Oasis Security Services Use Cases And Requirements

Consensus Draft 1, 30 May 2001

Purpose

This document describes the consensus of the Security Services Technical Committee as to the requirements and use cases for the Security Assertions Markup Language (SAML) to be created by the Oasis Security Services TC.   This is a draft committee specification document and as such will continue to be maintained and updated to reflect the work and decisions of the TC throughout the process of designing SAML.

Introduction

This document provides the 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 Modeling 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 represent 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.

Each use case scenario is also annotated with indicators showing what role the concrete actors (such as a Web user) play in the domain model, available here (draft-sstc-use-domain-05.pdf).

Readers will probably be interested in the accompanying glossary (draft-sstc-glossary-00.pdf) and issues list (draft-sstc-saml-issues-04.pdf)

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]** The SAML specification shall include support for Login functionality.
- **[R-UserSessionLogout]** In creating the SAML specification, the technical committee will do the prep work to ensure that logout, timein, and timeout will not be precluded from working with SAML later.
- **[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-Enveloped]**SAML messages and assertions should be fit to be enveloped in conversation-specific XML documents.
- **[R-Intermediaries]**SAML data structures (assertions and messages) will be structured in a way that they can be passed from an asserting party through one or more intermediaries to a relying party. The validity of a message or assertion can be established without requiring a direct connection between asserting and relying party.
- **[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-BackwardCompatibleExtensions]** Extension data in SAML will be clearly identified for all SAML processors, and will indicate whether the processor should continue if it does not support the extension.
- **[R-Confidentiality]** SAML data should be protected from observation by third parties or untrusted intermediaries.
- **[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
  - SOAP as a messaging protocol
  - ebXML as a messaging protocol
- **[R-BindingConfidentiality]** Bindings SHOULD (in the RFC sense) provide a means to protect SAML data from observation by third parties. Each protocol binding must include a description of how applications can make use of this protection. Examples: S/MIME for MIME, HTTP/S for HTTP.
- **[R-OptionalSigningAndEncryption]** The use of digital signatures and encryption to protect SAML assertions will be optional.

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.
- 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.
- SAML does not define a data format for expressing authorization policies.
- SAML does not need to specify a mechanism for additions, deletions or modifications to be made to assertions.
- SAML does not define a data format for encrypting assertions or messages independent of binding protocol. However, this non-goal will be revisited in a future version of the SAML spec after XML Encryption is published.

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.

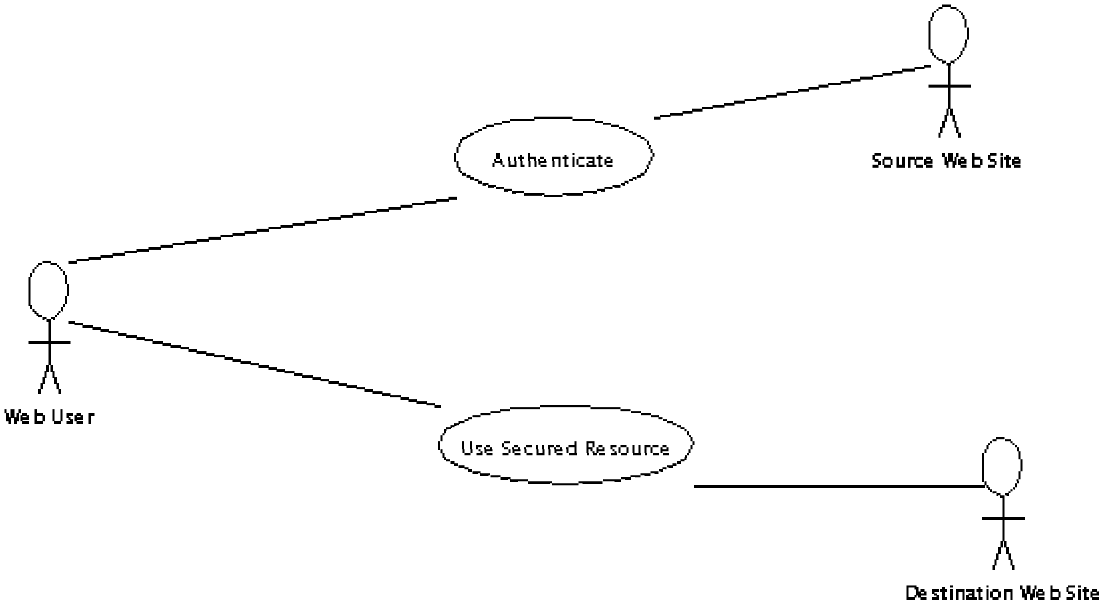

Fig 1. Single Sign-on.

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.

In this scenario, the source Web site acts as a Credentials Collector, Authentication Authority, and Attribute Authority. The Web user is the Principal. The destination Web site acts as a Policy Decision Point and Policy Enforcement Point.

