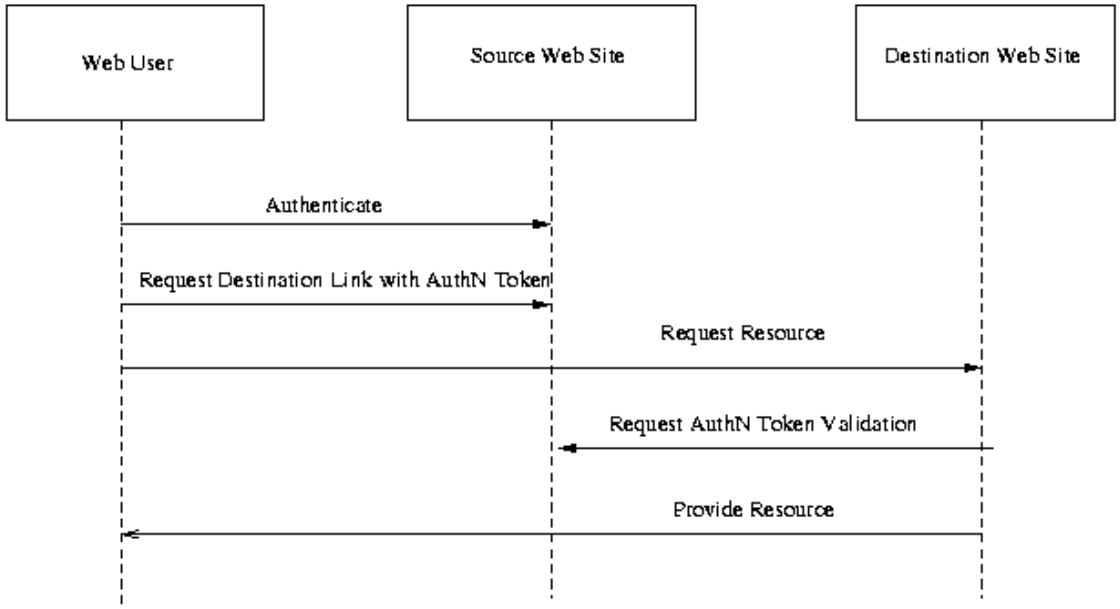
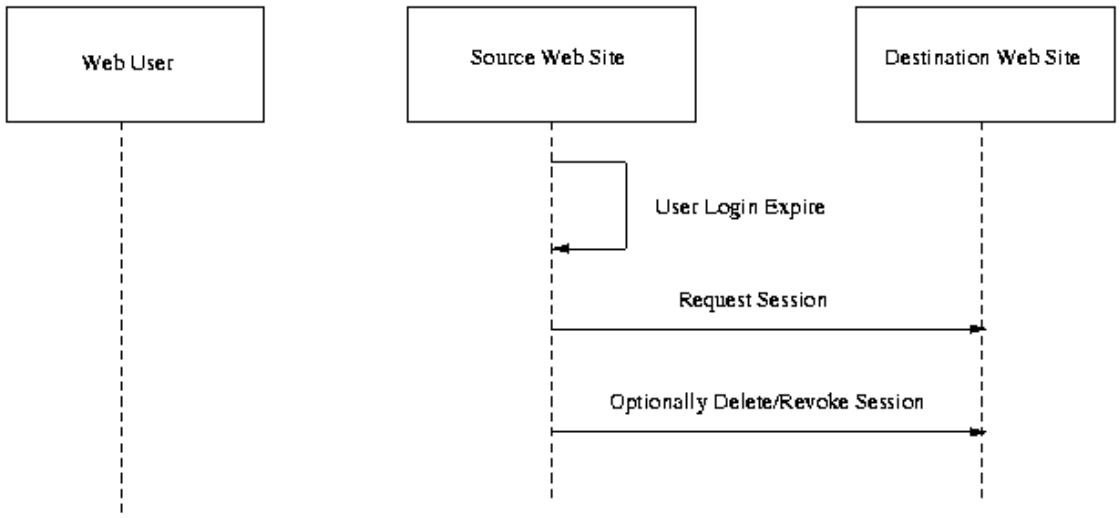



Fig. 5. Single Sign-on, User Session

Steps:

1. A user logs onto the source Web site. This results in the creation of a session on the source Web site.
2. User requests a link to a destination Web site. This link contains an authentication reference/token/ticket.
3. User requests resource represented by link on destination Web site, including reference.
4. Destination Web site requests validation of authentication reference from source Web site.
5. Source Web site returns success or failure, optionally additional session information.
6. Destination Web site returns Web site to user.




211

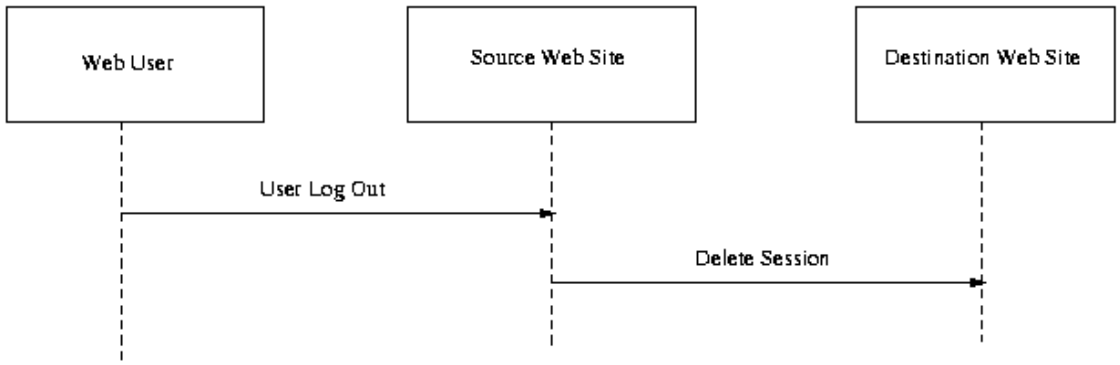
212

213 Fig. 6. User Session Timeout

214 Assume that the user has gone beyond the timeout limit on the source Web site.

215 1. The source Web site will query each participating Web site to determine if the user has
216 been active on their Web site.

217 2. If the user has not been active on any of the destination Web sites within the timeout
218 period, the destination Web sites are instructed to delete the session.




219

220

221 Fig. 6. User Session Logout

222 Logout

223 1. User logs out of the source Web site.

2. Each of the destination Web sites are instructed to delete the session.

Associated requirements: [R-AuthN], [R-AuthZ], [R-PullMessage], [R-PushMessage], [R-MultiDomain], [R-Bindings] (standard commercial browsers), [R-Reference], [R-UserSession].

## Use Case 2: Authorization Service

In this use case, a user attempts to access a resource or service. The security controller for that resource -- a policy enforcement point or PEP -- checks the user's authorization to access the resource with a policy decision point or PDP. The PDP provides an authorization service to the PEP.

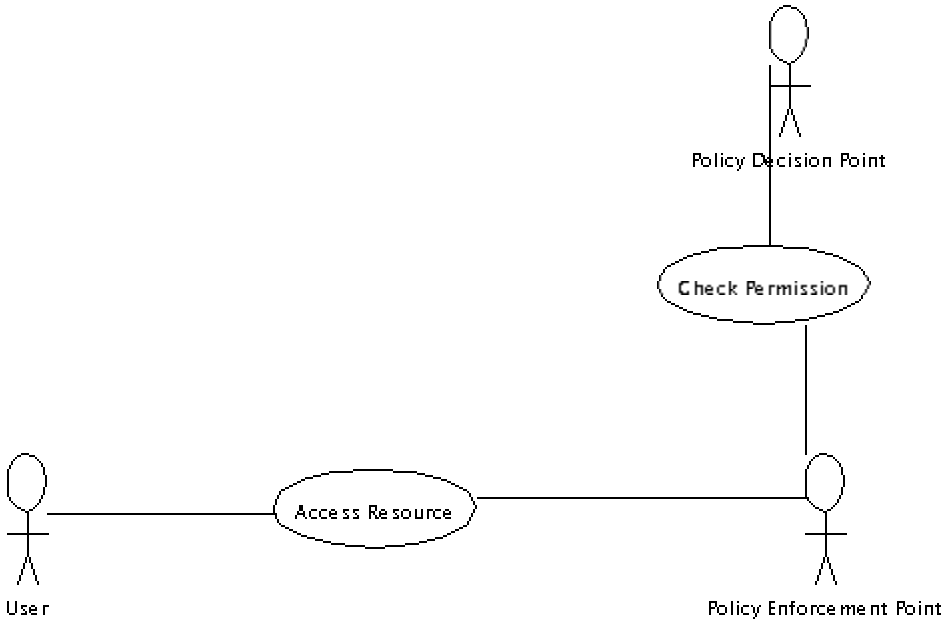



Fig 5. Authorization Service.

Steps:

1. User accesses a resource controlled by PEP.

2. PEP checks permission for user to access resource with PDP.

### Scenario 2-1: Application Chain

This scenario illustrates using SAML within a security zone. A Web user requests a dynamic resource from a Web server. The Web server passes authentication information to an application so that the application can check the user's authorization to execute a method.

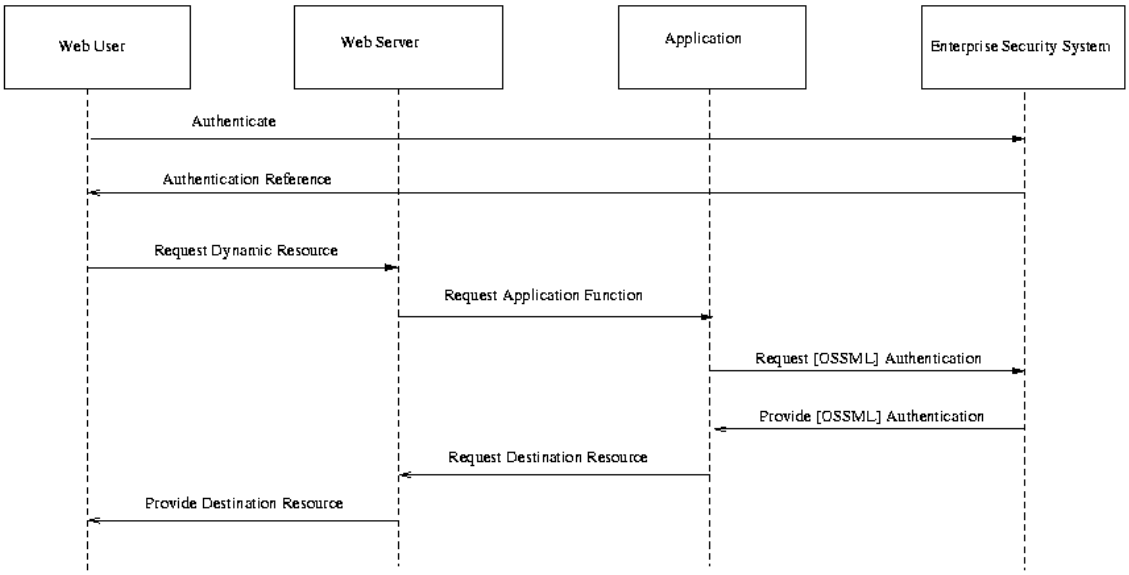



Fig 6. Application Chain.

Steps:

1. Web user authenticates with enterprise security system. Note that authentication may be through e.g. the Web server.
2. Enterprise security system provides an authentication reference to Web user.
3. Web user requests a dynamic resource from Web server, providing authentication reference.
4. Web server requests application function from application on behalf of Web user, providing Web user's authentication reference.
5. Application requests authentication document from enterprise security system, corresponding to Web user's authentication reference.
6. Enterprise security system provides authentication document, including authorization attributes for the Web user, and authn event description.
7. Application performs application function for Web server.
8. Web server generates dynamic resource for Web user.

Associated requirements: [R-AuthN], [R-PullMessage], [R-SingleDomain], [R-Bindings] (standard commercial browsers), [R-Reference].

## Use Case 3: Back Office Transaction

In this use case, two agents, a buyer and a seller, attempt to execute a business transaction.

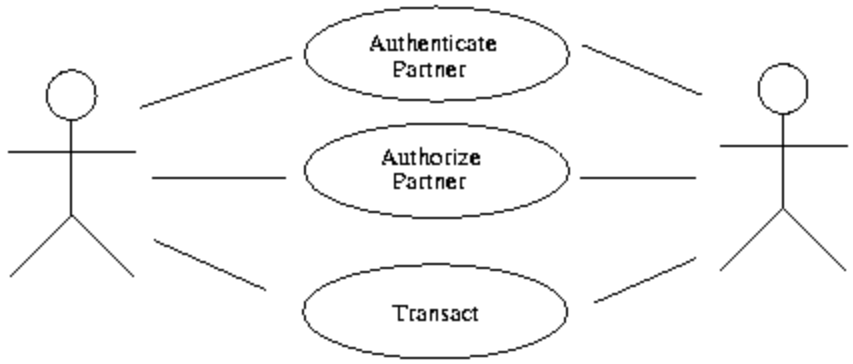



Fig 7. Back Office Transaction.

1. Buyer and seller authenticate that their partner in the transaction is the partner they expect to transact with.
2. Buyer and seller check permission of partner to execute transaction.
3. Buyer and seller execute the transaction.

### Scenario 3-1: Back Office Transaction

In this scenario, two parties, buyer and seller, wish to perform a transaction. Each authenticates to a security system responsible to their own security zone (buyer security system and seller security system, respectively). They exchange authentication data provided by their security systems to authenticate the transaction.

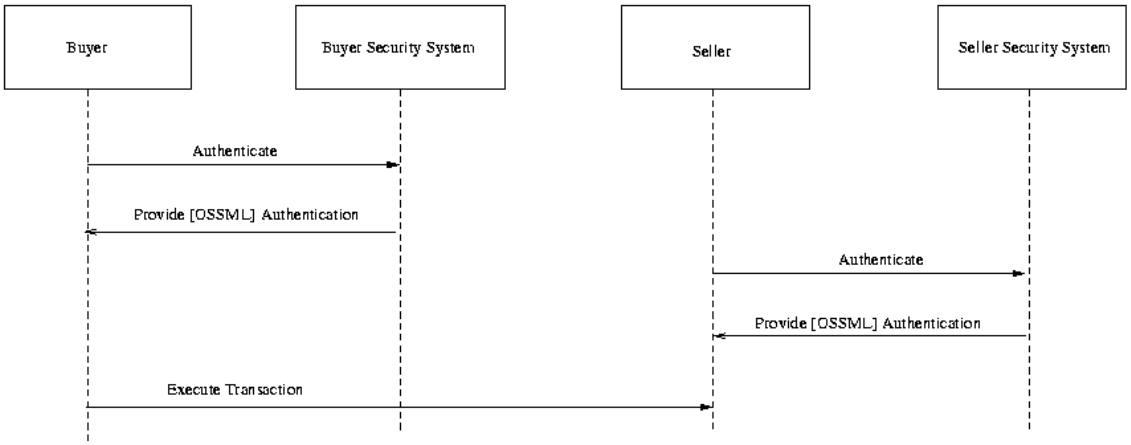



Fig 8. Back Office Transaction.

Steps:

1. Buyer authenticates with buyer security system.
2. Buyer security system provides authentication document to buyer.
3. Seller authenticates with seller security system.
4. Seller security system provides authentication document to seller.
5. Buyer and seller execute transaction, providing authentication documents to each other. Authentication documents include authz attributes and authn event description.

Associated requirements: [R-AuthN], [R-PushMessage], [R-AuthZ], [R-MultiDomain].

**Scenario 3-2: Back Office Transaction, Third-Party Security Service**

This scenario is similar to scenario 4. The same two parties, buyer and seller, wish to perform a transaction. In this case, however, each authenticates to a third-party security service responsible. The buyer and seller exchange authentication data provided by their security systems to authenticate the transaction.

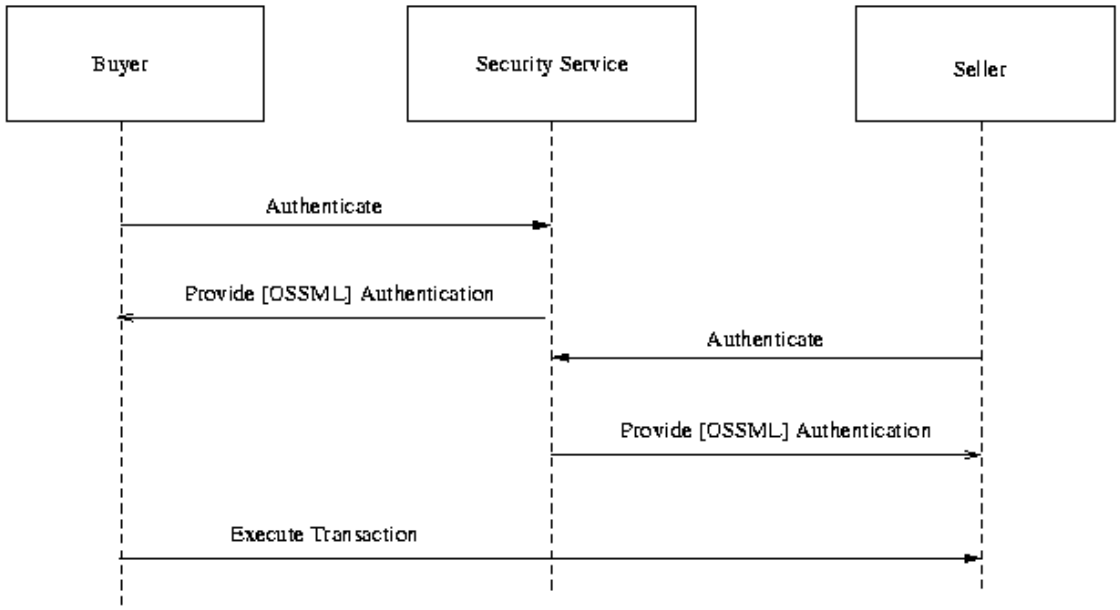



Fig 9. Back Office Transaction, Third Party Security Service.

Steps:

1. Buyer authenticates with security service.
2. Security service provides authentication document to buyer.
3. Seller authenticates with security service.

4. Security service provides authentication document to seller.

5. Buyer and seller execute transaction, providing authentication documents to each other. Authentication documents include authz attributes and authn event description.

Associated requirements: [R-AuthN], [R-AuthZ], [R-PushMessage].

# References

This document is derived from the following sources:

- Security Services Markup Language v0.8a, Prateek Mishra et. al.
- AuthXML: A Specification for Authentication Information In XML v0.3, Evan Prodromou et. al.