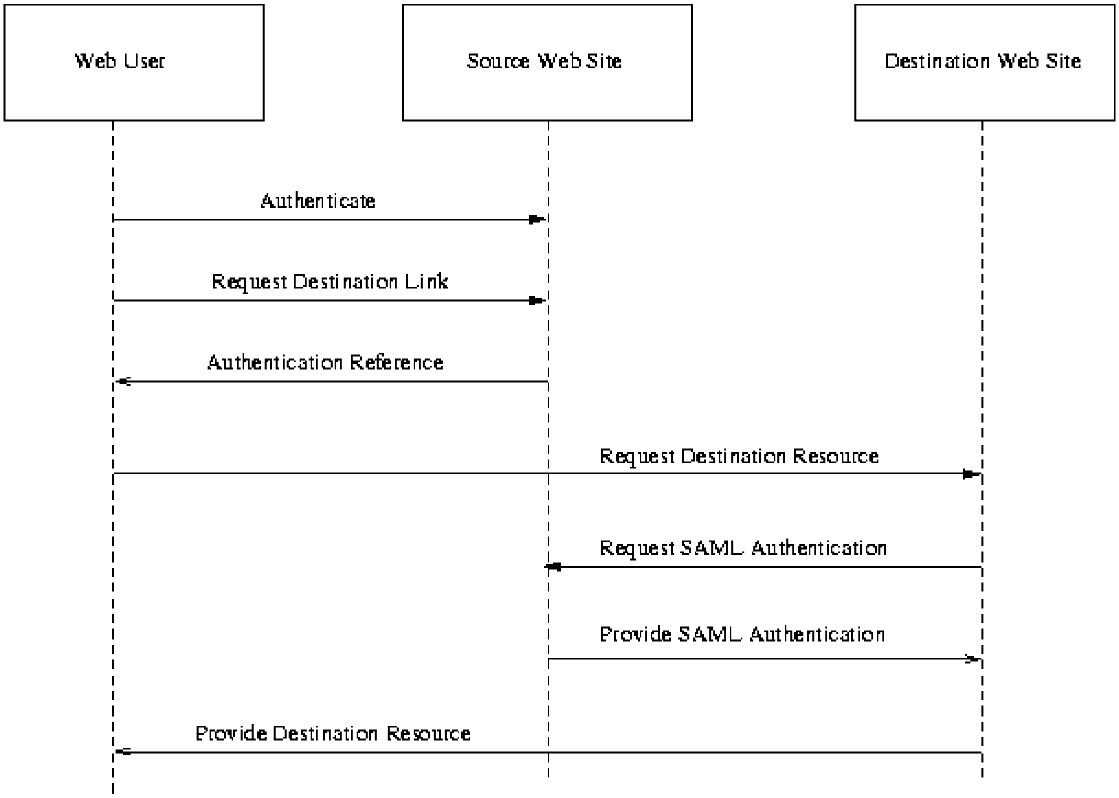
Web User
Source Web Site
Destination Web Site
Authenticate
Request Destination Link
Authentication Reference
Request Destination Resource
Request SAML Authentication
Provide SAML Authentication
Provide Destination Resource

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]**.

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

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

In this scenario, the source Web site acts as a Credentials Collector, Authentication Authority, and Attribute Authority. The Web user is the Principal. The destination Web site acts as a Policy Decision Point and Policy Enforcement Point.

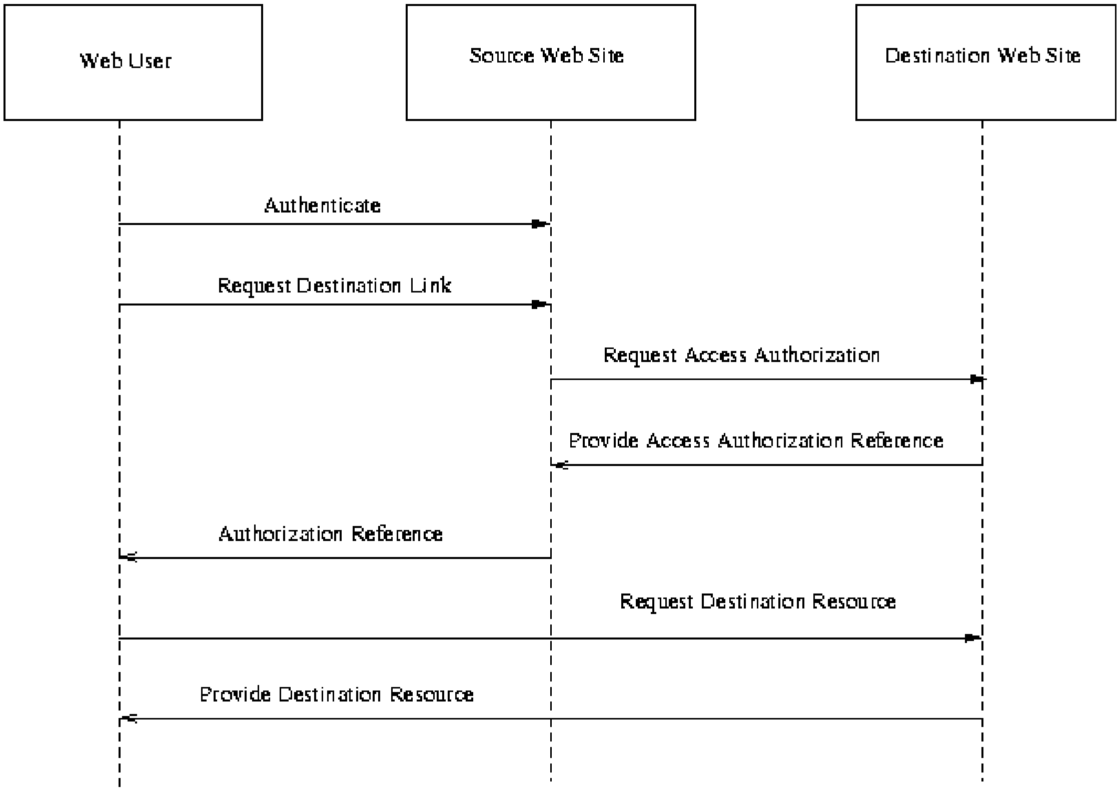
Web User
Source Web Site
Destination Web Site
Authenticate
Request Destination Link
Request Access Authorization
Provide Access Authorization Reference
Authorization Reference
Request Destination Resource
Provide Destination Resource

Fig 3. Single Sign-on, Push Model.

Steps:

1. Web user authenticates with source Web site.
2. Web user requests link to destination Web site.
3. Source Web site sends requests for Web user to use destination resource from destination Web site, pushing the authentication information (authentication assertion) for the user to the destination site. This assertion includes authorization attributes.

4. Destination Web site returns an authz decision reference to Source Web site, recording the decision to allow the user to access the resource.
5. Source Web site provides user with authz decision reference and redirects user to destination Web site.
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.

In this scenario, the security service acts as a Credentials Collector, Authentication Authority, and Attribute Authority. The Web user is the Principal. The destination Web sites act as Policy Decision Point and Policy Enforcement Point.

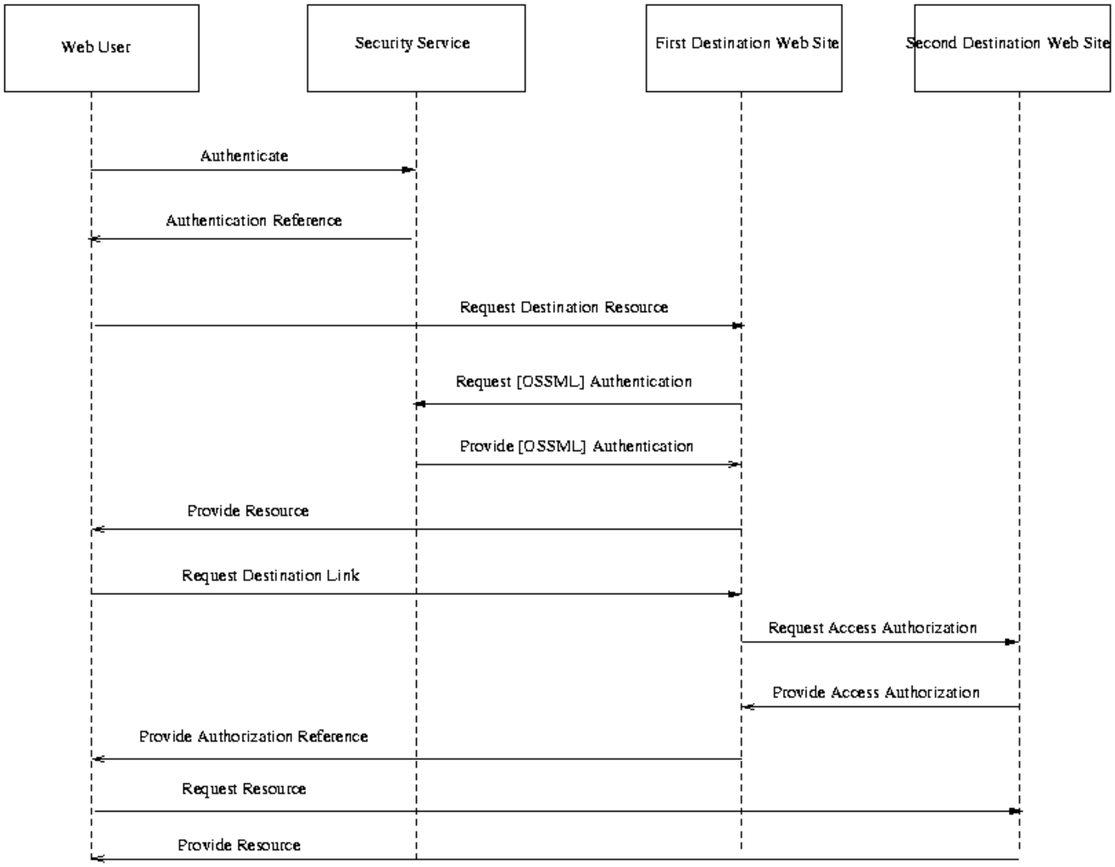
Web User
Security Service
First Destination Web Site
Second Destination Web Site
Authenticate
Authentication Reference
Request Destination Resource
Request [OSSML] Authentication
Provide [OSSML] Authentication
Provide Resource
Request Destination Link
Request Access Authorization
Provide Access Authorization
Provide Authorization Reference
Request Resource
Provide Resource

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]**.

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.