Other references that may be useful:

- Oasis Open Security Services Technical Committee, http://www.oasis-open.org/committees/security/index.shtml.
- Unified Modeling Language (UML), http://www.omg.org/uml/.

# Document History

- 25 Jan 2001 -- First draft derived from merge of S2ML and AuthXML specs.
- 9 Feb 2001 -- Second draft.
    - Incorporated comments from Use Case subcommittee of Oasis Security Services TC.
    - Added set of high-level use cases.
    - Changed diagrams of detailed use case scenarios to use sequence diagrams instead of use case diagrams.
    - Added description of each use-case scenario and list of requirements flowing from the scenario.
    - Added draft glossary (as link).
    - Added issues list (as link).
    - Gave requirements labelled names for easier reference.
    - Incorporated and merged requirements list from Core Assertions subcommittee of

320 Oasis Security Services TC (by Philip Hallam-Baker).

321 - Corrected various editorial mistakes.

322 - 26 Feb 2001 -- Third draft.

323 - Changed placeholder "[OSSML]" to new, official "SAML".

324 - Re-ordered scenarios so that each group of scenarios followed an associated use
325 case.

326 - Rephrased use case scenario 1-2 per Nigel Edwards.

327 - Updated use case scenario 1-3 per UC-1-02:ThirdParty.

328 - Added [R-Anonymity].

329 - Added [R-Pseudonymity].

330 - Noted exchange of authz attributes, per UC-1-08:AuthZAttrs.

331 - Added [R-AuthZDecision] and noted exchange of authz decisions, per UC-1-
332 09:AuthZDecisions.

333 - Edited [R-AuthN] and noted exchange of authn event data, per UC-1-
334 10:AuthNEvent.
335 Added user session use case, per UC-3-1.

# 336 Core Assertions

337 This section is currently empty.

# 338 Request/Response Protocols

339 ***Model***

340 The model contains eight elements:

341 The Principal,

342 The Primary Domain,

343 The Secondary Domain,

344 The Authentication Authority,

345 The Authorization Authority,

346 The Session Authority,

347 The Policy Enforcement Point, and

348 The Policy Decision Point.

349 The **Principal** is an entity that requires controlled access to resources in a Secondary Domain.

350 The **Primary Domain** is an administrative domain in which the Principal can be authenticated
351 without assistance from any other domain.

352 The **Secondary Domain** is an administrative domain in which the Principal cannot be
353 authenticated except with assistance from a Primary Domain.

354 The Principal has at least one name in a namespace sub-tree administered by the **Authentication**
355 **Authority** in the Primary Domain. The Authentication Authority binds the Principal's name to
356 an authentication mechanism in a "name assertion".

357 The Principal may have one or more entitlements in an entitlement-space sub-tree administered
358 by the **Authorization Authority** in the Primary Domain. The Authorization Authority binds the
359 Principal's entitlements to a name assertion in an "entitlement assertion".

360 The Principal may have a session state in a session state-space sub-tree administered by the
361 **Session Authority**. The Session Authority binds the Principal's session state to a name assertion
362 in a "session assertion".

363 The **Policy Enforcement Point** authenticates the Principal with the assistance of a Policy
364 Decision Point and controls its access to resources in the Secondary Domain.

365 The **Policy Decision Point** authenticates the Principal and determines its eligibility to access
366 resources in the Secondary Domain on the basis of the assertions.

367 Figure 1 indicates which elements of the model communicate with which other elements.

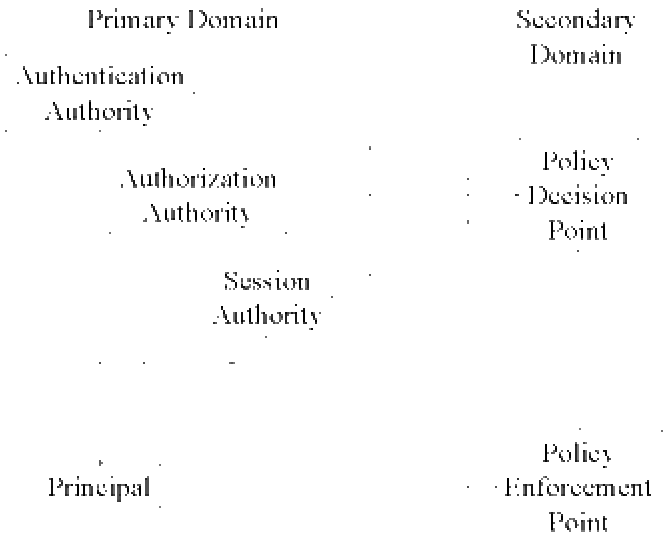



**Figure 1 - Model**

There are seven authentication data structures:

AuthnNotification,

AuthnAcknowlegment,

AuthnRequest,

AuthnResponse,

AuthnQuery,

AuthnResult and

Ref(AuthnNotification).

There are seven authorization data structures:

AuthzNotification,

AuthzAcknowlegment,

AuthzRequest,

AuthzResponse,

AuthzQuery,

AuthzResult and

385 Ref(AuthzNotification).

386 There are seven session data structures:

387 SessionNotification,

388 SessionAcknowlegment,

389 SessionRequest,

390 SessionResponse,

391 SessionQuery,

392 SessionResult and

393 Ref(SessionNotification).

394 For the purpose of explaining the model, only the authentication protocols will be described; the
395 authorization and session data structures are used in an analogous fashion. In the authorization
396 variants, the Policy Decision Point is responsible for obtaining the authorization policy definition
397 appropriate to the specified action and the environmental variables appropriate to the policy.
398 These two data structures are out of scope for the current version of the specification.

399 The Ref(AuthnNotification) data structure is defined in the Bindings section of the specification,
400 not in this, the Protocols, section. The step in which the Principal authenticates itself to the
401 Policy Enforcement Point is not defined in this specification. However, it is a requirement of this
402 step that it provide a posited name for the Principal and an authenticator. The posited name shall
403 include a domain name, identifying the Authentication Authority in the Principal's Primary
404 Domain, and a Principal name. The authenticator may be in any one of a number of forms,
405 including a password, a symmetric-key challenge/response pair, an asymmetric-key
406 challenge/response pair or a document/signature pair.

407 Discovery of services in a remote domain is outside the scope of this specification.

408 ### *Protocol exchanges*

409 **Principal-centered direct protocol**

410 This protocol may be used when the Principal is capable of relaying messages of unlimited
411 length between the Primary Domain and the Secondary Domain, and when the Secondary
412 Domain is not capable of communicating with the Primary Domain directly at the time at which
413 the Principal communicates with the Secondary Domain.

414 Figure 2 shows the Principal-centered direct protocol.

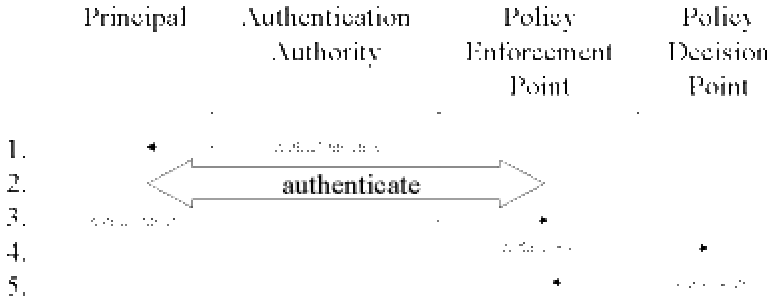



**Figure 2 - Principal-centered direct protocol**

It proceeds by the following steps.

1. The Principal obtains a name assertion from an Authentication Authority in the Primary Domain in an AuthnNotification message. The authentication of the Principal by the Authentication Authority is outside the scope of this specification.

2. The Principal conducts an authentication exchange with the Policy Enforcement Point. However, the Policy Enforcement Point is not capable of completing the authentication without the help of the Policy Decision Point.

3. The Principal provides the name assertion in an AuthnNotification message.

4. The Policy Enforcement Point sends the posited name, the authenticator and the name assertion to the Policy Decision Point in an AuthnQuery message.

5. The Policy Decision Point authenticates the Principal using the posited name, authenticator and name assertion provided in step 4 and returns the result to the Policy Enforcement Point in an AuthnResult message.

### Principal-centered indirect protocol

This protocol may be used when the Principal is only capable of relaying messages of limited size from the Primary Domain to the Secondary Domain and the Secondary Domain is capable of communicating with the Primary Domain at the time at which the Principal communicates with the Secondary Domain.

Figure 3 shows the Principal-centered indirect protocol.

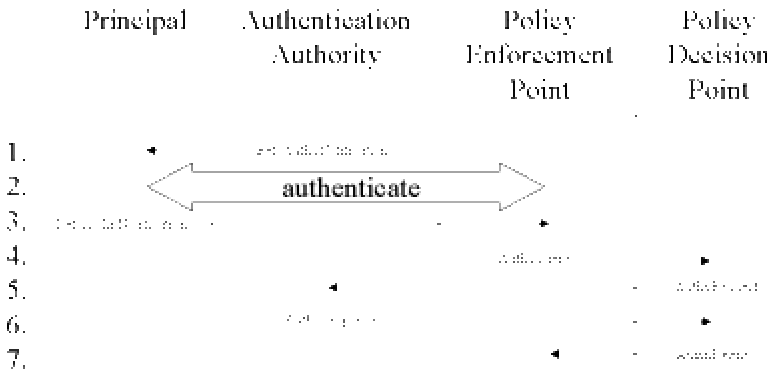



**Figure 3 - Principal-centered indirect protocol**

It proceeds by the following steps.

1. The Principal obtains a reference to a name assertion from an Authentication Authority in the Primary Domain in the Ref(AuthnNotification) message. As in the previous protocol, the authentication of the Principal by the Authentication Authority is out of scope.

2. The Principal conducts an authentication exchange with the Policy Enforcement Point. As before, the Policy Enforcement Point is not capable of completing the authentication without the help of the Policy Decision Point.

3. The Principal provides the reference to the name assertion in the Ref(AuthnNotification) message.

4. The Policy Enforcement Point sends the posited name, the authenticator and the reference to the name assertion to the Policy Decision Point in the AuthnQuery message.

5. The Policy Decision Point sends a request for the name assertion to the Authentication Authority in the Primary Domain in the AuthnRequest message.

6. The Authentication Authority sends the name assertion in an AuthnResponse message.

7. The Policy Decision Point authenticates the Principal and returns the result to the Policy Enforcement Point in an AuthnResult message.

### Pull protocol

This protocol may be used when the Principal communicates with the Secondary Domain without being directed by the Primary Domain.

Figure 4 shows the pull protocol.

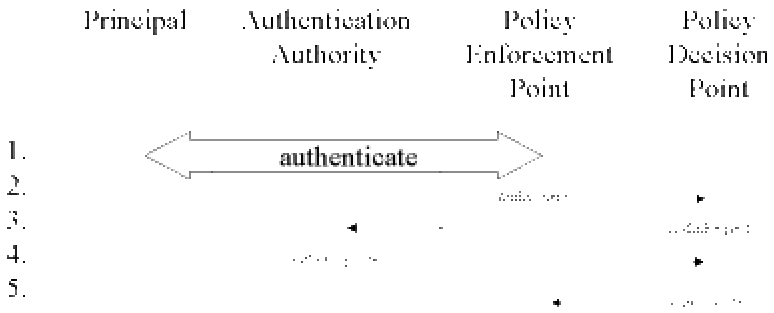



**Figure 4 - Pull protocol**

It proceeds by the following steps.

1. The Principal conducts an authentication exchange with the Policy Enforcement Point. As before, the Policy Enforcement Point is not capable of completing the authentication without the help of the Policy Decision Point.

2. The Policy Enforcement Point sends the posited name and the authenticator to the Policy Decision Point in the AuthnQuery message.

3. The Policy Decision Point sends a request for the name assertion to the Authentication Authority in the Primary Domain.

4. The Authentication Authority sends the name assertion in an AuthnResponse message.

5. The Policy Decision Point authenticates the Principal using the posited name and authenticator obtained from the Policy Enforcement Point in step 2 and the name assertion obtained from the Authentication Authority in step 4 and returns the result to the Policy Enforcement Point in the AuthnResult message.

### Push protocol

This protocol may be used when the Principal communicates with the Secondary Domain under the direction of the Primary Domain. Because it requires the Policy Decision Point to maintain state between communication sessions with the Authentication Authority and the Principal, it is less favoured than the Principal-centered protocols.

Figure 5 shows the Push protocol.

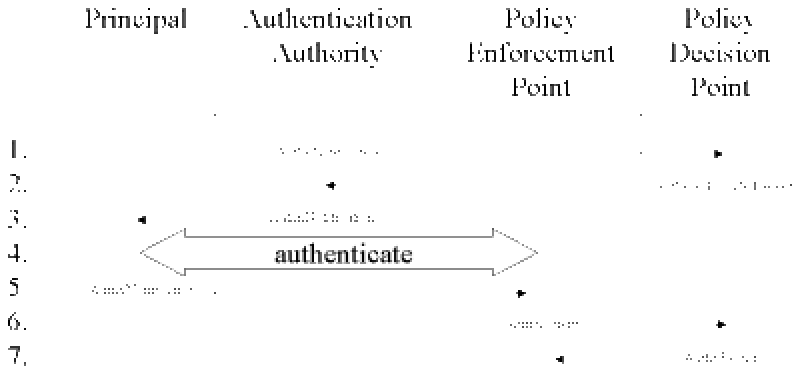



**Figure 5 - Push Protocol**

It proceeds by the following steps.

1. The Authentication Authority sends a name assertion in an AuthnNotification message to the Policy Decision Point in the Secondary Domain.

2. The Policy Decision Point sends an acknowledgment for the name assertion, including a reference, to the Authentication Authority in the Primary Domain in an AuthnAcknowledgment message.

3. The Authentication Authority sends the reference to the Principal in an AuthnNotification message.

4. The Principal conducts an authentication exchange with the Policy Enforcement Point. As before, the Policy Enforcement Point is not capable of completing the authentication without the help of the Policy Decision Point.

5. The Principal sends the reference to the Policy Enforcement Point in an AuthnNotification message.

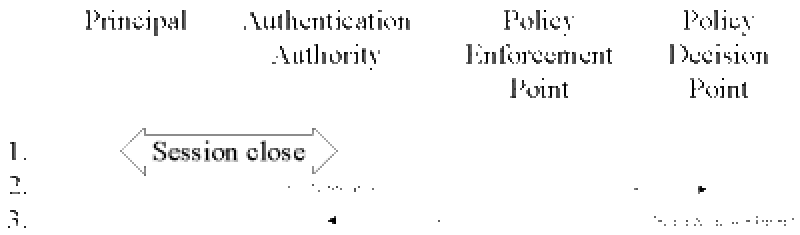
6. The Policy Enforcement Point sends the posited name and the authenticator to the Policy Decision Point in an AuthnQuery message.

7. The Policy Decision Point authenticates the Principal using the name assertion obtained in step 2 and the posited name and authenticator obtained in step 4 and returns the result to the Policy Enforcement Point in an AuthnResult message.

### Primary domain session-close protocol

This protocol may be used to notify Secondary Domains when a Principal logs off in the Primary Domain.

Figure 6 shows the Primary Domain session-close protocol.




507

508 **Figure 6 - Primary domain session close protocol**

509 It proceeds by the following steps.

510 1. The Principal closes the existing session with the Authentication Authority.

511 2. The Authentication Authority sends a SessionNotification message to the Policy Decision
512 Point in the Secondary Domain indicating that the Principal has closed the session.

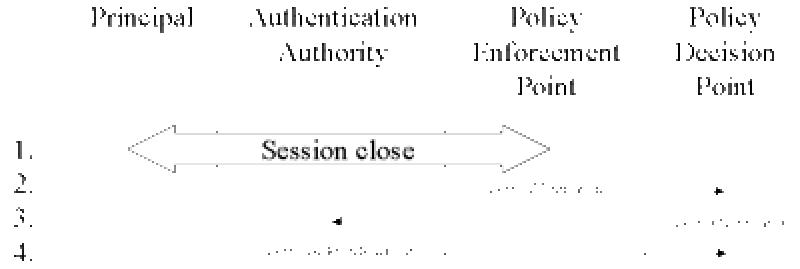
513 3. The Policy Decision Point sends an acknowledgment to the Authentication Authority in
514 the Primary Domain using the SessionAcknowledgment message.

515 Note: the Policy Enforcement Point should confirm the session status of the Principal with the
516 Policy Decision Point before processing each exchange between itself and the Principal. In this
517 way, the session closure will be effective immediately.

518 ### Secondary domain session-close protocol

519 This protocol may be used when the Principal logs off in the Secondary Domain.

520 Figure 7 shows the Secondary Domain session-close protocol.




521

522 **Figure 7 - Secondary domain session close protocol**

523
524
525
526
527

It proceeds by the following steps.

1. The Principal closes the existing session with the Policy Enforcement Point.

2. The Policy Enforcement Point notifies the Policy Decision Point in a SessionNotification message.

3. The Policy Decision Point sends a SessionnNotification message to the Authentication Authority in the Primary Domain, indicating that the Principal has closed the session.

4. The Authentication Authority sends a SessionAcknowledgment message to the Policy Decision Point in the Secondary Domain.

### *Data structures*

Note: there are separate data structures for authentication, authorization and session exchanges. If an entity needs information on any combination of name, entitlements and session status, it must conduct separate protocols for each. However, these separate protocols may proceed in parallel.

All the data structures incorporate an "extension" field. In the current version of the specification no extensions are defined. Therefore, the extension field must be empty. However, in future versions, the extension may be used to convey policy information or privacy-related release-authorization information, etc.. At such time, this enhanced functionality may be added without disturbing the core structure of the messages

Schema for the data structures can be found in the Schema section of this specification.

#### AuthnNotification

The AuthnNotification message is used in the Principal-centered direct authentication protocol to send the name assertion from the Authentication Authority to the Principal and from the Principal to the Policy Enforcement Point. It is also used in the Push protocol to send the name assertion from the Authentication Authority to the Policy Decision Point. It contains the following information.

version - this specification version number.

notification-identifier - an identifier assigned by the message originator. It must be unique among all the outstanding AuthnNotification messages.

name-assertion - the name assertion. Optional.

reference - reference to the name assertion. Optional, if the name assertion is absent, then the reference must be present.

sender - the name of the sender, as agreed between the sender and receiver during

initialization. It must be unique among all the sender names recognized by the receiver.

intended-receiver - the name of the receiver, as agreed between the sender and receiver during initialization. It must be unique among all the receiver names recognized by the sender. Optional.

extension

Note: the name assertion contains identifiers for the Authentication Authority and the Principal. It also includes validity dates and authentication information (e.g. a public key).

### AuthnAcknowlegment

The AuthnAcknowlegment message is used in the Push protocol for the Policy Decision Point to acknowledge receipt of the name assertion from the Authentication Authority. It contains the following information.

version - this specification version number.

notification-identifier - the notification identifier supplied in the corresponding AuthnNotification message.

success-indicator - an indication of whether the receiver was able to process the AuthnNotification message.

reference - a reference to the name assertion. Optional.

sender

intended-receiver -

error-code - error code.

The following error codes shall be supported.

Unsupported version

Unsupported authentication method

### AuthnRequest

The AuthnRequest message is used in the Principal-centered indirect protocol and the Pull protocol for the Policy Decision Point to request the name assertion from the Authentication Authority. It contains the following information.

version - this specification version number.

request-identifier - an identifier assigned by the message originator. It must be unique

among all the outstanding AuthnRequest messages.

posited-name - the Primary Domain and Principal names claimed by the Principal. Optional.

reference to name assertion - a reference to the name assertion. Optional, if the posited name is not present, then this field must be present.

sender - the name of the sender, as agreed between the sender and receiver during initialization. It must be unique among all the sender names recognized by the receiver.

intended-receiver - the name of the receiver, as agreed between the sender and receiver during initialization. It must be unique among all the receiver names recognized by the sender. Optional.

Note: the Authentication Authority receives no evidence that the Principal has correctly authenticated to the Policy Enforcement Point.

### AuthnResponse

The AuthnResponse message is used in the Principal-centered indirect protocol and the Pull protocol for the Authentication Authority to return the name assertion to the Policy Decision Point. It contains the following information.

version - this specification version number.

request-identifier - the request identifier supplied in the corresponding AuthnRequest message.

name-assertion - the name assertion.

success indicator

sender -

intended-receiver -

error code

### AuthnQuery

This protocol is used in the Principal-centered direct and indirect protocols and the Pull and Push protocols for the Policy Enforcement Point to request the Policy Decision Point to perform the authentication of the Principal.

version - this specification version number.

request-identifier - an identifier assigned by the message originator. It must be unique

among all the outstanding AuthnQuery messages.

posited name - the name claimed by the Principal.

authenticator - the data used in the authentication exchange between the Policy Enforcement Point and the Principal. This may be a user-name/password combination, a symmetric-key challenge/response combination, an asymmetric-key challenge response combination or a document/signature combination.

name-assertion - the name assertion. Optional.

reference to name assertion - a reference to a name assertion. Optional, at least one of "posited name", "name assertion" or "reference to name assertion" must be present.

sender -

intended-receiver -

### AuthnResult

This protocol is used in the Principal-centered direct and indirect protocols and the Pull and Push protocols for the Policy Decision Point to return the result of the authentication of the Principal to the Policy Enforcement Point.

version - this specification version number.

request-identifier - the request identifier from the corresponding AuthnQuery message.

success indicator

sender -

intended-receiver -

error code

### AuthzNotification

The AuthzNotification message is used in the Principal-centered direct authorization protocol to send the entitlement assertion from the Authorization Authority to the Principal and from the Principal to the Policy Enforcement Point. It is also used in the Push protocol to send the entitlement assertion from the Authorization Authority to the Policy Decision Point. It contains the following information.

version - this specification version number.

notification-identifier - an identifier assigned by the message originator. It must be unique among all the outstanding AuthzNotification messages.

entitlement-assertion - the entitlement assertion. Optional.

reference - reference to entitlement assertion. Optional, if the entitlement assertion is absent, then the reference must be present.

sender - the name of the sender, as agreed between the sender and receiver during initialization. It must be unique among all the sender names recognized by the receiver.

intended-receiver - the name of the receiver, as agreed between the sender and receiver during initialization. It must be unique among all the receiver names recognized by the sender.

Note: the entitlement assertion contains an identifier for the Authorization Authority and a reference to the associated Principal name-assertion. It also contains validity dates.

### AuthzAcknowlegment

The AuthzAcknowlegment message is used in the Push protocol for the Policy Decision Point to acknowledge receipt of the entitlement assertion from the Authorization Authority. It contains the following information.

version - this specification version number.

notification-identifier - the notification identifier supplied in the corresponding AuthzNotification message.

reference - reference to the entitlement assertion. Optional.

success-indicator - an indication of whether the receiver was able to process the AuthzNotification message.

sender -

intended-receiver -

error-code - error code.

### AuthzRequest

The AuthzRequest message is used in the Principal-centered indirect protocol and the Pull protocol for the Policy Decision Point to request the entitlement assertion from the Authentication Authority. It contains the following information.

version - this specification version number.

request-identifier - an identifier assigned by the message originator. It must be unique among all the outstanding AuthzRequest messages.

679 posited name - the posited name of the Principal. Optional.

680 reference to entitlement assertion - reference to an entitlement assertion. Optional. If the
681 posited name is absent, then this field must be present.

682 sender - the name of the sender, as agreed between the sender and receiver during
683 initialization. It must be unique among all the sender names recognized by the receiver.

684 intended-receiver - the name of the receiver, as agreed between the sender and receiver
685 during initialization. It must be unique among all the receiver names recognized by the
686 sender. Optional.

687 Note: the Authorization Authority receives no evidence that the Principal correctly authenticated
688 to the Policy Enforcement Point. In the Pull protocol, all suitable entitlement assertions are
689 requested.

690 ### AuthzResponse

691 The AuthzResponse message is used in the Principal-centered indirect protocol and the Pull
692 protocol for the Authorization Authority to return the entitlement assertion to the Policy Decision
693 Point. It contains the following information.

694 version - this specification version number.

695 request-identifier - the request identifier supplied in the corresponding AuthzRequest
696 message.

697 entitlement assertion - the entitlement assertion.

698 sender -

699 intended-receiver -

700 success indicator

701 error code

702 ### AuthzQuery

703 This protocol is used in the Principal-centered direct and indirect protocols and the Pull and Push
704 protocols for the Policy Enforcement Point to request the Policy Decision Point to confirm the
705 authorization of the Principal.

706 version - this specification version number.

707 request-identifier - an identifier assigned by the message originator. It must be unique
708 among all the outstanding AuthzQuery messages.

action - a compound variable comprising the name of the object method and a sensitivity value for the object that the Principal is attempting to access.

principal name - the authenticated or claimed name of the Principal. Optional. Must be identical to the posited name in any accompanying AuthnQuery message.

entitlement-assertion - the entitlement assertion. Optional.

reference to the entitlement assertion - a reference to the entitlement assertion. Optional, it should be present if the entitlement assertion is absent. Optional. At least one of "principal name", "entitlement assertion" or "reference to entitlement assertion" must be present.

sender -

intended-receiver -

### AuthzResult

This protocol is used in the Principal-centered direct and indirect protocols and the Pull and Push protocols for the Policy Decision Point to return the result of the authorization of the Principal to the Policy Enforcement Point.

version - this specification version number.

request-identifier - the request identifier supplied in the corresponding AuthzRequest message.

sender -

intended-receiver -

success indicator

error code

### SessionNotification

The SessionNotification message is used in the Principal-centered direct session protocol to send the session assertion from the Session Authority to the Principal and from the Principal to the Policy Enforcement Point. It is also used in the Push protocol to send the session assertion from the Session Authority to the Policy Decision Point. It is also used in the Primary Domain session close and Secondary Domain session close protocols to indicate that the session with the Principal has been closed. It contains the following information.

version - this specification version number.

notification-identifier - an identifier assigned by the message originator. It must be

unique among all the outstanding SessionNotification messages.

session-assertion - the session assertion. Optional.

reference - reference to the session assertion. Optional, if the session assertion is absent, then the reference must be present.

sender - the name of the sender, as agreed between the sender and receiver during initialization. It must be unique among all the sender names recognized by the receiver.

intended-receiver - the name of the receiver, as agreed between the sender and receiver during initialization. It must be unique among all the receiver names recognized by the sender. Optional.

Note: the session assertion identifies the Principal either directly or by reference to a name assertion. It also contains an indication of the Principal's session state (e.g. "session closed").

### SessionAcknowlegment

The SessionAcknowlegment message is used in the Push protocol for the Policy Decision Point to acknowledge receipt of the session assertion from the Session Authority. It is also used in the Primary Domain session close and Secondary Domain session close protocols to acknowledge that the session with the Principal has been closed. It contains the following information.

version - this specification version number.

notification-identifier - the notification identifier supplied in the corresponding SessionNotification message.

Reference - reference to the session assertion. Optional.

sender -

intended-receiver -

success-indicator - an indication of whether the receiver was able to process the SessionNotification message.

error-code - error code.

The following error codes shall be supported.

Unsupported version

### SessionRequest

The SessionRequest message is used in the Principal-centered indirect protocol and the Pull protocol for the Policy Decision Point to request the session assertion from the Session

Authority. It contains the following information.

version - this specification version number.

request-identifier - an identifier assigned by the message originator. It must be unique among all the outstanding SessionRequest messages.

principal name - the name of the Principal. Optional.

reference to session assertion - reference to the session assertion. Optional, is the principal name field is absent, then this field must be present.

sender - the name of the sender, as agreed between the sender and receiver during initialization. It must be unique among all the sender names recognized by the receiver.

intended-receiver - the name of the receiver, as agreed between the sender and receiver during initialization. It must be unique among all the receiver names recognized by the sender. Optional.

Note: the Session Authority receives no evidence that the Principal correctly authenticated to the Policy Enforcement Point.

### SessionResponse

The SessionResponse message is used in the Principal-centered indirect protocol and the Pull protocol for the Session Authority to return the session assertion to the Policy Decision Point. It contains the following information.

version - this specification version number.

request-identifier - the notification identifier supplied in the corresponding SessionRequest message.

session-assertion - the session assertion.

success indication

error code

### SessionQuery

This protocol is used in the Principal-centered direct and indirect protocols and the Pull and Push protocols for the Policy Enforcement Point to request the Policy Decision Point to confirm the session status of the Principal.

version - this specification version number.

request-identifier - an identifier assigned by the message originator. It must be unique

among all the outstanding SessionQuery messages.

principal name - the authenticated or claimed name of the Principal. Optional. Must be identical to the posited name in any associated AuthnQuery message.

session assertion - a session assertion. Optional.

reference to session assertion - a reference to a session assertion. Optional, at least one of "principal name", "session assertion" or "reference to session assertion" must be present.

Sender -

intended-receiver -

### SessionResult

This protocol is used in the Principal-centered direct and indirect protocols and the Pull and Push protocols for the Policy Decision Point to return the result of the status evaluation of the Principal to the Policy Enforcement Point.

version - this specification version number.

request-identifier - the identifier from the corresponding SessionQuery message.

session assertion

sender -

intended-receiver -

success indicator

error code

Note: the session assertion returned in the SessionResult message may be integrity-protected by means other than XML Digital Signature. Alternatively, it may protected by the XML Digital Signature mechanism, signed by the Policy Decision Point.

#### *Security considerations*

With the exception of the session assertion in the SessionResult message, all assertions must be protected for integrity and authenticity using the XML Digital Signature mechanism. In addition, all protocol exchanges must be protected for integrity and authenticity. Mechanisms other than XML Digital Signature may be used for this latter purpose.

The exchange of Authority keys, certificates and certificate status information between domains is out of scope for this specification.

# Bindings

## Introduction

The purpose of this document 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 concall or two. The spirit of including this subsection is therefore for present completness' 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.

# 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

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

[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.html

[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

# Security Considerations

This section is currently empty.

# Glossary

**Modification Log**

| Date | By Whom | What |
|---|---|---|
| 21 Jan 2001 v00 | Jeff Hodges | Created. |
| 8 Feb 2001 v01 | Jeff Hodges | Added variouis terms supplied by Bob Blakley and others culled from S2ML 0.8a doc. |
| 9 Feb 2001 v01 | Jeff Hodges | Cleaned up refs, added refs, added definitions, enhanced or otherwise mangled others. |

Many of the definitions in this glossary are based on those found in these references: _edn1_edn1[1], _edn2_edn2[2], _edn3_edn3[3], _edn4_edn4[4], _edn5_edn5[5] (page 57), _edn6_edn6[6], _edn7_edn7[7] (Appendix K *Glossary*), _edn8_edn8[8], _edn9_edn9[9], _edn10_edn10[10], _edn11_edn11[11], _edn12_edn12[12], _edn13_edn13[13], _edn14_edn14[14], _edn15_edn15[15], _edn16_edn16[16], _edn17_edn17[17], _edn18_edn18[18], _edn19_edn19[19], _edn20_edn20[20],_edn21_edn21[21], _edn22_edn22[22], _edn23_edn23[23], _edn24_edn24[24], _edn25_edn25[25], _edn26_edn26[26], , , , , , , , -- to one degree or another. 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. Occasionally, definitions are quoted directly from the sources and the source(s) is (are) referenced.

Definitions to be added or otherwise enhanced are marked with a **?**

| | |
|---|---|
| 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. |
| ACI | See Access Control Information |

| | |
|---|---|
| ADF | See Access Decision Function |
| ADI | See Access Decision Information |
| AEF | See Access 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 functionality. |
| 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. [23]<br><br>This component may act in these roles: |
| AAA System | A set of AAA system components implementing a network service 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 service to a given administrative domain, It also may be utilized to provide services to other administrative domains. |
| Access | The ability and means to communicate with or otherwise interact with a system in order to use system resources to either handle information or gain knowledge of the information the system contains. (definition from [1] ) |
| 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. (definition from [1] ) |

| | |
|---|---|
| | 2. The prevention of unauthorized use of a resource, including the prevention of use of a resource in an unauthorized manner [9] |
| 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 [10] ) |
| Access Control Information | Any information used for access control purposes, including contextual information [10]. |
| Access Control Factors | A request, when it is being processed by a server, may be associated with a wide variety of security-related *factors* (e.g. section 4.2 of [17]). 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". [25] |
| Access Control Policy | The set of rules that define the conditions under which an access may take place [10]. |
| Access Control Policy Rules | Security policy rules concerning the provision of the access control service [10]. |
| Access Control Request | See access request. |
| Access Decision Function | A specialized function that makes access control decisions by applying access control policy rules to an access request, Access Decision Information (of initiators, targets, access requests, or that retained from prior decisions), and the context in which the access request is made [10]. |
| Access 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 [10]. |
| Access Enforcement Function | A specialized function 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]. |
| Access Path | **?**(haven't been able to find a concise def for this with a modicum of |

| | |
|---|---|
| | looking**)** |
| Access Request | the operations and operands that form part of an attempted access. [10] |
| Active Role | **?** A role that an actor has donned when performing some operation, e.g. accessing a resource. |
| Actor | **?** From [2]: 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.<br><br>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). 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, business administrator, and local administrator. |
| Anonymity | The quality or state of being anonymous. |
| Anonymous | The condition of having a name [or identity] that is unknown or concealed. [1] |
| 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 piece of data constituting a declaration of identity or authorizations. See also: credential. |

| | |
|---|---|
| | "Data that is transferred to establish the claimed identity of an entity." [9] |
| Asserting Party | **?** An AAA system component performing a role wherein it generates assertions on behalf of other actors. |
| 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. (definition from [1]). |
| 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.<br><br>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. [11][17] |
| 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). [12] |
| 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. [17] |
| Attribute Value | An attribute value is one or more pieces of data, encoded according to the syntax of the attribute's type. [17] |
| Attribute Value Assertion | An Attribute Value Assertion is an assertion with the general abstract form of "attribute type IS attribute value". [17] |
| 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. [8] |

| | |
|---|---|
| Audit Identity | An identity attribute containing an identity used only for accountability purposes (ECMA 219). [13] |
| Authc | See Authentication |
| Authn | See Authentication |
| Authz | See Authorization |
| Authentication | Authentication is the process of confirming an entity's asserted identity with a specified, or understood, level of confidence. [7]<br><br>The process of verifying an identity claimed by or for a system entity. [12] |
| Authority | An identified computer-based entity which implements a security service (e.g. creation of PACs). [12] |
| Authorization | **?** The process of determining what 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. [8]<br><br><roughThe "act of authorization" is when an AEF acts upon information received from an ADF.</rough<br><br>The granting of access rights to a subject (for example, a user, or program). [12] |
| Authorization Assertion | **?** In concept an authorization assertion is a statement of policy about a resource, such as:<br><br>the user "noodles" is granted "execute" privileges on the resource "/usr/bin/guitar." |
| Authorization Identity | **?** import from rfc2829 and rfc2222 |
| 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. [12] |
| Clearance | Initiator-bound ACI that can be compared with security labels of targets [10]. |
| Context | See Contextual Information. |
| Contextual Information | Information about or derived from the context in which an access request is made (e.g. time of day). [10]. 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. [19]**?** |
| Credential | 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.) [1]<br><br>"Data that is transferred to establish the claimed identity of an entity." [9] |
| Decision | The response of an Access Decision Function to a decision request[12]. |
| Decision Request | The message an Access Enforcement Function sends to an Access Decision Function to ask it whether a particular access request should be granted or denied [12]. |
| 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. |
| Deployment Time | The time at which a product 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 [6] and DMZ network. |
| DMZ network | DMZ network is a commonly-used, equivalent term for (see also) perimeter network. |
| DNS | See Domain Name System. |

| | |
|---|---|
| Domain Name System | The general-purpose distributed, replicated, data query service used on the Internet for translating host names into Internet addresses. See [6]. |
| 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 | Variouis 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. |
| End User System | Typically the combination of: an End User, plus the End User's computer, plus the browser running on that computer. The term "EU System" is used in this document, rather than just the terms "client" or "user" because given the many-tiered architecture, there are many components that act as clients of other components. |
| Entitlement | A data structure containing Access Decision Information and/or access control policy rule information in a form which can be used by applications to customize their behavior based on access control policy or to make access control decisions in their own code [12]. |
| Entity | See System Entity. |
| EU System | A contraction for 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 (definition from [14] ). See also extranet in [6]. |
| 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 existence of a firewall. |
| 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. |
| HTTP | See Hypertext Transfer Protocol. |
| Hypertext Transfer Protocol | A protocol for distributed, collaborative, hypermedia information systems. It is the protocol used by web browsers to communicate with web servers, when the browsers process URLs specified as "http://host…". See also RFC1945 [15] and RFC2616. [16] |
| Identity | A representation (e.g. a string) uniquely mapped to an entity (e.g. an end user, an administrator, or some process, or some network device).<br><br>Initiator ACI passed to the aznAPI. [aznAPI] uses the term to describe anything used as initiator ACI, including names, identity certificates, and capabilities. Note that this usage is unique to [aznAPI] and should not be confused with other uses of the term "identity" in other systems [12]. |
| IETF | See Internet Engineering Task Force. |
| Initiator | An entity (e.g. human user or computer-based entity) that *attempts to access* other entities [10]. |
| Intermediary | An entity which, after receiving an access request from an initiator, issues another access request on that initiator's behalf [12]. |
| Internal Network | See Intranet. |
| Intranet | A local area network which 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. (definition from [14]) See also intranet in [6]. |
| IP | Internet Protocol. See also TCP/IP. |