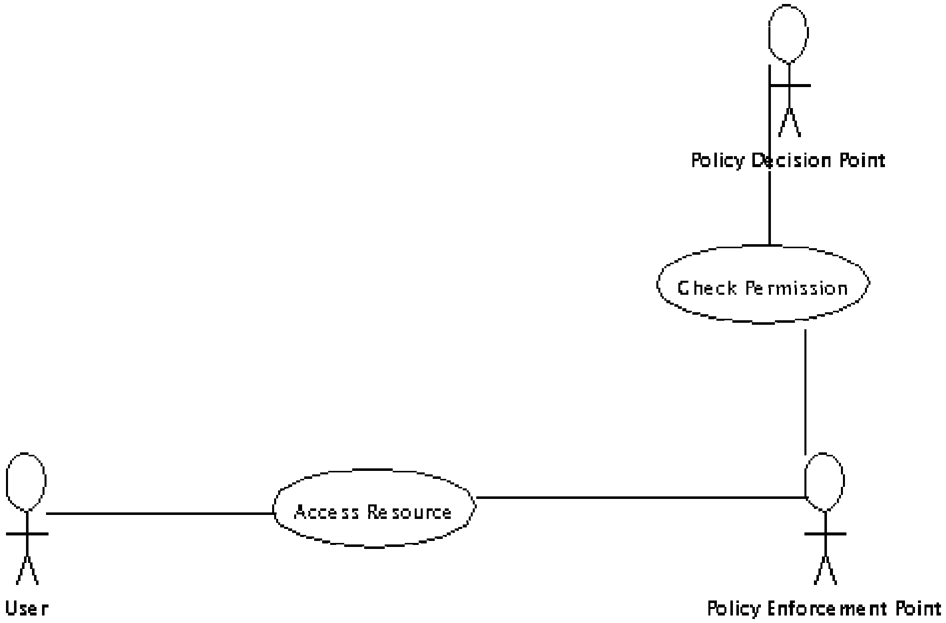

Fig 5. Authorization Service.

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.

In this scenario, the security service acts as a Credentials Collector, Authentication Authority, and Attribute Authority, as well as Policy Decision Point. The Web user is the Principal. The application acts as a Policy Enforcement Point.

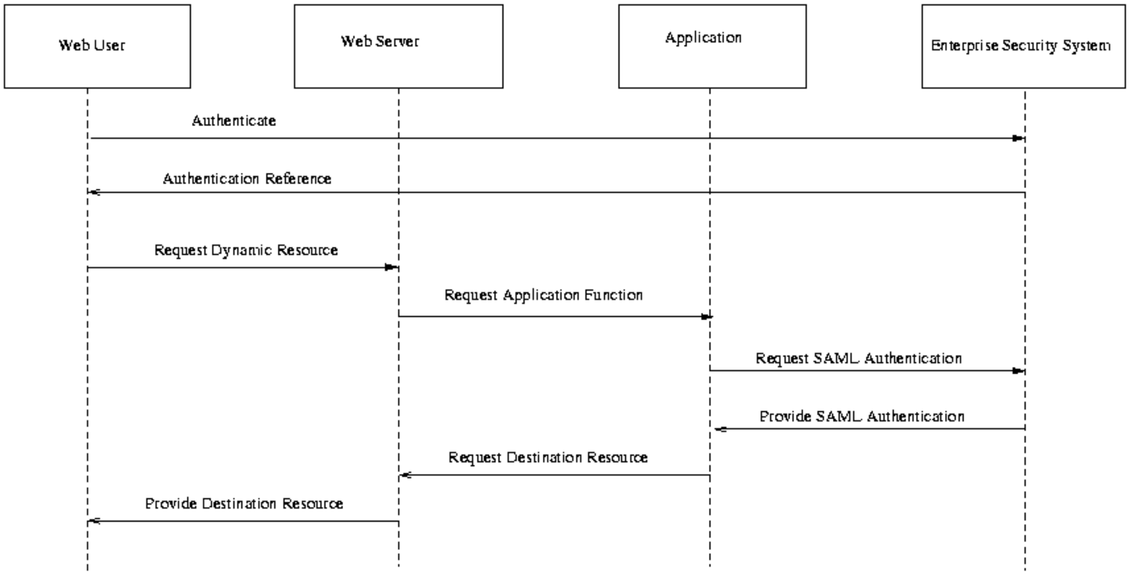
Web User
Web Server
Application
Enterprise Security System
Authenticate
Authentication Reference
Request Dynamic Resource
Request Application Function
Request SAML Authentication
Provide SAML Authentication
Request Destination Resource
Provide Destination Resource

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.

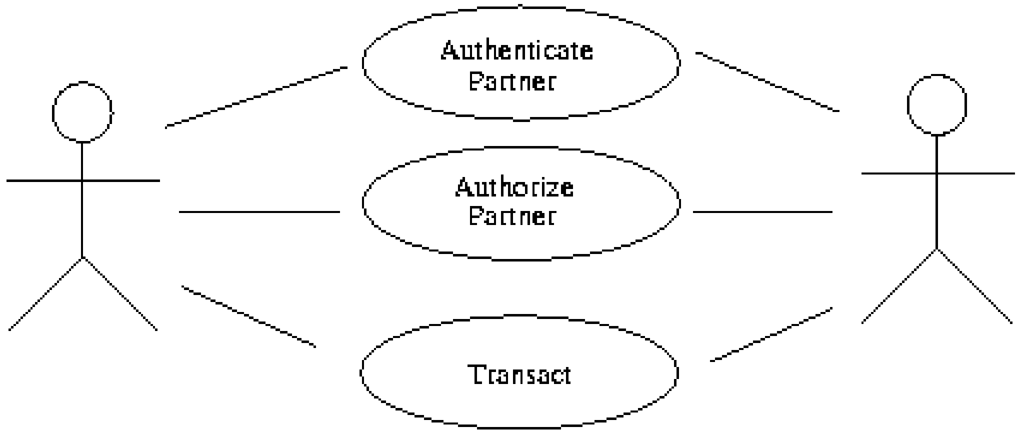

Fig 7. Back Office Transaction.

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.

In this scenario, the buyer and seller are principals. The buyer and seller security systems act as a Credentials Collector, Authentication Authority, and Attribute Authority, as well as Policy Decision Point. The Web user is the Principal. The buyer acts as a Policy Enforcement Point.