| | |
|---|---|
| Label | A marking that is bound to a protected resource and that names or designates the security-relevant attributes of that resource (derived from [9]). |
| LDAP | See Lightweight Directory Access Protocol. |
| Lightweight Directory Access Protocol | A directory access protocol defined in IETF RFCs 2251..2256 (for LDAP version 3). It is largely based on X.500. [17] |
| MIME | Multipurpose Internet Mail Extensions [18] -- a standard for imparting structure within otherwise “flat” ascii text. |
| 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 [5]. |
| 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, or application servers. (definition largely from [6]) |
| Network Topology | A configuration of network devices and hosts, and their interconnections. |
| Operation | The action that an initiator’saccess request asks to have performed on a protected resource [12]. |
| Origin Server | The server on which a given resource resides or is to be created. |
| Origin Site, Originating Site | **?** The site where the origin server resides. |
| PAC | See Privilege Attribute Certificate. |
| Package | = assertions [+ entitlements] + payload **?** |
| Party | **?** An actor or actors participating in some process, such as accessing a resource. See also: 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. [26] |
| 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 [5]. |
| Perimeter Security | Network-based security applied at the perimeter of one's security domain. See also [5]. |
| Policy, Policies | Concisely, a policy is a mapping of user *credentials* with authority to act [8]. Policies are often essentially access control *lists* [8]. |
| Principal | **?** A uniquely named client or server instance that participates in a network communication. [RFC1510] |
| 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). [19] |
| Privilege Attribute Certificate | A data structure containing privilege attributes. May be signed by the authority which generated it [12]. |
| Protected Resource | A target, access to which is restricted by an access control policy [12]. |
| Protected Web Resources | Web resources whose availability to requesters is being managed, i.e. protected, via some access control mechanism. |
| 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. |
| 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*. |
| Risk | (a) In the computer system and networking sense: An expectation of loss expressed as the probability that a particular threat will exploit a particular vulnerability with a particular harmful result. (b) More generally: possibility of loss or injury. |
| 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 [22]. |
| Role | **?**Dictionaries define a *role* as "a character or part played by a performer" or "a function or position." Actors *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. |
| Secure Sockets Layer | A network session-layer protocol which can be sandwiched between application-layer protocols, such as LDAP and HTTP, and the underlying transport protocol, TCP. SSL features facilities for mutual authentication of the client and server, as well as session encryption and integrity protection. See [20]. |
| 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. (definition from [7]) |
| 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. (definition largely from [1]) |
| 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. |
| Security Mechanism | The logic or algorithm that implements a particular security-enforcing or security-relevant function in hardware and software. [8] |
| Security Object | An entity in a passive role to which a security policy applies. [19] |
| Security Package | **?** one or more security assertions or credentials combined into a single overall, for example, MIME entity. |
| Security Perimeter | The boundary of a security domain. |
| Security Policy | 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). Significant portions of security policies are implemented via security services. Security policies are components of security architectures. |
| 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 ]. (definition from [8]) |
| 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. Security services typically implement portions |

| | |
|---|---|
| | of security policies, and are implemented by security mechanisms. |
| Security Subject | An entity in an active role to which a security policy applies. [19] |
| Server | Either (1) a host that is used for running applications and or services that are network-accessible. Servers are *typically not* also used as end users' computers. See also Server Host; or (2) a process or set of processes running on a host providing a network service. |
| 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 a server host. |
| Service | See Network Service. |
| Site | A term commonly used to refer to an administrative domain in a geographical sense. Thus site may refer to a particular geographical and/or topological subportion of an administrative domain, or, a site my 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 performs load balancing of incoming HTTP requests to web server hosts is site-specific. From the vendor's perspective, site-specific decisions are made at deployment time. |
| SSL | See Secure Sockets Layer. |
| 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. |
| System Entity | 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]) |
| System Resource | 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]) |
| Target | An entity to which access may be attempted [10].<br><br>A resource an entity attempts to access. |
| 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). (definition from [1], See especially [8]) |
| TCP or TCP/IP | See Transmission Control Protocol. |
| Ticket | **?** Aka a token. Specific example: Kerberos Tickets. See [RFC1510]. A ticket *may* be a credential. |
| TLS | See Transport Layer Security. |
| Token | **?** See ticket. |
| Transmission Control Protocol | The transport-layer protocol used on the Internet and most Internet-connected networks. It is layered on top of the Internet Protocol (IP) and the combination of the two is commonly termed “TCP/IP”. |
| Transport Layer Security | The IETF version of SSL 3.0. It is essentially/effectively regarded as SSL 3.1. It is specified in RFC2246. A small, but growing, number of servers and clients on the Internet at large presently support it. |
| Unauthorized | The opposite of a system entity or requester being authorized. |
| URL | See Uniform Resource Locator. |
| User | A corporeal human making use of a AAA system component and/or application(s) inhabiting a given administrative domain(s), *as a means* rather than as an end. (based on “user” from [6]). See also Administrator, End User. |
| User Profile or User’s Profile | See End User Profile. |

| | |
|---|---|
| Uniform Resource Locator | Defined as “a compact string representation for a resource available via the Internet.” See [21]. |
| 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. (definition from [1]) |
| 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”. |
| 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 [6]. 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. |

# Appendix A. References

_ednref1 _ednref1[1]**Authentication Markup Language – AuthXML.** Evan Prodromou, Darren Platt, Robert L. Grzywinski, Eric Olden, Third Draft - Version 0.3 - 12/14/2000.

Available at: http://www.authxml.org/**?** _ednref2 _ednref2 [2]**Security Services Markup Language (S2ML)**. P. Mishra, P. Hallam-Baker, Zahid Ahmed, Alex Ceponkus, Marc Chanliau, Jeremy Epstein, Chris Ferris, David Jablon, Eve Maler, David Orchard. Rev 0.8a, 8-Jan-2001.

Available at: http://www.s2ml.org/downloads/S2MLV08a.pdf _ednref3 _ednref3[3]**ITML**. Dave Orchard et al. Jamcraker 2001.

available at: **?** _ednref4_ednref4[4]**Internet Security Glossary**. Robert W. Shirey, RFC 2828, May 2000.

Available at: http://www.ietf.org/rfc/rfc2828.txt _ednref5_ednref5[5]**Building Internet Firewalls**. D. Brent Chapman & Elizabeth D. Zwicky, O'Reilly, ISBN 1-56592-124-0, September 1995.

Available at: http://www.oreilly.com/catalog/fire/ _ednref6_ednref6[6]**Free On-Line Dictionary of Computing**. Denis Howe, on-going.

Available at: http://foldoc.doc.ic.ac.uk/foldoc/ _ednref7_ednref7[7]**Trust in Cyberspace**. Committee on Information Systems Trustworthiness, Fred B. Schneider - Editor, National Research Council, ISBN 0-309-06558-5, 1999.

On-line copy and ordering information available at: http://www.nap.edu/readingroom/books/trust/ _ednref8_ednref8[8]**Security Taxonomy and Glossary**. Lynn Wheeler, on-going.

Available at: http://www.garlic.com/~lynn/secure.htm _ednref9_ednref9[9]**Information processing systems -- Open Systems Interconnection -- Basic Reference Model -- Part 2: Security Architecture**. ISO 7498-2:1989.

Available at: http://www.iso.ch/infoe/catinfo.html _ednref10_ednref10[10]**Information technology -- Open Systems Interconnection -- Security frameworks for open systems: Access control framework**. ISO/IEC 10181-3:1996.

Available at: http://www.iso.ch/infoe/catinfo.html _ednref11_ednref11[11]**Understanding and Deploying LDAP Directory Services**. Tim Howes, Mark Smith, and Gordon Good, Macmillan Technical Publishing & Netscape Communications Corporation, 1999, ISBN: 1578700701.

Description at: http://www.informit.com/product/1578700701/ _ednref12_ednref12[12]**Authorization (AZN) API**. Open Group Technical Standard, C908, ISBN 1-85912-266-3, January 2000.

Available at: http://www.opengroup.org/publications/catalog/c908.htm _ednref13_ednref13[13]**Authentication and Privilege Attribute Security Application with**

**related Key Distribution Functions - Part 1, 2 and 3**. Standard ECMA-219, 2nd edition (March 1996).

Available at: http://www.ecma.ch/ecma1/STAND/ECMA-219.HTM _ednref14_ednref14[14]**Computer Currents High-Tech Dictionary**. On-going

Available at: http://www.currents.net/resources/dictionary/ _ednref15_ednref15[15]**Hypertext Transfer Protocol -- HTTP/1.0**. T. Berners-Lee, R. Fielding, H. Frystyk, RFC1945, May 1996.

Available at: http://www.normos.org/ietf/rfc/rfc1945.txt _ednref16_ednref16[16]**Hypertext Transfer Protocol -- HTTP/1.1**. R. Fielding, J. Gettys, J. Mogul, H. Frystyk, T. Berners-Lee, RFC2616, June 1999.

Available at: http://www.normos.org/ietf/rfc/rfc2616.txt _ednref17_ednref17[17]**Lightweight Directory Access Protocol (v3).** M. Wahl, T. Howes, S. Kille, RFC2251, December 1997.

Available at: http://www.normos.org/ietf/rfc/rfc2251.txt _ednref18_ednref18[18]**Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies**. N. Freed, N. Borenstein, RFC2045, November 1996.

Available at: http://www.normos.org/ietf/rfc/rfc2045.txt _ednref19_ednref19[19]**Security in Open Systems - A Security Framework**. ECMA Technical Report TR/46, July 1988.

Available at: http://www.ecma.ch/ecma1/TECHREP/E-TR-046.HTM _ednref20_ednref20[20]**SSL 3.0 Specification**. Alan O. Freier, Philip Karlton, Paul C. Kocher, Netscape Communications Corp., 1996.

Available at: http://www.netscape.com/eng/ssl3/ _ednref21_ednref21[21]**Uniform Resource Locators (URL).** T. Berners-Lee, L. Masinter, M. McCahill, RFC1738, December 1994.

Available at: http://www.rfc-editor.org/rfc/rfc1738.txt _ednref22_ednref22[22]**Practical Unix & Internet Security, 2nd Edition**. Simson Garfinkel & Gene Spafford, O'Reilly, ISBN 1-56592-148-8, April 1996.

Available at: http://www.oreilly.com/catalog/puis/ _ednref23_ednref23[23]**AAA Authorization Framework**. J. Vollbrecht, P. Calhoun, S. Farrell, L. Gommans, G. Gross, B. de Bruijn, C. de

Laat, M. Holdrege, D. Spence. RFC 2904, August 2000.

Available at: http://www.rfc-editor.org/rfc/rfc2904.txt _ednref24_ednref24[24]**Uniform Resource Identifiers (URI): Generic Syntax**. T. Berners-Lee, R. Fielding, L. Masinter. RFC 2396, August 1998.

Available at: http://www.rfc-editor.org/rfc/rfc2396.txt _ednref25_ednref25[25]**Authentication Methods for LDAP**. M. Wahl, H. Alvestrand, J. Hodges, R. Morgan. May 2000.

Available at: http://www.rfc-editor.org/rfc/rfc2829.txt _ednref26_ednref26[26]**Whatis: IT-specific encyclopedia**. On-going.

Available at: http://whatis.techtarget.com/

# Issues

## Purpose

This document catalogs issues with the requirements and use cases for the Security Assertions Markup Language (SAML) developed the Oasis Security Services Technical Committee.

## Introduction

The issues list presented here documents issues brought up in response to Use Case and Requirements drafts as well as other issues mentioned on the security-use and security mailing lists, in conference calls, and in other venues. Each issue is formatted according to the proposal of David Orchard to the general committee:

ISSUE:[Document/Section Abbreviation-Issue Number: Short name]
Issue long description.
Possible resolutions, with optional editor resolution
Decision

The issues are informally grouped according to general areas of concern. For this document, the "Issue Number" is given as "#-##", where the first number is the number of the issue group.

The issues are in varying levels of resolution. Some are stated as questions or placeholders for further investigation. Others are stated as problems with resolutions, and still others have full-blown use case scenarios attached.

# Issues

## Group 0: Document Format & Strategy

ISSUE:[UC-0-01:MergeUseCases] There are several use case scenarios in the Straw Man 1 that overlap in purpose. For example, there are several single sign-on scenarios. Should these be merged into a single use case, or should the multiplicity of scenarios be preserved? Possible Resolutions:

1. Merge similar use case scenarios into a few high-level use cases, illustrated with UML use case diagrams. Preserve the detailed use case scenarios, illustrated with UML interaction diagrams. This allows casual readers to grasp quickly the scope of SAML, while keeping details of expected use of SAML in the document for other subcommittees to use.

2. Merge similar use case scenarios, leave out detailed scenarios.

Status: Open ISSUE:[UC-0-02:Terminology] Several subcommittee members have found the current document, and particularly the use case scenario diagrams, confusing in that they use either domain-specific terminology (e.g., "Web User", "Buyer") or vague, undefined terms (e.g., "Security Service.").

One proposal is to replace all such terms with a standard actor naming scheme, suggested by Hal Lockhart and adapted by Bob Morgan, as follows:

1. User

2. Authn Authority

3. Authz Authority

4. Policy Decision Point (PDP)

5. Policy Enforcement Point (PEP)

A counter-argument is that abstraction at this level is the point of design and not of requirements analysis. In particular, the real-world naming of actors in use cases makes for a more concrete goal for other subcommittees to measure against. Another proposal is, for each use case scenario, to add a section that maps the players in the scenario to one or more of the actors called out above.

Possible Resolutions:

1. Replace domain-specific or vague terms with standard vocabulary above.

2. Map domain-specific or vague terms to standard vocabulary above for each use-case and

scenario.

3. Don't make global changes based on this issue.

Status: Open ISSUE:[UC-0-02:Arrows] Another problem brought up is that the use case scenarios have messages (arrow) between actors, but not much detail about the actual payload of the arrows. Although this document is intended for a high level of analysis, it has been suggested that more definite data flow in the interaction diagrams would make them clearer.

UC-1-08:AuthZAttrs, UC-1-09:AuthZDecisions, and UC-1-11:AuthNEvents all address this question to some degree, but this issue is added to state for a general editorial principle for the document.

Possible Resolutions:

1. Edit interaction diagrams to give more fine-grained detail and exact payloads of each message between players.

2. Don't make global changes based on this issue.

Status: Open

## Group 1: Single Sign-on Push and Pull Variations

ISSUE:[UC-1-01:Shibboleth] The Shibboleth security system for Internet 2 (http://middleware.internet2.edu/shibboleth/index.shtml) is closely related to the SAML effort. An attempt has been made to address the requirements and design of Shibboleth in the SAML requirements document to allow implementation of SAML to be part of, or at least interoperable with, Shibboleth implementations. In particular, the following issues have been introduced to address Shibboleth requirements:

- UC-1-04:ARundgrenPush
- UC-1-06:Anonymity
- UC-1-07:Pseudonymity
- UC-1-10:UntrustedPartners
- UC-4-04:SecurityDiscovery
- UC-9-03:PrivacyStatement
- UC-9-04:RuntimePrivacy

If these issues, along with the straw man 2 document, have addressed the requirements of Shibboleth, then the subcommittee can address each issue on its own, rather than Shibboleth as a

monolithic problem. Possible Resolutions:

1. The above list of issues, combined with the straw man 2 document, address the requirements of Shibboleth, and no further investigation of Shibboleth is necessary.

2. Additional investigation of Shibboleth requirements are needed.

Status: Voted, Resolution 1 Carries Voting Results

| | |
|---|---|
| Date | 23 Feb 2001 |
| Eligible | 18 |
| Resolution 1 | 6 |
| Resolution 2 | 0 |
| Abstain | 3 |

ISSUE:[UC-1-02:ThirdParty] Use case scenario 3 (single sign-on, third party) describes a scenario in which a Web user logs in to a particular 3rd-party security provider which returns an authentication reference that can be used to access multiple destination Web sites. Is this different than Use case scenario 1 (single sign-on, pull model)? If not, should it be removed from the use case and requirements document?

As written, the use case is not truly different from use case scenario 1. However, if the use case scenario is expanded to include multiple destination sites, the importance of this use case becomes more apparent.

The following edition to the single sign-on, third party use case scenario would be added:

In this single sign-on scenario, a third-party security service provides authentication assertions for the user. Multiple destination sites can use the same authentication assertions to authenticate the Web user. Note that the first interaction, between the security service and the first destination site, uses the pull model as described above. The second interaction uses the push model. Either of the interactions could use a different single sign-on model.