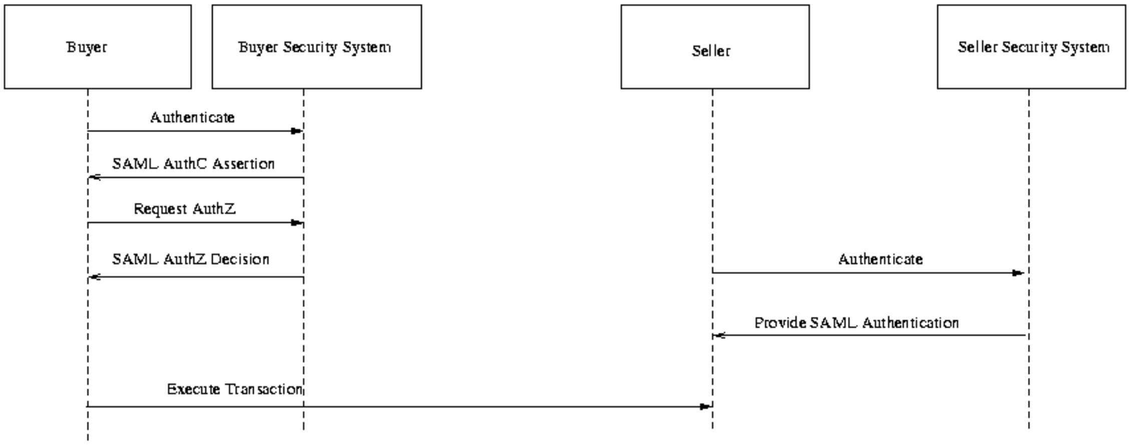

Fig 8. Back Office Transaction.

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 3-1. 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.

In this scenario, the buyer and seller are Principals. The third-party security service acts as a Credentials Collector, Authentication Authority, and Attribute Authority.

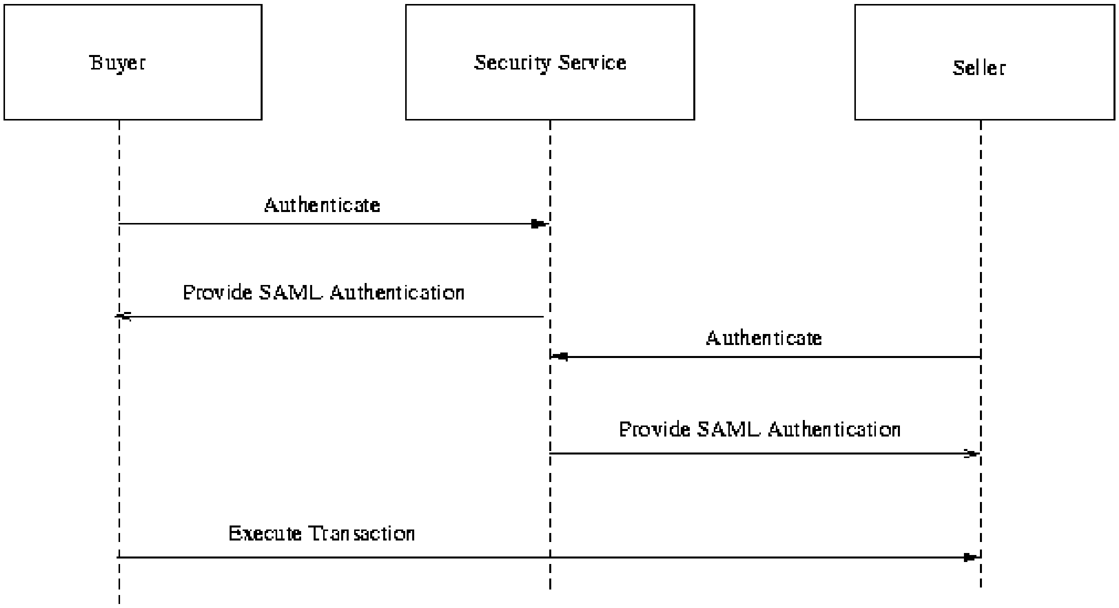

Fig 9. Back Office Transaction, Third Party Security Service.

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]**.

Scenario 3-3: Intermediary Add

In this use case scenario, two parties -- a buyer and a seller -- perform a transaction using a B2B exchange as an intermediary. The intermediary adds AuthN and AuthZ data to orders as they go through the system, giving additional points for decisions made by the parties.

In this scenario, the buyer and seller are Principals, and act as Policy Enforcement Point. The buyer and seller security security systems acts as Credentials Collector, Authentication Authority, and Attribute Authority, and Policy Decision Point. The exchange also acts as an Authentication Authority and Attribute Authority.