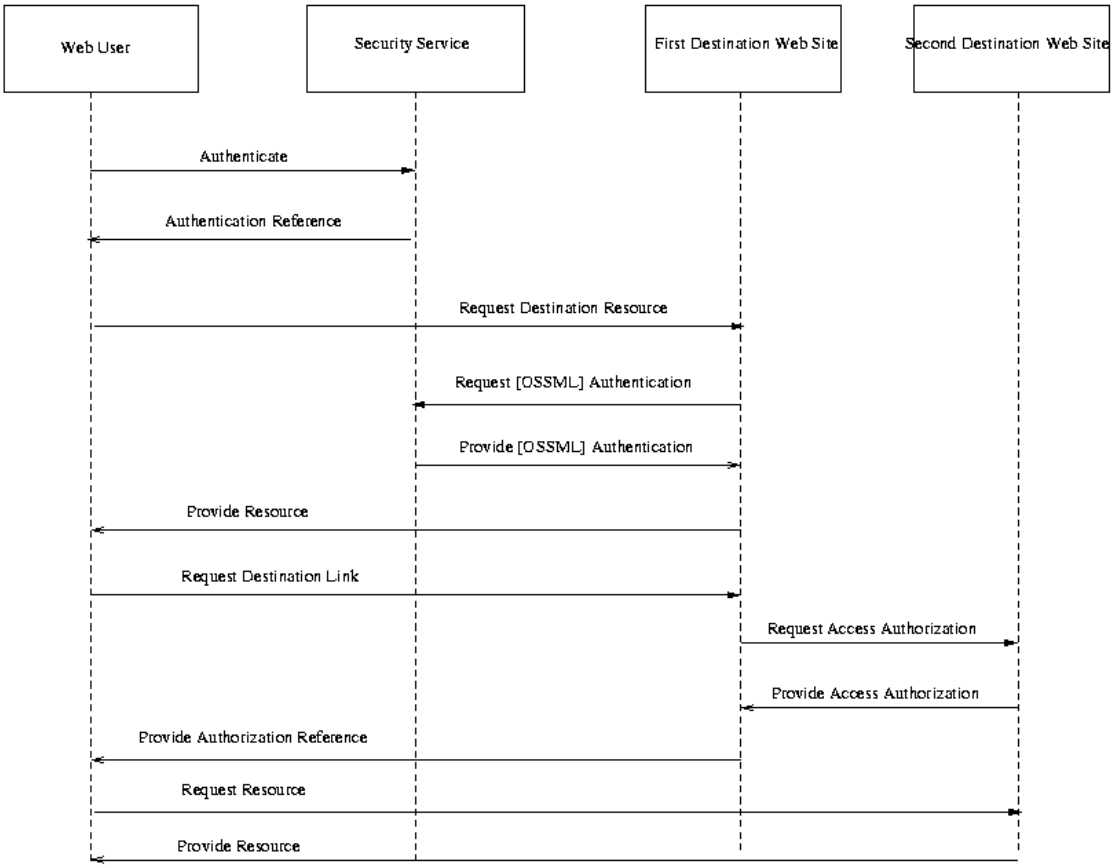



Fig. X. Single Sign-on, Third-Party Security Service

Steps:

1. Web user authenticates with security service.
2. Security service returns SAML authentication reference to Web user.
3. Web user requests resource from first destination Web site, providing authentication reference.
4. First destination Web site requests authentication document from security service, passing the Web user's authentication reference.
5. Security service provides authentication document to first destination Web site.
6. First destination Web site provides resource to Web user.
7. Web user requests link to second destination Web site from first destination Web site.
8. First destination Web site requests access authorization from second destination Web site,

providing third-party security service authentication document for user.

9. Second destination Web site provides access authorization. 10. First destination Web site provides authorization reference to Web user.

10. Web user requests resource from second destination Web site, providing authorization reference.

11. Second destination Web site provides resource.

Possible Resolutions:

1. Edit the current third-party use case scenario to feature passing a third-party authentication assertion from one destination site to another.

2. Remove the third-party use case scenario entirely.

Status: Voted, Resolution 1 Carries Voting Results

| | |
|---|---|
| Date | 23 Feb 2001 |
| Eligible | 18 |
| Resolution 1 | 7 |
| Resolution 2 | 2 |
| Abstain | 0 |

ISSUE:[UC-1-03:ThirdPartyDoable] Questions have arisen whether use case scenario 3 is doable with current Web browser technology. An alternative is using a Microsoft Passport-like architecture or scenario.

It seems that at least one possible solution for the third-party security system exists -- that each destination site pass the authentication assertion from the third party security service to the next destination site, just as in peer source and destination scenarios such as use case scenarios 1 and 2.

Therefore, it seems that the scenario is at least theoretically implementable. It will be up to the other subcommittees and implementors of the standard to decide on how to define that implementation.

Possible Resolutions:

1. The use case scenario should be removed because it is unimplementable.

2. The use case scenario is implementable, and whether it should stay in the document or not should be decided based on other factors.

Status: Voted, Resolution 2 Carries Voting Results

| | |
|---|---|
| Date | 23 Feb 2001 |
| Eligible | 18 |
| Resolution 1 | 2 |
| Resolution 2 | 8 |
| Abstain | 0 |

Bob Blakley noted, "I think the proposed implementation only works if you follow direct links, and not if you pick destinations from a history list, use bookmarks, etc..."

ISSUE:[UC-1-04:ARundgrenPush] Anders Rundgren has proposed on security-use an alternative to use case scenario 2 (single sign-on, push model). The particular variation is that the source Web site requests an authorization profile for a resource (e.g., the credentials necessary to access the resource) before requesting access.

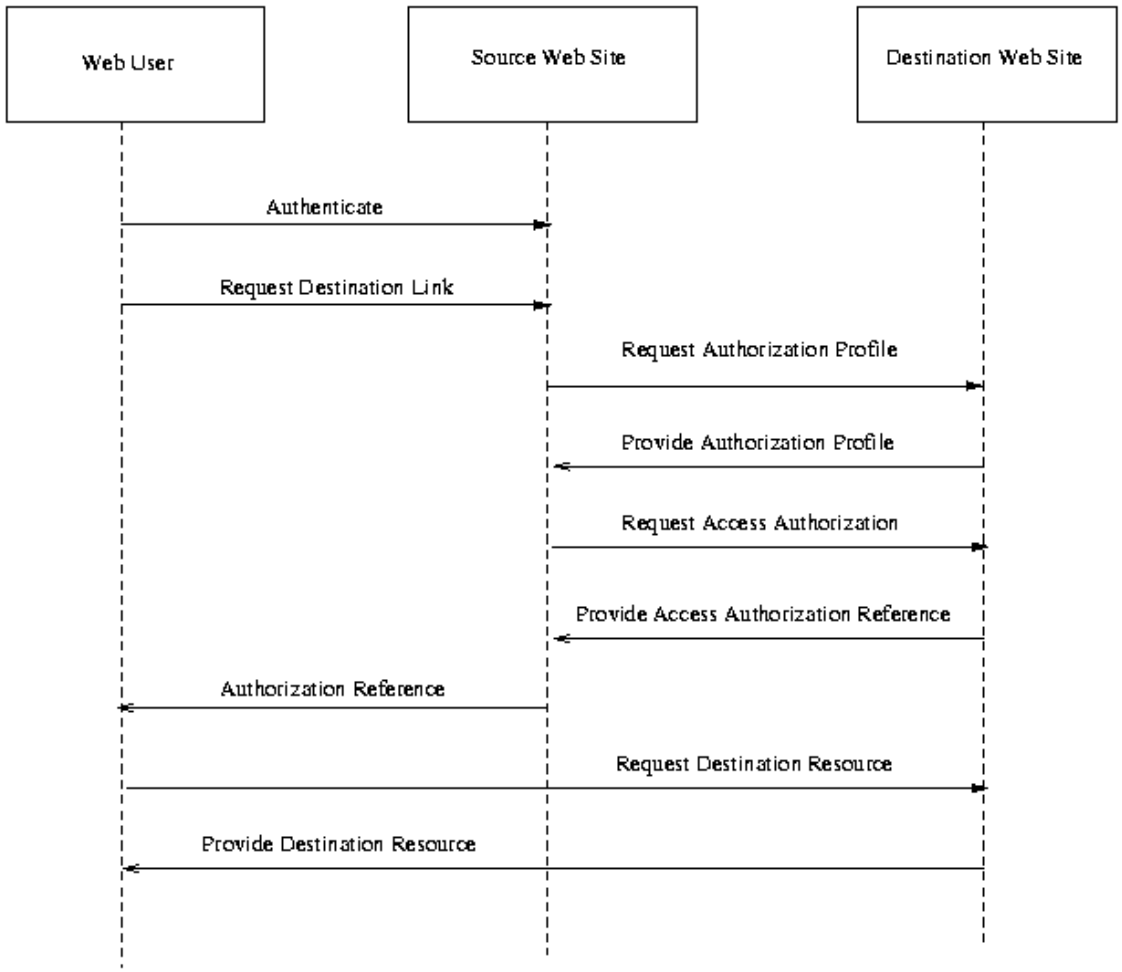



Fig X. Single Sign-on, Alternative Push Model.

Possible Resolutions:

1. Use this variation to replace scenario 2 in the use case document.
2. Add this variation as an additional scenario in the use case document.
3. Do not add this use case scenario to the use case document.

Status: Voted, No Conclusion Voting Results

| | |
|---|---|
| Date | 23 Feb 2001 |

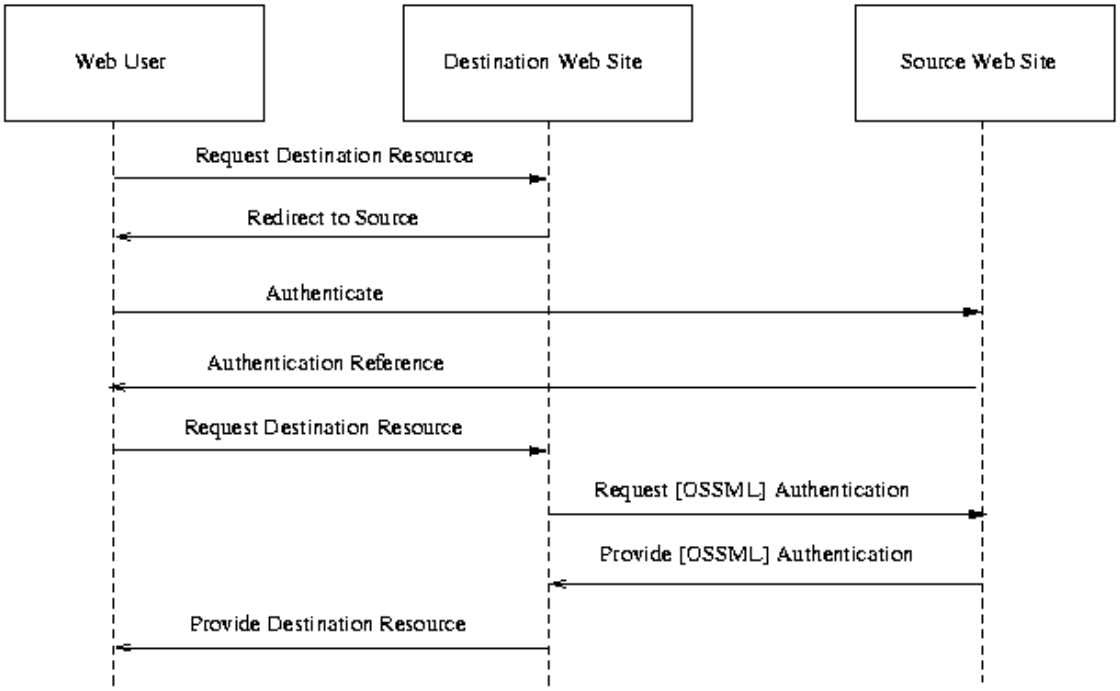
| | |
|---|---|
| Eligible | 18 |
| Resolution 1 | 0 |
| Resolution 2 | 3 |
| Resolution 3 | 6 |
| Abstain | 0 |

Bob Blakley noted, "I can't really see how to do this without significant changes to the current link resolution architecture of web sites -- specifically without making sure both source and destination are expecting to have to handle this flow."

ISSUE:[UC-1-05:FirstContact] A variation on the single sign on use case that has been proposed is one where the Web user goes directly to the destination Web site without authenticating with a definitive authority first.

A single sign-on use case scenario would be added as follows:

In this single sign-on scenario, the user does not first authenticate with their home security domain. Instead, they go directly to the destination Web site, first. The destination site must then redirect the user to a site they can authenticate at. The situation then continues as if in a single sign-on, push model scenario.




Single Sign-on, Alternative Push Model

Steps:

1. Web user requests resource from destination Web site.
2. Destination Web site determines that the Web user is unauthenticated. It chooses the appropriate home domain for that user (deployment dependent), and redirects the Web user to that source Web site.
3. Web user authenticates with source Web site.
4. Source Web site provides user with authentication reference (AKA "name assertion reference"), and redirects user to destination Web site.
5. Web user requests destination Web site resource, providing authentication reference.
6. Destination Web site requests authentication document ("name assertion") from source Web site, passing authentication reference.
7. Source Web site returns authentication document.
8. Destination Web site provides resource to Web user.

Possible Resolutions:

1. Add this use case scenario to the use case document.
2. Do not add this use case scenario to the use case document.

Status: Voted, No Conclusion Voting Results

| | |
|---|---|
| Date | 23 Feb 2001 |
| Eligible | 18 |
| Resolution 1 | 6 |
| Resolution 2 | 3 |
| Abstain | 0 |

Bob Blakley said, " I agree that servers will have to do this, but it can easily be done by writing HTML with no requirement for us to provide anything in our specification."

ISSUE:[UC-1-06:Anonymity] What part does anonymity play in SAML conversations? Can assertions be for anonymous parties? Here, "anonymous" means that an assertion about a principal does not include an attribute uniquely identifying the principal (ex: user name, distinguished name, etc.).

A requirement for anonymity would state:

> [CR-1-06-Anonymity] SAML will allow assertions to be made about anonymous principals, where "anonymous" means that an assertion about a principal does not include an attribute uniquely identifying the principal (ex: user name, distinguished name, etc.).

Possible Resolutions:

1. Add this requirement to the use case and requirement document.
2. Do not add this requirement.

Status: Voted, Resolution 1 Carries Voting Results

| | |
|---|---|
| Date | 23 Feb 2001 |
| Eligible | 18 |
| Resolution 1 | 9 |
| Resolution 2 | 0 |
| Abstain | 0 |

ISSUE:[UC-1-07:Pseudonymity] What part do pseudonyms play in SAML conversations? Can assertions be made about principals using pseudonyms? Here, a pseudonym is an attribute in an assertion that identifies the principal, but is not the identifier used in the principal's home domain.

A requirement for pseudonymity would state:

> [CR-1-07-Pseudonymity] SAML will allow assertions to be made about principals using pseudonyms for identifiers.

Possible Resolutions:

1. Add this requirement to the use case and requirement document.
2. Do not add this requirement.

Status: Voted, Resolution 1 Carries Voting Results

| | |
|---|---|
| Date | 23 Feb 2001 |
| Eligible | 18 |
| Resolution 1 | 7 |
| Resolution 2 | 2 |
| Abstain | 0 |

In support of Resolution 1, while voting, Bob Blakley said, "I'm really ambivalent about this. At an implementation level AND at a specification level, I can't see how a pseudonym should differ from a 'real' name. If it shouldn't, then we have no work to do. However, we should at least

discuss the issue."

ISSUE:[UC-1-08:AuthZAttrs] It's been pointed out that the concept of an "authentication document" used in the use case and requirements document does not clearly specify the inclusion of authz attributes. Here, authz attributes are attributes of a principal that are used to make authz decisions, e.g. an identifier, or group or role membership.

Since authz attributes are important and are required by [R-AuthZ], it has been suggested that the single sign-on use case scenarios specify when authz assertions are passed between actors.

Possible Resolutions:

1. Edit the use case scenarios to specify passing authz attributes with authentication documents.
2. Do not specify the passing of authz attributes in the use case scenarios.

Status: Voted, Resolution 1 Carries Voting Results

| | |
|---|---|
| Date | 23 Feb 2001 |
| Eligible | 18 |
| Resolution 1 | 9 |
| Resolution 2 | 0 |
| Abstain | 0 |

ISSUE:[UC-1-09:AuthZDecisions] The current use case and requirements document mentions "Access Authorization" and "Access Authorization References." In particular, this data is a record of a authorization decision made about a particular principal performing a particular action on a particular resource.

It would be more clear to label this data as "AuthZ Decision Documents" to differentiate from other AuthZ data, such as AuthZ attributes or AuthZ policy. To this point, the mentions of "access authorization" would be changed, and a new requirement would be added as follows:

[CR-1-09-AuthZDecision] SAML should define a data format for recording authorization decisions.

Possible Resolutions:

1. Edit the use case scenarios to use the term "authz decision" and add the [CR-1-09-AuthZDecision] requirement.
2. Do not make these changes.

Status: Voted, Resolution 1 Carries Voting Results

| | |
|---|---|
| Date | 23 Feb 2001 |
| Eligible | 18 |
| Resolution 1 | 8 |
| Resolution 2 | 0 |
| Abstain | 1 |

ISSUE:[UC-1-10:UnknownParty] The current straw man 2 document does not have a use case scenario for exchanging data between security services that are previously unknown to each other. For example, a relying party may choose to trust assertions made by an asserting party based on the signatures on the AP's digital certificate, or through other means.

The following use case scenario would illustrate using assertions from an unknown party.

In this scenario, an application service provider has a policy to allow access to resources for all full-time students at accredited 4-year universities and colleges. It would be difficult for the application service provider to maintain agreements with hundreds of such organizations in order to verify assertions made by those parties. Instead, it chooses to check the key of the asserting party to ensure that the asserting party is a 4-year university.

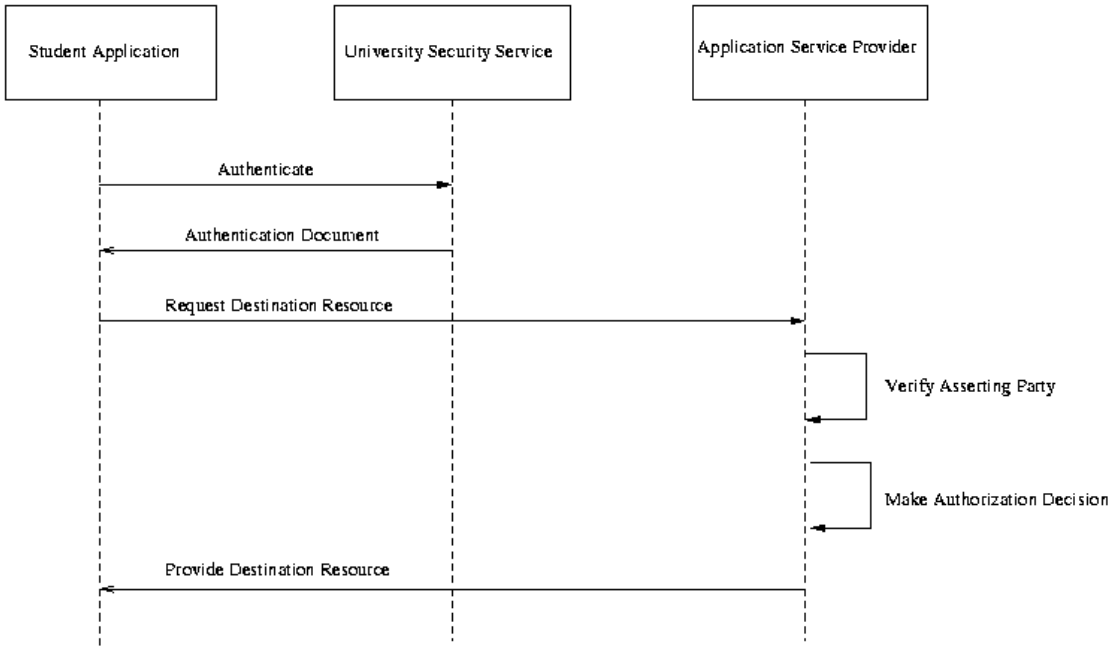



Fig X. Unknown Partner

Steps:

1. Student authenticates to university security system.
2. University provides authentication document to student application, including authentication event data and authorization attributes.
3. Student application requests resource from application service provider. Request includes authentication document.
4. Application service provider makes a trust decision about the authn and authz data, based on the key used to sign the assertion. It determines that the signing party is an accredited 4-year university, based on a signature on the key made by an accrediting organization.
5. Application service provider makes an authorization decision based on the authz attributes of the student.
6. Application service provider returns resource to the student.

Possible Resolutions:

1. Add this use case scenario to the use case document.
2. Do not add this use case scenario to the use case document.

Status: Voted, Resolution 2 Carries Voting Results

| | |
|---|---|
| Date | 23 Feb 2001 |
| Eligible | 18 |
| Resolution 1 | 2 |
| Resolution 2 | 7 |
| Abstain | 0 |

In voting for resolution 2, Bob Blakley said, " I think this overspecifies behavior... both the 'interesting' flows in the diagram here are from the Application Service Provider to *itself*. Why should we tell the A.S.P. how to make trust decisions about assertions?"

ISSUE:[UC-1-11:AuthNEvents] It is not specified in straw man 2 what authentication information is passed between parties. In particular, specific information about authn events, such as time of authn and authn protocol are alluded to but not specifically called out.

The use case scenarios would be edited to show when information about authn events would be transferred, and the requirement for authn data would be edited to say:

> [CR-1-11-AuthN] SAML should define a data format for authentication assertions, including descriptions of authentication events.

Possible Resolutions:

1. Edit the use case scenarios to specifically define when authn event descriptions are transferred, and edit the R-AuthN requirement.
2. Do not change the use case scenarios or R-AuthN requirement.

Status: Voted, Resolution 1 Carries Voting Results

| | |
|---|---|
| Date | 23 Feb 2001 |
| Eligible | 18 |

| | |
|---|---|
| Resolution 1 | 9 |
| Resolution 2 | 0 |
| Abstain | 0 |

ISSUE:[UC-1-12:SignOnService] Bob Morgan suggests changing the title of use case 1, "Single Sign-on," to "Sign-on Service."

Possible Resolutions:

1. Make this change to the document.
2. Don't make this change.

Status: Open ISSUE:[UC-1-13:ProxyModel] Irving Reid suggests an additional use case scenario for single sign-on, based on proxies.

A scenario would be added to the document as follows:

Scenario X: Single Sign-on, Proxy Model

In this model, the user authenticates to a proxy and then sends a request, including credentials, to the proxy. The proxy generates OSSML assertions, attaches them to the request, and forwards the request to the destination web site. The destination web site replies to the proxy, and the proxy forwards the reply back to the client.

In this model, the user authenticates to a proxy and then sends a request, including credentials, to the proxy. The proxy generates OSSML assertions, attaches them to the request, and forwards the request to the destination web site. The destination web site replies to the proxy, and the proxy forwards the reply back to the client.

Alternatively, the initial message from the client to the proxy could include both the authentication credentials and the request rather than having a separate round-trip for authentication.

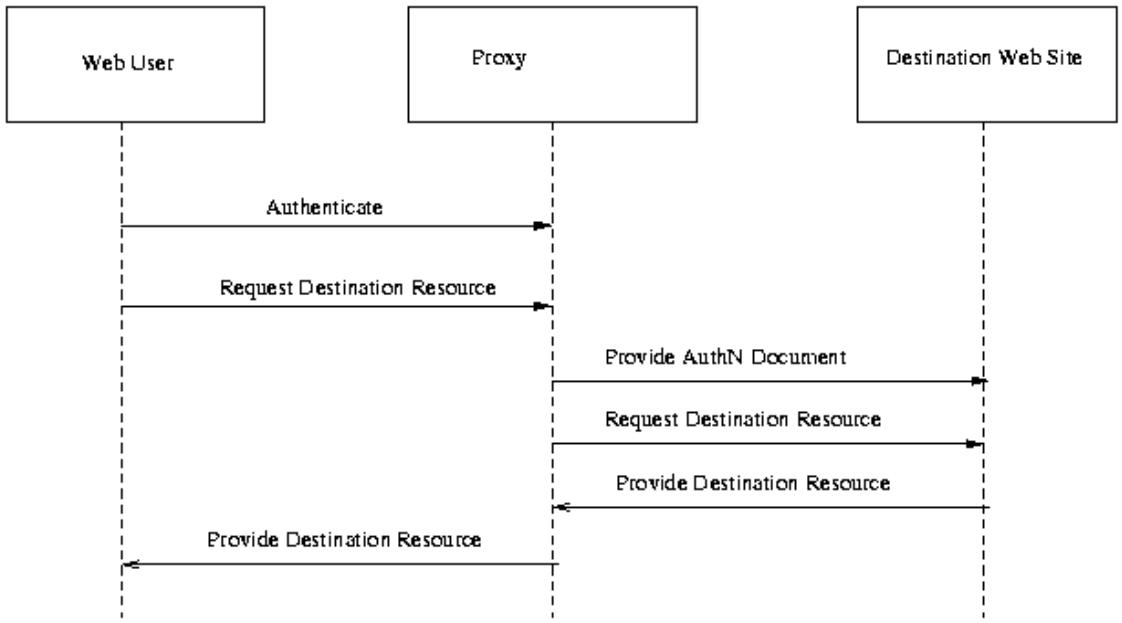



Fig X. Single Sign-on, Proxy Model

Steps:

1. Web user authenticates to proxy.
2. Web user requests destination resource through proxy.
3. Proxy provides authentication document to destination Web site.
4. Proxy requests destination resource from destination Web site.
5. Destination Web site provides destination resource to proxy.
6. Proxy provides destination resource to Web user.

There are two sub-variants to this use case: In some cases the proxy will return OSSML tokens of some sort to the client, and the client will use those tokens (most likely in the form of HTTP cookies) to make subsequent requests within the single-sign-on session. In the other variant, the proxy has an existing session mechanism with the client. In that case, the proxy can store the OSSML tokens and transparently attach them to subsequent requests within that session.
Possible Resolutions:

1. Add this use case scenario to the document.
2. Don't make this change.

Status: Open

# Group 2: B2B Scenario Variations

ISSUE:[UC-2-01:AddPolicyAssertions] Some use cases proposed on the security-use list (but not in the straw man 1 document) use a concept of a "policy document." In concept a policy document is a statement of policy about a particular resource, such as that user "evanp" is granted "execute" privileges on file "/usr/bin/emacs." Another example may be that all users in domain "Acme.com" with role "backup administrator" may perform the "shutdown" method on resource "mail server," during non-business hours. Use cases where policy documents are exchanged, and especially activities like security discovery as in UC-4-04:SecurityDiscovery, would require this type of assertion. If these use cases and/or services were adapted, the term "policy document" should be used. In addition, the following requirement would be added:

[CR-2-01-Policy] SAML should define a data format for security policy about resources.

In addition, the explicit non-goal for authorization policy would be removed. Possible Resolutions:

1. Remove the non-goal, add this requirement, and refer to data in this format as "policy documents."

2. Maintain the non-goal, leave out the requirement.

Status: Open ISSUE:[UC-2-02:OutsourcedManagement] A use case scenario provided by Hewlett Packard illustrates using SAML enveloped in a CIM/XML request. Should this scenario be included in the use case document?

The use case would be inserted as follows (some editing for clarity):

This scenario shows an enterprise A that has outsourced the management of its network devices to a management service provider B. Management messages are exchanged using CIM/XML over HTTP. (CIM or Common Information Model, is a management standard being developed by the Distributed Management Task Force - http://www.dmtf.org/, an XML DTD for CIM has been defined.)

Suppose the operator, Joe, wants to invoke the StopService method. This will be executed by the XML/CIM agent on the managed device, if authorized.

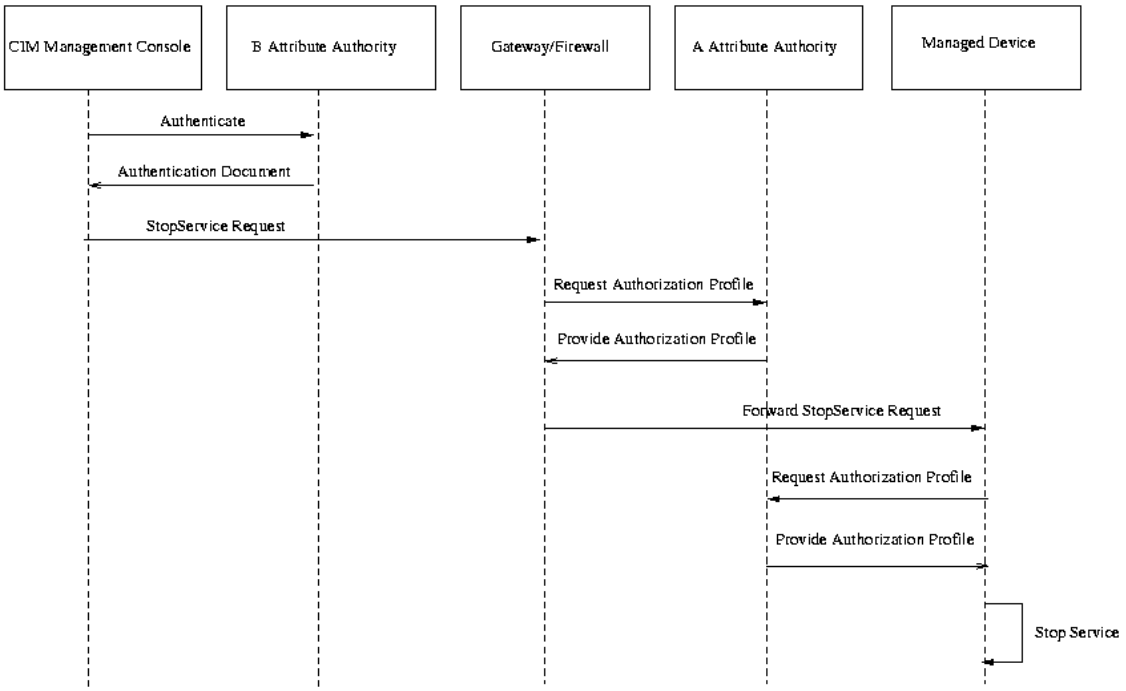



Fig X. Outsourced Management.

Steps:

1. This SAML assertion has been generated by B's attribute authority (or Policy Decision Point) and confers the role "System Manager for A" to Joe.

2. The CIM management console generates the XML content and attaches an SAML assertion. The CIM management console signs the request and sends it as an HTTP request.

3. The request now has to traverse A's firewall or the boundary into A's network. The gateway at this boundary uses its SAML evaluation engine (or Policy Enforcement Point) to verify that this incoming message is allowed. It does this, by verifying the signature and discovering the request is from Joe. Next it uses two assertions to authorize the incoming message: the assertion issued by B's attribute authority that is attached to the message (conferring the role "System Manager for A" on Joe); an assertion issued by A's attribute authority granting "Gateway Access" to any entity that has a valid "System Manager for A" assertion issued by B's attribute authority. Note that the second assertion can be pushed to the gateway (part of its configuration), or retrieved dynamically from a repository (or indeed the issuer) (the last case is shown here).

4. The request is forwarded by the gateway to the managed device.

5. The SAML evaluation engine on the managed device needs to determine if a

"StopService" request from Joe is allowed. It does this by using two assertions: the "System Manager for A" assertion issued by B's attribute authority; an assertion issued by A's attribute authority granting "Full Management Rights" to any entity with a valid "System Manager for A" assertion issued by B's attribute authority.

6. The managed device executes the "StopService" method.

Status: Open ISSUE:[UC-2-03:ASP] A use case scenario provided by Hewlett Packard illustrates using SAML for a secure interaction between an application service provider (ASP) and a client. Should this scenario be included in the use case document?

The use case would be inserted as follows (some editing for clarity):

In this scenario an ASP, A, is providing an application (possible examples could be a word processor or an ERP application) to users in another enterprise, B. A VPN (for example IPSEC) is used to provide a secure end-to-end tunnel between the client and server.

A major difference between this scenario and the outsource management service scenario is that all assertions are "pulled" in this scenario. This means the assertions are not attached to application messages; instead they must be retrieved either directly from the attribute authority, or a repository. For example, once the client has been authenticated, the SAML evaluation engine in the server needs to retrieve the SAML assertions issued by A and B. This will involve making a request to a repository inside B, traversing both A and B's firewall as shown in the diagram. Similarly the SAML engines in the gateway and client will have to retrieve assertions issued by both authorities.

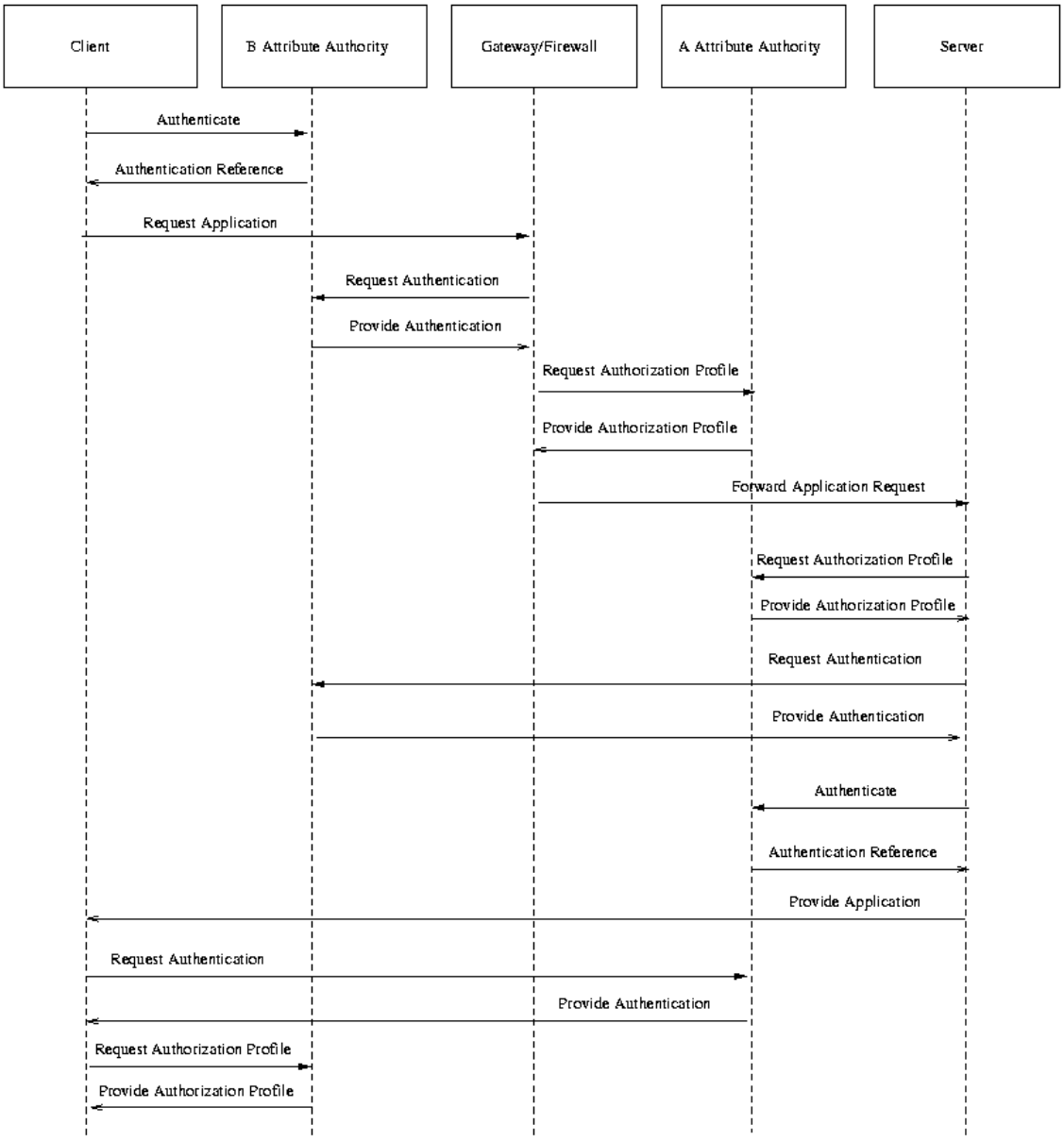



Fig X. Application Service Provider.

Steps:

1. The client authenticates with B's attribute authority.
2. B's attribute authority provides an authentication assertion that the client is a "valid user."
3. The client requests an application through A's gateway, providing a reference to the authentication assertion.

4. The gateway needs to know that incoming packets from a client in B are allowed. It needs an assertion from B's attribute authority that the client is a valid user, and an assertion from A's attribute authority that entities issued "valid user" assertions from B are allowed access. The gateway requests the assertion from B's attribute authority.
5. B's attribute authority provides the assertion.
6. The gateway requests an authorization assertion from A's attribute authority.
7. A's attribute authority provides the authorization assertion.
8. The gateway forwards the request to the Server.
9. The server requests the assertion from B's attribute authority.
10. B's attribute authority provides the assertion.
11. The server requests an authorization assertion from A's attribute authority.
12. A's attribute authority provides the authorization assertion.
13. The server authenticates with A's attribute authority.
14. A's attribute authority provides a reference to an authentication assertion that the server is an "Approved Application".
15. The server returns the application to the client.
16. It is also important that the client check that the application is valid. This avoids problems such as an attacker spoofing the service provider and providing a word processor service that silently emails copies of all documents generated by the client to the attacker. This might be done by the client SAML evaluation engine checking two assertions: one from A granting "Approved Application" status to the server; one from B granting the attribute "execute" to any entity with "Approved Application" status issued by A. The Client requests the authentication assertion from A's attribute authority.
17. A's attribute authority provides the assertion.
18. The client requests an authorization assertion from B's attribute authority.
19. B's attribute authority provides the authorization assertion.