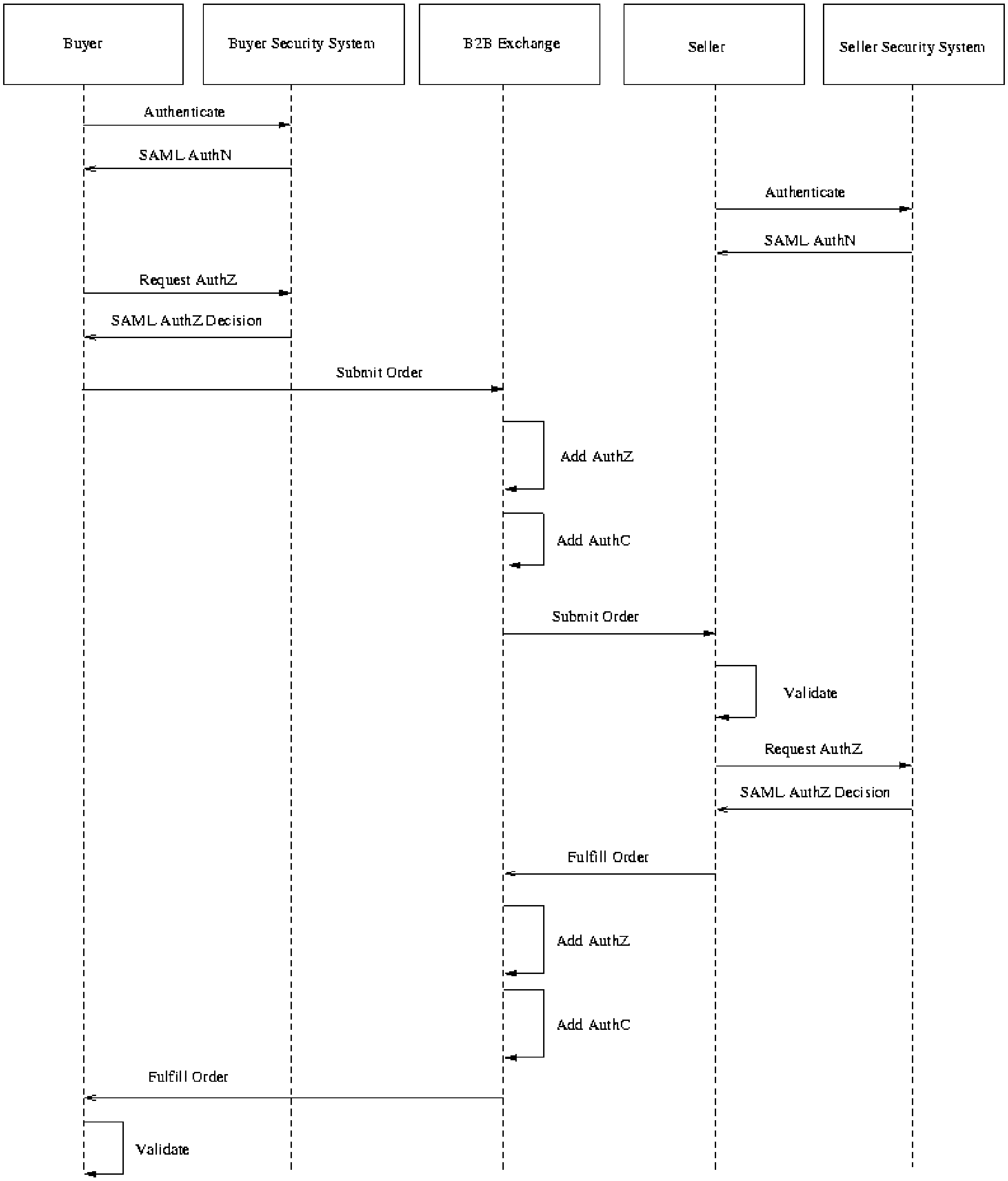

Fig 10. Intermediary Add.

Steps:

- Buyer authenticates to Buyer Security System.
- Buyer Security System provides a SAML AuthN assertion to Buyer, containing data about the authentication event and authorization attributes about the Buyer.
- Seller authenticates to Seller Security System.

- Seller Security System provides a SAML AuthN assertion to Seller, containing data about the authentication event and authorization attributes about the Seller.
- Buyer requests authorization from Buyer Security System to submit a given order.
- Buyer Security System provides a SAML AuthZ Decision assertion to Buyer, stating that Buyer is allowed to submit the order.
- Buyer submits order to B2B Exchange, providing AuthN assertion and AuthZ decision assertion.
- B2B exchange adds AuthN assertion data, specifying that the exchange authenticated the buyer (using the assertion). The exchange adds its own assertion, and does not modify the Buyer Security System assertion.
- B2B exchange adds AuthZ decision assertion data, stating that the Buyer is permitted to use the exchange to make this order. The exchange adds its own assertion, and does not modify the Buyer Security System assertion.
- B2B exchange submits order to Seller.
- Seller validates the order, using the assertions.
- Seller requests authorization from Seller Security System to fulfill a given order.
- Seller Security System provides a SAML AuthZ Decision assertion to Seller, stating that Seller is allowed to fulfill the order.
- Seller submits intention to fulfill the order to the B2B exchange, including AuthN assertions and AuthZ decision assertions.
- B2B exchange adds AuthN data, specifying that it used the original SAML AuthN assertion to authenticate the Seller. The exchange adds its own assertion, and does not modify the Seller Security System assertion.
- B2B exchange add AuthZ decision data, specifying that the seller is authorized to fulfill this order through the exchange. The exchange adds its own assertion, and does not modify the Seller Security System assertion.
- B2B exchange sends the order fulfillment to the Buyer.
- Buyer validates the order fulfillment based on AuthN assertion(s) and AuthZ decision assertion(s).

Associated requirements: **[R-AuthN]**, **[R-AuthZ]**, **[R-Intermediaries]**, **[R-MultiDomain]**, **[R-Enveloped]**.

Use Case 4: User Session

In this use case, two applications share a user session.

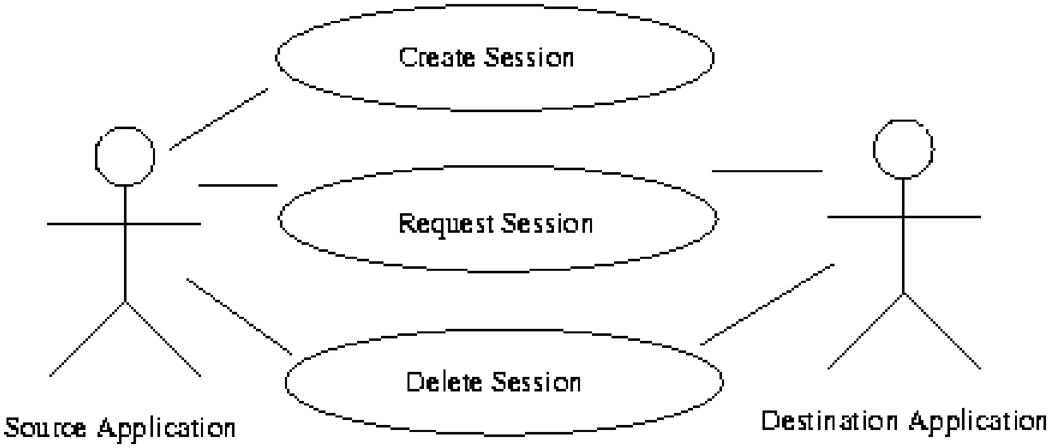

Fig 11. User Session.

1. Source application creates a session.
2. Source and/or destination application request the session.
3. Source and/or destination application delete the session.

Scenario 4-1: 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.

In this scenario, the Web user is the Principal. The source Web site acts as Credentials Collector, Authentication Authority, and Attribute Authority, and a Session Authority. The destination Web site acts as a Policy Decision Point and Policy Enforcement Point.

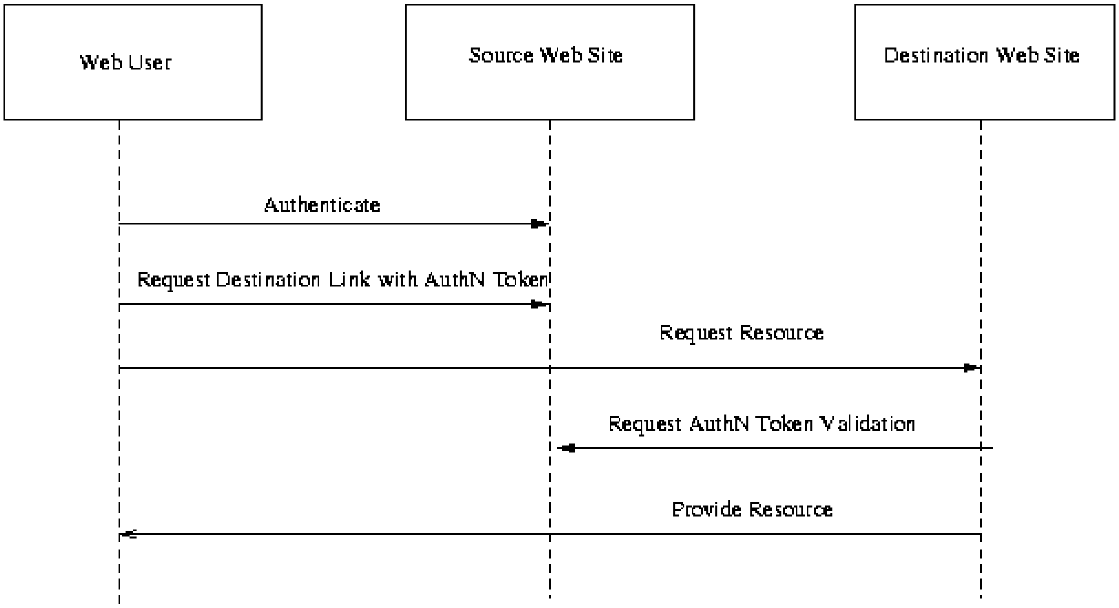

Fig. 12. Single Sign-on, User Session

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.

NOTE: *The following 2 scenarios (represented by fig.13 and fig.14) are non-normative. Instead they represent functionality that is intended to be added to SAML at some point in the future. The reason for including it here is to begin to satisfy the goal R-UserSessionLogout which is to "the technical committee will do the prep work to ensure that logout, timein, and timeout will not be precluded from working with SAML later."*

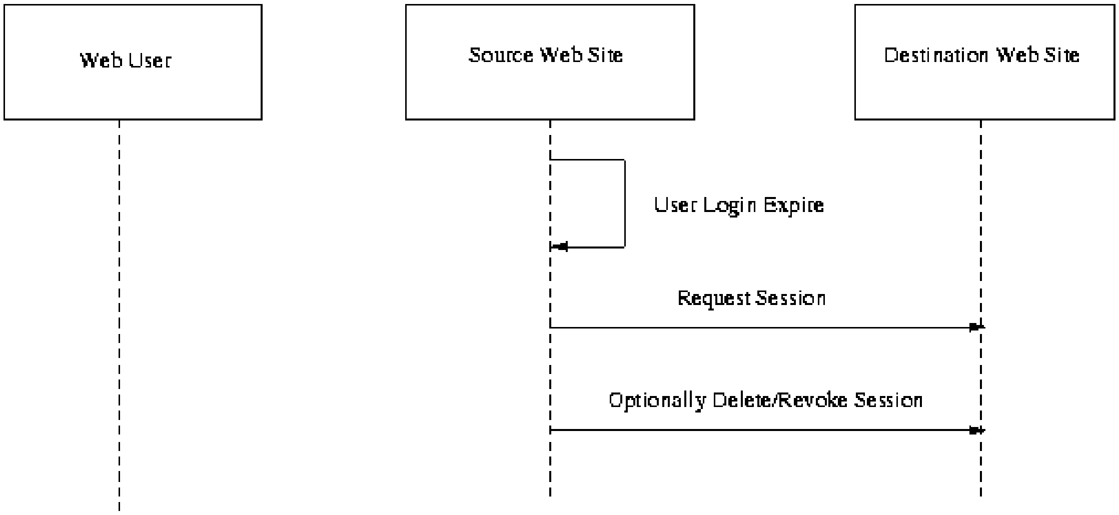

Fig. 13. User Session Timeout

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

1. The source Web site will query each participating Web site to determine if the user has been active on their Web site.
2. 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.