Status: Open ISSUE:[UC-2-04:HealthCare] A request for a use case focussing on health care and particularly HIPPA has been made in the Security Services TC. Should such a use case scenario be added to the use case document? What are the particulars of HIPPA and how are they different from other use cases?

Status: Open

ISSUE:[UC-2-05:B2B Transaction via an e-marketplace or trading hub] Zahid Ahmed proposes the following additional use case scenario for inclusion in the use case and requirements document.

A B2B Transaction involving buyers and suppliers that conduct trade via an e-marketplace that provides trading party authentication and authorization services, and other business services, in support of secure transaction and routing of business document exchanges between trading parties.

Steps:

1. A trading party (e.g., buyer) creates a business document for subsequent transaction with another trading party (e.g., supplier) accessible via its e-marketplace.

2. The sending, i.e., transaction-initiating trading party (TP) application creates a SAML Credential to be authenticated by the authentication and security service operated by an e-marketplace.

3. The trading party application transaction client packages the XML-based SAML Credential alongwith the other XML-based business document over a specific transport, messaging, and application protocol.

   Some examples of such (layered) protocols are following (but not limited to):

   - Secure transports: SSL and/or HTTPS
   - Messaging protocol: S/MIME and JMS.
   - Message Enveloping Formats: SOAP, etc.
   - B2B Application Protocol: ebXML, BizTalk, etc.

4. E-marketplace Authentication Service validates the TP Credential and creates a SAML Named Assertion and any Entitlements for the transaction-initiating TP.

5. The E-marketplace Messaging Service then packages the Named Assertion and Entitlements along with the original message payload into a tamper-proof envelope (i.e., S/MIME multi-part signed)

6. The resulting message envelope is transmitted to the target trading party (service provider).

7. The receiving trading party application extracts and processes the TP identity (i.e., Named Assertion) and authorization (i.e., Entitlement) information available in the received envelope.

8. Receiving TP application then processes the business document of the sending TP.

9. Receiving TP sends back a response to sending TP via its e-marketplace by repeating Steps 1 through 6.

Possible Resolutions:

1. The above scenario should be part of OASIS Use Cases/Requirements.

2. The above scenario should not be added to the document.

Status: Open ISSUE:[UC-2-06:B2B Transaction using different messaging and application protocols] Zahid Ahmed has proposed that the following use case scenario be added to the use case and requirements document.

A B2B Document Exchange Transaction that involves two trading parties such that sending trading party (e.g., Buyer) uses one messaging and transport protocol (e.g., OBI) and receiving party (e.g., Supplier) uses a another messaging/transport protocol (e.g., ebXML). A B2B transaction service must provide relevant security interoperability services as part of its general messaging and application interoperability mechanism.

Steps:

1. The sending trading party employs a specific messaging and application protocol.

2. The sending TP application then transacts with the receiving TP via its e-marketplace following Steps# 1 through 3 in Issue# UC-2-05 described above.

3. The e-marketplace authentication and security service provider authenticated and validates the sending TP and produce relevant SAML security assertions as described in Step# 4in Issue# UC-2-05 described above.

4. The e-marketplace interoperability service transforms the incoming message to target trading party messaging and application protocol such that SAML Named Assertions and any SAML Entitlement document instances are included into the newly transformed message for subsequent transmission to the receiving TP.

5. The receiving TP extracts, processes the security assertions about the sending TP as described in Step# 7 in Issue# UC-2-05 above.

6. Receiving TP sends back a response to sending TP via its e-marketplace by repeating Steps 1 through 5.

Possible Resolutions:

1. The above use case scenario should be added to the of OASIS Use Cases/Requirements document.

2. This use case scenario should not be added to the document.

Status: Open ISSUE:[UC-2-07:B2B Transaction over multiple e-marketplace or trading hubs/portals] Zahid Ahmed proposes the following use case scenario for inclusion in the document. This use case/issue is a variant of ISSUE# [UC-2-05].

In this scenario the transacting trading parties are members of different e-marketplaces or trading communities. To support B2B transactions between trading parties of different e-markletplaces, the e-marketplaces will provide secure interconnectivity between the set of trading hubs involved in the transaction between the transaction parties. In this manner e-marketplaces will act as trusted intermediaries between transacting trading parties.

Steps:

1. Repeat Steps# 1-5 in Issue# [UC-2-07].

2. Receiving e-marketplace, e.g., e-marketplace A, message service transmits the message to target e-marketplace, e-marketplace B.

3. E-marketplace B Authentication Service validates the Signed Envelope that contains the E-marketplace signature used to package the SAML security assertions about the sending TP.

4. E-marketplace B Authentication Service may additionally validate And/or insert new SAML Named Assertion and Entitlements, depending on its inter-portal connectivity security policies.

5. E-marketplace B transmits the authenticated message received from E-marketplace A to either another e-marketplace, e-marketplace C (repeat of Steps 1 through 4), or to the target TP.

Possible Resolutions:

1. Above scenario involving multiple trading hubs should be added to the document.

2. The above scenario should not be added to the document.

Status: Open ISSUE:[UC-2-08:ebXML] Maryann Hondo proposed this use case scenario for inclusion in the use case document. (Note that an interaction diagram illustrating this use case still must be developed, to replace the current diagram. Also, the steps involved should be brought in line with other use case scenarios in the use case and requirements document.)

Use Case Scenario X: ebXML

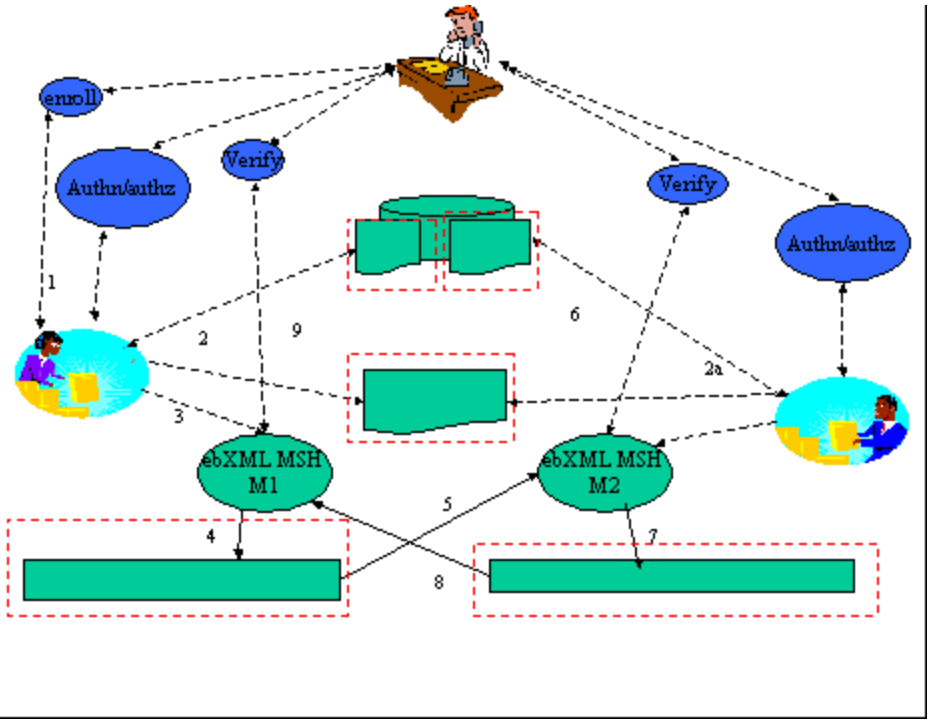



Fig X. ebXML

Steps:

1. Party A wishes to engage with Party B in a business transaction. To do this, Party A accesses information [stored in an ebXML Collaboration Party Profile (CPP)] about Party B's requirements for doing business. Party A and Party B negotiate at Collaboration Party Agreement (CPA). Some of the information in a CPP or CPA might include:
    - Party B requires authorization credentials from AuthServiceXYZ
    - Party B requires that Party A be authorized by XYZ in the BuyerQ role.
2. Party A then must be able to determine:
    - How to get these authorization credentials.
    - where/how to insert these credentials in an ebXML message
3. Party B's Message Service Handler (MSH) has received a digitally-signed ebXML message from Party A and wishes to obtain authorization information about Party A
    - Authorization data must be retrievable based on the DN in the certificate used to sign the ebXML message.
4. Party A has enrolled with AuthServiceXYZ. Party A engages in ebXML business

transactions and wants to restrict what entities are able to retrieve its authorization data.

Potential Resolutions:

1. Add this use case scenario to the use case and requirements document.
2. Do not add this scenario.

Status: Open

# Group 3: Sessions

ISSUE:[UC-3-1:UserSession] Should the use cases of log-off and timeout be supported? These result in the notion of session management. Advantage: Allows complete web user experience across multiple web sites. If not done as part of this specification, then some other body or work will have to standardize this functionality. Disadvantage: More complex than just passing authentication references between source and destination. Will slow down Technical committees work on specification of authentication/authorization only queries. Candidate Requirement:

[CR-3-1-UserSession] SAML shall support web user session(s).

The following use case scenario would be added to the use case and requirements document. A Single Sign-on and hand-off

Note that this is a duplicate of Oasis security Services Scenario #1

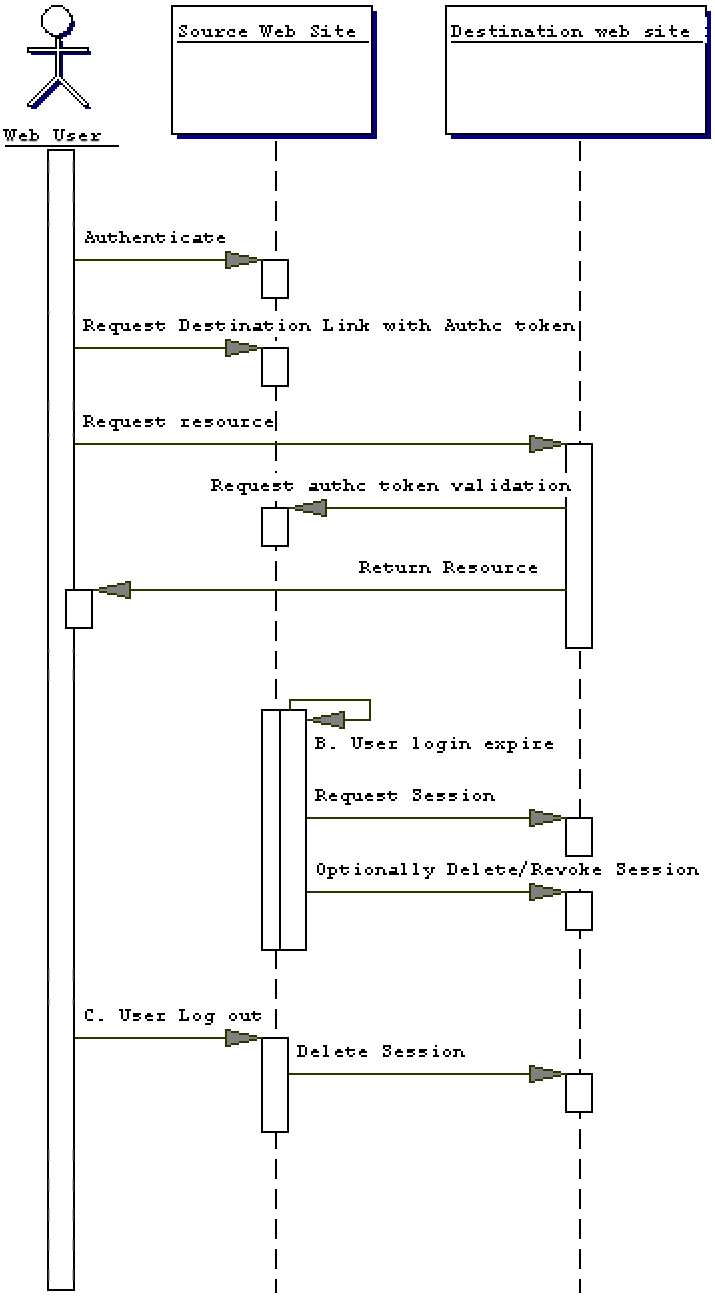



Fig X. Single Sign-on, User Session.

Steps:

1. A user logs onto the source Web site. This results in the creation of a session on the source web site.

2. User requests a link to a destination web site. This link contains an authentication reference/token/ticket.

3. User requests resource represented by link on destination web site, including reference
4. Destination web site requests validation of authentication reference from source web site.
5. Source web site returns success or failure, optionally additional session information.
6. Destination web site returns web site to user

Timeout

1. Assume that the user has gone beyond the timeout limit on the source web site.
2. The source web site will query each participating web site to determine if the user has been active on their web site.
3. If the user has not been active on any of the destination web sites within the timeout period, the destination web sites are instructed to delete the session.

Logout

1. User logs out of the source web site.
2. Each of the destination web sites are instructed to delete the session.

Possible Resolutions:

1. Add this requirement and/or use cases to SAML.
2. Do not add this requirement and/or use cases.

Status: Voted, Resolution 1 Carries Voting Results

| | |
|---|---|
| Date | 23 Feb 2001 |
| Eligible | 18 |
| Resolution 1 | 8 |
| Resolution 2 | 2 |
| Abstain | 0 |

In voting for resolution 1, Jeff Hodges added, "rationale: if there's these "assertions" floating

about between various entities that serve to assert the identity of some particular entity, there's notions of "validity time period" (however implemented), and there's notions of "state" relative to the asserted identity, then I feel what we have here meets the definition of a "session", and we ought to use that term (and really figure out what all the implications are)." He also attached the following URLs:

```
http://foldoc.doc.ic.ac.uk/foldoc/foldoc.cgi?query=session&action=Search
      http://foldoc.doc.ic.ac.uk/foldoc/foldoc.cgi?query=state
```

ISSUE:[UC-3-02:ConversationSession] Is the concept of a session between security authorities separate from the concept of a user session? If so, should use case scenarios or requirements supporting security system sessions be supported? [DavidO: I don't understand this issue, but I have left in for backwards compatibility]. [DarrenP: I think this issue arose out of a misunderstanding/miscommunication on the mailing list and has been resolved. This is more of a formality to vote this one to a closed status.] Possible Resolutions:

1. Do not pursue this requirement as it is not in scope.
2. Do further analysis on this requirement to determine what it is specifically.

Status: Voted, No Conclusion Voting Results

| | |
|---|---|
| Date | 23 Feb 2001 |
| Eligible | 18 |
| Resolution 1 | 5 |
| Resolution 2 | 5 |
| Abstain | 0 |

ISSUE:[UC-3-03:Logout] Should SAML support transfer of information about logout (e.g., a principal intentionally ending a session)? [DavidO: Isn't this covered in UC-3-1? I've kept here for backwards compatibility]

Candidate Requirement:

[CR-3-3-Logout] SAML shall support web user logout.

Possible Resolutions:

1. Add this requirement and/or use cases to SAML.
2. Do not add this requirement and/or use cases

Status: Voted, No Conclusion Voting Results

| | |
|---|---|
| Date | 23 Feb 2001 |
| Eligible | 18 |
| Resolution 1 | 5 |
| Resolution 2 | 5 |
| Abstain | 0 |

In voting for resolution 2, Jeff Hodges added, "rationale: I believe this is subsumed within the topic of [UC-3-1:UserSession] and we should deal with it explicitly in that context."

ISSUE:[UC-3-6:Destination Logout] Should logging out of a destination web site be supported? Advantage: allows web sites control over their local domain, current model implemented on the web. Disadvantage: potentially more interactions between source and destination web sites

Candidate Requirement:

[CR-3-6-Destination Logout] SAML shall support logout at destination web sites

Possible Resolutions:

1. Add this requirement and/or use cases to SAML
2. Do not add this requirement and/or use cases

Status: Voted, No Conclusion Voting Results

| | |
|---|---|
| Date | 23 Feb 2001 |
| Eligible | 18 |

| | |
|---|---|
| Resolution 1 | 4 |
| Resolution 2 | 5 |
| Abstain | 1 |

ISSUE:[UC-3-7:Logout Extent] What is the impact of logging out at a destination web site?

Possible Resolution:

1. Logout from destination web site is local to destination [DavidO recommendation]
2. Logout from destination web site is global, that is destination + source web sites.

Status: Voted, Resolution 1 Carries Voting Results

| | |
|---|---|
| Date | 23 Feb 2001 |
| Eligible | 18 |
| Resolution 1 | 7 |
| Resolution 2 | 0 |
| Resolution 3 | 1 |
| Abstain | 2 |

Jeff Hodges, abstaining, said, "rationale: needs clarification. E.g. BobB's point in Group3VoteBlakley.html should be considered."

ISSUE:[UC-3-04:StepUpAuthn] "Step-up" authentication is when a receiving party refuses to accept an authentication from an authenticating party and asks for a higher level of authentication. For example, the RP can refuse password authn and require certificate authn. Should SAML support step-up authentication? Should a use case be developed illustrating step-up authn?[DavidO: I don?t think this is applicable to the session requirements, but I?ve kept here for backwards compatibility].

Possible Resolutions:

1. Move this issue to the AuthN issue group and leave open for discussion and voting.

2. Step up Authentication is not a requirement. Close the issue.

Status: Voted, No Conclusion Voting Results

| | |
|---|---|
| Date | 23 Feb 2001 |
| Eligible | 18 |
| Resolution 1 | 5 |
| Resolution 2 | 4 |
| Abstain | 1 |

ISSUE:[UC-3-05:SessionTimeout] Should timeout be supported?

Candidate requirement:

[CR-3-5-Timeout] SAML shall support timeout of a user log-on.

Possible Resolutions:

1. Add this requirement and/or use cases to SAML.
2. Do not add this requirement and/or use cases.

Status: Voted, No Conclusion Voting Results

| | |
|---|---|
| Date | 23 Feb 2001 |
| Eligible | 18 |
| Resolution 1 | 6 |
| Resolution 2 | 4 |
| Abstain | 0 |

In voting for resolution 2, Jeff Hodges added, "rationale: I believe this is subsumed within the topic of [UC-3-1:UserSession] and we should deal with it explicitly in that context."

Bob Blakley said, "However I believe that the phrasing of the requirement is wrong. I think what we should support is expiration of assertions. Timeout is an action a receiving system implements based on observing that an assertion has timed out."

ISSUE:[UC-3-8:Destination Timeout] Should timing out of a session at a destination web site be supported?

Candidate requirement:

[CR-3-8-DestinationTimeout] SAML shall support destination web site timeout.