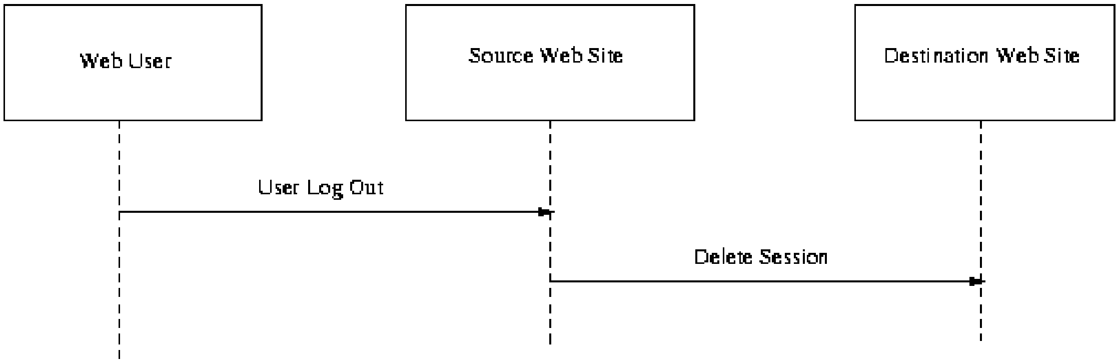

Fig. 14. User Session Logout

Logout

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].

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/.
- XML-Encryption: http://www.w3.org/Encryption/2001/.

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 Oasis Security Services TC (by Philip Hallam-Baker).
  - Corrected various editorial mistakes.
- *26 Feb 2001* -- Third draft.
  - Changed placeholder "[OSSML]" to new, official "SAML".
  - Re-ordered scenarios so that each group of scenarios followed an associated use case.
  - Rephrased use case scenario 1-2 per Nigel Edwards.
  - Updated use case scenario 1-3 per UC-1-02:ThirdParty.
  - Added [R-Anonymity].
  - Added [R-Pseudonymity].
  - Noted exchange of authz attributes, per UC-1-08:AuthZAttrs.
  - Added [R-AuthZDecision] and noted exchange of authz decisions, per UC-1-09:AuthZDecisions.
  - Edited [R-AuthN] and noted exchange of authn event data, per UC-1-10:AuthNEvent.
  - Added user session use case, per UC-3-1.
- *10 Apr 2001* -- Fourth draft.
  - Changed placeholder "[OSSML]" to new, official "SAML" in diagrams.
  - Removed non-goal for challenge-response protocol based on TC motion.
  - Modified non-goal for policies based on TC motion.
  - Removed non-goal for protection from third parties based on TC motion.

  - Added new use case for user sessions per TC motion, and moved session scenario there.
  - Added [R-Logout] per [UC-3-03:Logout].
  - Added [R-SessionTermination] per [UC-3-05:SessionTermination].
  - Added [R-BackwardCompatibleExtensions] per [UC-10-06:BackwardCompatibleExtensions].
  - Added [R-Confidentiality] per [UC-12-01:Confidentiality].
  - Added [R-BindingConfidentiality] per [UC-12-03:BindingConfidentiality].
  - Added non-goal for assertion and message encryption, per [UC-12-03:EncryptionMethod], and reference to XML-Encryption site.
  - Added [R-Enveloped] per [UC-7-02:Enveloped].
  - Added [R-Intermediaries] per [UC-8-01:Intermediaries].
  - Added [R-Intermediaries] per [UC-8-01:Intermediaries].
  - Added Use Case Scenario 3-3 per [UC-8-02:IntermediaryAdd].
  - Added non-goal for atomic assertions per [UC-8-05:AtomicAssertion].
- *15 May 2001* -- Fifth draft.
  - Added [R-UserSessionLogout] and modified [R-UserSession] to reflect decisions of TC from second face to face meeting.
  - Added text to denote figures 13 and 14 as non-normative to reflect a TC decision.
  - Changed the Purpose section to indicate that this is a consensus draft.
  - Added [R-OptionalSigningAndEncryption] per decision of 5/15/01 concall.
- *30 May 2001* -- Sixth draft.
  - Updated Purpose section to note that this is a working draft per 5/29 concall

Editors and Contributors

Editors

Darren Platt, Securant Technologies
Evan Prodromou, Securant Technologies

Contributors

Zahid Ahmed, Commerce One
Carlisle Adams, Entrust Technologies
Bob Blakely, Tivoli
Nigel Edwards, Hewlitt Packard
Kelly Emo, Jamcracker
Marlena Erdos, IBM
Jeff Hodges, Oblix
MaryAnn Hondo, IBM
Hal Lockhart, Entegrity Solutions
Eve Maler, Sun Microsystems
Prateek Mishra, Netegrity
Bob Morgan, University of Washington
Tim Moses, Verisign
David Orchard, Jamcracker
Gilbert Pilz, Jamcracker
Irving Reid, Baltimore Technologies