Possible Resolutions:

1. Add this requirement and/or use cases to SAML
2. Do not add this requirement and/or use cases

Status: Voted, No Conclusion Voting Results

| Date | 23 Feb 2001 |
|---|---|
| Eligible | 18 |
| Resolution 1 | 4 |
| Resolution 2 | 6 |
| Abstain | 0 |

In voting for resolution 2, Jeff Hodges added, "rationale: I believe this is subsumed within the topic of [UC-3-1:UserSession] and we should deal with it explicitly in that context."

Bob Blakley said, "I don't feel that I understand well enough what we'd consider doing here to express an opinion yet."

# Group 4: Security Services

ISSUE:[UC-4-01:SecurityService] Should part of the use case document be a definition of a security service? What is a security service and how is it defined? Status: Open

ISSUE:[UC-4-02:AttributeAuthority] Should a concept of an attribute authority be introduced into the SAML use case document? What part does it play? Should it be added in to an existing use case scenario, or be developed into its own scenario?

Status: Open

ISSUE:[UC-4-03:PrivateKeyHost] A concept taken from S2ML. A user may allow a server to host a private key. A credentials field identifies the server that holds the key. Should this concept be introduced into the SAML use case document? As a requirement? As part of an existing use case scenario, or as its own scenario?

Status: Open

ISSUE:[UC-4-04:SecurityDiscover] UC-1-04:ARundgrenPush describes a single sign-on scenario that would require transfer of authorization data about a resource between security zones. Should a service for security discovery be part of the SAML standard?

Possible Resolutions:

1. Yes, a service could be provided to send authorization data about a service between security zones. This would require some sort of AuthZ assertions (UC-2-01:AddAuthzAssertions).

2. No, this extends the scope of SAML too far. AuthZ in SAML should be concerned with AuthZ attributes of a principal, not of resources.

Status: Open

# Group 5: AuthN Protocols

ISSUE:[UC-5-03:AuthNThrough] All the scenarios in Straw Man 1 presume that the user provides authentication credentials (password, certificate, biometric, etc) to the authentication system out-of-band. Possible Resolutions (not mutually exclusive):

1. Should SAML be used directly for authentication? In other words should the SAML model or express one or more authentication methods or a framework for authentication?

2. Should this be explicitly stated as a non-goal?

3. Should the following statement would be added to the non-goals section?

   [NO-Authn] Authentication methods or frameworks are outside the scope of SAML.

Status: Voted, Resolution 1 Fails, Resolution 2 Passes, Resolution 3 No Conclusion Voting Results

| | |
|---|---|
| Date | 23 Feb 2001 |
| Eligible | 18 |
| Resolution 1 For | 1 |
| Resolution 1 Against | 10 |
| Resolution 2 For | 10 |
| Resolution 2 Against | 1 |
| Resolution 3 For | 7 |
| Resolution 3 Against | 4 |
| Abstain | 0 |

NOTE: resolutions for this issue were voted on separately.

ISSUE:[UC-5-02:SASL] Is there a need to develop materials within SAML that explore its relationship to SASL [SASL]?

Possible Resolutions:

1. Yes
2. No

Status: Voted, No Conclusion Voting Results

| | |
|---|---|
| Date | 23 Feb 2001 |
| Eligible | 18 |
| Resolution 1 | 3 |
| Resolution 2 | 5 |

| | |
|---|---|
| Abstain | 2 |

[ISSUE:[UC-5-01:AuthNProtocol] Straw Man 1 explicitly makes challenge-response authentication a non-goal. Is specifying which types of authn are allowed and what protocols they can use necessary for this document? If so, what types and which protocols?

As written, this issue covers a lot of ground. [UC-5-03:AuthNthrough] covers the core issue of the removal of all considerations of modeling authentication methods within SAML, which need not be discussed further in 5-01.

There is an aspect of these requirements that has been discussed and noted as important on the list. There is a need for describing different forms of credentials (name-password, public key, X509 certificates etc) within OSSML. In this sense there is a connection to the different "permitted forms of authn" [2] and OSSML.

REFERENCES: I believe these requirements are consistent with or can be derived from Nigel's suggestion [1] but is perhaps closer to the current style of specification in Strawman 2. It also reflects the discussion in [2] and [3].

```
      [1] http://lists.oasis-open.org/archives/security-
use/200102/msg00029.html
      [2] http://lists.oasis-open.org/archives/security-
use/200102/msg00038.html
      [3] http://lists.oasis-open.org/archives/security-
use/200102/msg00064.html
```

Possible Resolutions (not mutually exclusive):

1. The Non-Goal

   "Challenge-response authentication protocols are outside the scope of the SAML"

   should be removed from the Strawman 3 document.

2. The following requirements should be added to the Strawman 3 document:

   [CR-5-01-1-StandardCreds] SAML should provide a data format for credentials including those based on name-password, X509v3 certificates, public keys, X509 Distinguished name, and empty credentials.

   [CR-5-01-2-ExtensibleCreds] SAML The credentials data format must support extensibility in a structured fashion.

Status: Voted, No Conclusion Voting Results

| | |
|---|---|
| Date | 23 Feb 2001 |
| Eligible | 18 |
| Resolution 1 For | 8 |
| Resolution 1 Against | 3 |
| Resolution 2 For | 8 |
| Resolution 2 Against | 3 |
| Abstain | 0 |

In voting for resolution 2, Bob Blakley said, "My thinking here is that we need to provide a way to assert what mechanism was used to authenticate the user (e.g. certificate-based authentication) and what the user's authenticated credential resulting from that authentication (e.g. X.509 cert) was. I'm still nervous about allowing the VALUE of the password to be used as credential information as in S2ML, but I do understand why this was done and that it's useful."

# Group 6: Protocol Bindings

ISSUE:[UC-6-01:XMLProtocol] Should mention of a SOAP binding in the use case and requirements document be changed to a say "an XML protocol" (lower case, implying generic XML-based protocols)? Or "XML Protocol", the specific W3 RPC-like protocol using XML (http://www.w3.org/2000/xp/)? Status: Open

# Group 7: Enveloping vs. Enveloped

ISSUE:[UC-7-01:Enveloping] SAML data will be transferred with other types of XML data not specific to authn and authz, such as financial transaction data. What should the relationship of the documents be? Note that of the solutions below, 2. is more useful when the conversation is mostly an SAML conversation, such as for single sign-on. 3. is more useful for conversations that are mostly "other," such as XML-based server-to-server commerce.

Possible Resolutions:

1. SAML data and other data are sent as separate messages.
2. Enveloping the other data in an SAML message.
3. SAML data is enveloped in the other data message type.

4. Some combination of the above.

Status: Open

# Group 8: Intermediaries

ISSUE:[UC-8-01:Intermediaries] The use case scenarios in the S2ML 0.8a specification include one where an intermediary passes an S2ML message from a source party to a destination party. What is the part of intermediaries in an SAML conversation? Can intermediaries add, subtract, or alter data in an SAML message? Should a use case scenario involving a 3rd-party intermediary be included in the use case and requirements document? Status: Open

# Group 9: Privacy

ISSUE:[UC-9-02:PrivacyStatement] Important private data of end users should be shared as needed between peers in an SAML conversation and protected entirely from hostile 3rd parties. In addition, the user should have control over what data is exchanged. How should the requirement be expressed in the use case and requirements document? Should a use case scenario illustrating privacy protection be provided? One statement suggested by Bob Morgan is as follows:

> [CR-9-02-3-DisclosureMorgan] SAML should support policy-based disclosure of subject security attributes, based on the identities of parties involved in an authentication or authorization exchange.

> [CR-9-02-2-DisclosureBlakley] SAM should support *restriction of* disclosure of subject security attributes, *based on a policy stated by the subject*. *This policy might be* based on the identities of parties involved in an authentication or authorization exchange.

Status: Open ISSUE:[UC-9-01:RuntimePrivacy] Should protecting the privacy of the user be part of the SAML conversation? In other words, should user consent to exchange of data be given at run time, or at the time the user establishes a relationship with a security system?

Status: Open

# Group 10: Framework

ISSUE:[UC-10-01:Framework] Should SAML provide a framework that allows delivery of security content negotiated out-of-band. A typical use case is authorization extensions to the core SAML constructs. The contra-position is to rigidly define the constructs without allowing extension. Possible Resolutions:

1. Specify only the explicitly allowable content of messages, no framework

2. Allow full extensibility of message content (verbs and nouns) as well as flexible intermediary processing

3. Allow full extensibility of message content (verbs and nouns) with rigidly defined intermediary processing.

Status: Open

# Group 11: AuthZ Use Case

ISSUE:[UC-11-01:AuthzUseCase] The use case scenarios outlined in straw man 2 include explicitly only authn use cases. Should a use case featuring an authz conversation, such as a policy enforcement point (PEP) querying a policy decision point (PDP) for authorization for a user to execute an action? The use case would be included as follows:

Scenario N: Authorization Service

This use case illustrates an authorization service that provides authorization checks for resource access. This authorization service is expected to operate within a single security domain, where the owner of the resource also controls the policies at the Policy Decision Point corresponding to those resources.

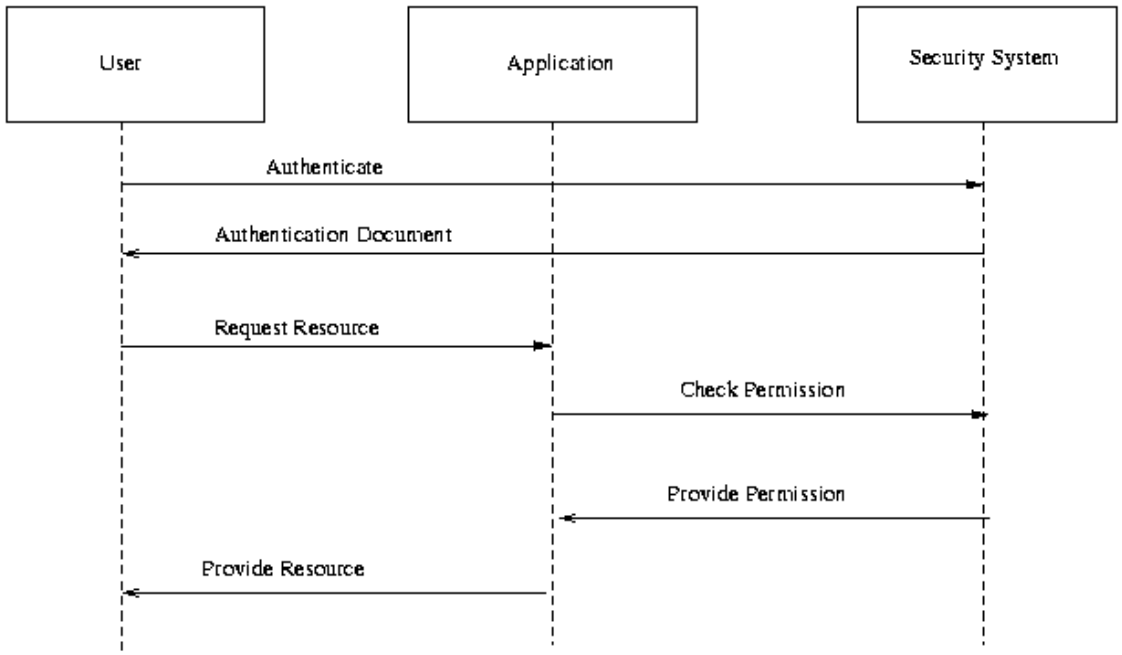



Fig X. Authorization Service.

Steps:

2137 1. User authenticates to security system.

2138 2. Security system provides authentication assertion to user.

2139 3. User requests resource from application (where the resource can be execution of an
2140 action, a file, a database record, etc.), providing authentication assertion.

2141 4. Application requests a check of permissions from the security server for user to access
2142 resource.

2143 5. Security system decides on user's authorization and provides permission information.

2144 6. Application provides resource to user.

2145 Possible Resolutions:

2146 1. SAML should include a use case describing an authorization service, as described above
2147 in Scenario N.

2148 2. No such use case is necessary.

2149 Status: Open ISSUE:[UC-11-02:AuthzFirstContact] A second scenario for the Authorization use
2150 case combines first contact single-sign-on (UC-1-05:FirstContact), authentication (UC-5-
2151 01:AuthNProtocol) and authorization.

2152 Scenario N+1: Authorization Service, First Contact with Authentication

2153 In this scenario, the client makes contact only with the application; there is not a separate
2154 authentication phase between the user and the security system.

2155 Steps:

2156 1. Client sends a single message containing both an authentication request and a resource
2157 request, to the application. A typical example would be an HTTP request with a client
2158 certificate or HTTP Basic Auth username and password.

2159 2. The application sends a combined authentication and authorization request to the security
2160 system.

2161 3. The security system replies with an authentication reference ([R-Reference]) and
2162 permission information.

2163 4. The application returns the authentication reference and the requested resource to the
2164 client.

2165 5. On subsequent requests, the client makes simple resource requests (including the
2166 authentication reference). These requests are identical to those in steps 3-6 of Scenario N.

Possible Resolutions:

1. SAML should include a scenario for an authorization service that also supports user authentication.

2. SAML should include a scenario where authentication and authorization requests can be combined in a single message exchange.

3. Both such scenarios should be added.

4. No such scenarios should be added.

Status: Open

# Group 12: Encryption

ISSUE:[UC-12-01:Encryption] UC-9-02:PrivacyStatement addresses the importance of sharing data only as needed between security zones (from asserting party to relying party). However, it is also important that data not be available to third parties, such as snoopers or untrusted intermediaries. One possible solution for implementors is to use secure channels between relying party and asserting party. Another is to use encryption, either with a shared secret or with public keys.

Possible Resolutions:

1. Allow for explicit use of encryption, such as XML Encryption (http://www.w3.org/Encryption/2001/). SAML messages would then be transferred securely on any protocol.

2. Require transport protocols to support some form of encryption. Examples: S/MIME for MIME, HTTP/S for HTTP.

Status: Open

# Group 13: Business Requirements

ISSUE:[UC-13-01:Scalability] Bob Morgan brought up several "business requirements" on security-use. One was scalability. This issue is a placeholder for further elaboration on the subject. A candidate requirement might be:

[CR-13-01-Scalability] SAML should be appropriate for high volume of messages, and for messages between parties made up of several physical machines.

Status: Open ISSUE:[UC-13-02:EfficientMessages] Philip Hallam-Baker's core assertions requirement document included several requirements that were efficiency-oriented. When that requirement document was merged into Straw Man 2, the efficiency requirements were

excluded.

One such requirement was:

> [CR-13-02-EfficientMessages] Should support efficient message exchange Integrity checks such as digital signature can add excessive overhead to messages.

Potential Resolutions:

1. Add this requirement to the use case and requirements document.
2. Leave this requirement out of use case and requirements document.

Status: Open ISSUE:[UC-13-03:OptionalAuthentication] Philip Hallam-Baker's core assertions requirement document included several requirements that were efficiency-oriented. When that requirement document was merged into Straw Man 2, the efficiency requirements were excluded.

One such requirement was:

> [CR-13-03-OptionalAuthentication] Authentication should be optional To Satisfy [R-EfficientMessages] Messages may omit authentication altogether.

Potential Resolutions:

1. Add this requirement to the use case and requirements document.
2. Leave this requirement out of use case and requirements document.

Status: Open ISSUE:[UC-13-04:OptionalSignatures] Philip Hallam-Baker's core assertions requirement document included several requirements that were efficiency-oriented. When that requirement document was merged into Straw Man 2, the efficiency requirements were excluded.

One such requirement was:

> [CR-13-04-OptionalSignatures] Signatures should be optional To Satisfy [R-EfficientMessages] Messages may use a shared secret and Message Authentication code for Authentication in place of digital signature.

Status: Open ISSUE:[UC-13-05:SecurityPolicy] Bob Morgan proposed a business-level requirement as follows:

> [CR-13-05-SecurityPolicy] Security measures in [OSSML] should support common institutional security policies regarding assurance of identity, confidentiality, and integrity.

Potential Resolutions:

2229 1. Add this requirement to the use case and requirements document.

2230 2. Leave this requirement out of use case and requirements document.

2231 Status: Open ISSUE:[UC-13-06:ReferenceReqt] Bob Morgan has questioned requirement [R-
2232 Reference] in that it is not specific enough. in particular, he said:

2233 "Goal [R-Reference] either needs more elaboration or (likely) needs to be dropped. What
2234 is a 'reference'? It doesn't have a standard well-understood security meaning nor is it
2235 defined in the glossary. This Goal seems to me to be making an assumption about a low-
2236 level mechanism for optimizing some of the transfers."

2237 One possible, more specific elaboration might be:

2238 [CR-13-06-1-Reference] SAML should define a data format for providing references to
2239 authentication and authorization assertions. Here, a "reference" means a token that may
2240 not be a full assertion, but can be presented to an asserting party to request a particular
2241 assertion.

2242 [CR-13-06-2-Reference-Message] SAML should define a message format for requesting
2243 authentication and authorization assertions using references.

2244 [CR-13-06-2-Reference-Size] SAML references should be small. In particular, they
2245 should be small enough to be transfered by Web browsers, either as cookies or as CGI
2246 parameters.

2247 Potential Resolutions:

2248 1. Replace [R-Reference] with these requirements.

2249 2. Leave [R-Reference] as it is.

2250 3. Remove mention of references entirely.

2251 Status: Open

# 2252 Document History

2253 • 5 Feb 2001 First version for Strawman 2.

2254 • 26 Feb 2001 Made the following changes:

2255 • Changed references to [OSSML] to SAML.

2256 • Added rewrites of Group 1 per Darren Platt.

2257 • Added rewrites of Group 3 per David Orchard.

- Added rewrites of Group 5 per Prateek Mishra.
- Added rewrites of Group 11 per Irving Reid.
- Converted the abbreviation "AuthC" (for "authentication") to "AuthN."
- Added Group 13.
- Added UC-1-12:SignOnService.
- Converted candidate requirement naming scheme from [R-Name] (as used in the main document) to [CR-issuenumber-Name], per David Orchard.
- Added UC-0-02:Terminology.
- Added UC-0-03:Arrows.
- Updated UC-9-02:PrivacyStatement with suggested requirements from Bob Morgan and Bob Blakley.
- Added UC-1-13:ProxyModel per Irving Reid.
- Added status indications for each issue.
- Recorded votes and conclusions for issue groups 1, 3, and 5.
- Added Zahid Ahmed's use cases for B2B transactions.
- Added Maryann Hondo's use case scenario for ebXML.
  Added comments to votes by Jeff Hodges, Bob Blakley.