draft-sstc-saml-reqs-00.doc

15

16 Purpose

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

19 Introduction

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

23 Notes on This Document

24 Requirements are specified as a list of goals and non-goals for the project.

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

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

32 • Stick figures represents actors or roles in a scenario. These can be human beings or
33 software systems.

34 • Ellipses represent use cases, i.e. actions or units of functionality in a system.

35 • Lines between actors and use cases indicate a participation of the actor in the use case.
36 Note that no direction or payload of data flow is expressed by the lines between actors
37 and use cases.

38 Use-case diagrams capture high-level functionality of a system or interaction without providing
39 excessive implementation detail.

40 The document uses UML sequence diagrams to illustrate detailed use case scenarios. For quick
41 reference, a sequence diagram works as follows:

42 • Boxes at the top of the diagram represent an actor in the scenario.

43 • Arrows with a solid head represent a message sent from one actor to another. The arrow
44 points from sender to receiver.

45   • Arrows with a line head represent the return value of a message. The arrow points from
46   the receiver of the earlier message to the sender.

47   • A dotted line ("swim lane") running down the diagram from a box indicates that arrows
48   whose endpoints (tail or head) is on the line apply to that actor.

49   • Intersections between arrows and dotted lines are meaningless.

50   • Vertical layout represents time. Messages (arrows) farther down on the page happen after
51   messages higher on the page.

52   • Horizontal layout has no formal meaning. Since right-pointing arrows look better, actors
53   that initiate a scenario tend to appear leftward of actors they send messages to.

54 Note that sequence diagrams are often used for more concrete design, and that actors and
55 messages are often objects and object methods. They provide value for this document in that they
56 give a clearly ordered message layout. The actors and messages in the sequence diagrams below
57 are more properly roles in a scenario and actions associated with that scenario.

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

60 Readers will probably be interested in the accompanying glossary and issues list.

61 Requirements

62 The requirements describe the scope of the SAML standard.

63 Goals

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

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

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

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

74 • [R-Logout] SAML shall support a message format for to end a session due to logout by
75   the principal.

76 • [R-SessionTermination] SAML shall support a message format for timeout of a SAML-
77   level session. Here, "termination" is defined as the ending of a SAML-level session by a
78   security system not based on user input. For example, if the user has not used any of the
79   application-level sub-sessions for a set amount of time, the session may be considered
80   "timed out."

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

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

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

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

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

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

94 • [R-Enveloped]SAML messages and assertions should be fit to be enveloped in
95   conversation-specific XML documents.

96 • [R-Intermediaries]SAML data structures (assertions and messages) will be structured in a
97   way that they can be passed from an asserting party through one or more intermediaries
98   to a relying party. The validity of a message or assertion can be established without
99   requiring a direct connection between asserting and relying party.

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

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

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

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

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

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

109 • [R-BackwardCompatibleExtensions] Extension data in SAML will be clearly identified
110   for all SAML processors, and will indicate whether the processor should continue if it
111   does not support the extension.

112 • [R-Confidentiality] SAML data should be protected from observation by third parties or
113   untrusted intermediaries.

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

117   • standard commercial browsers

118   • HTTP as a transport protocol

119   • MIME as a packaging protocol

120   • SOAP as a messaging protocol

121   • ebXML as a messaging protocol

122   •   [R-BindingConfidentiality] Bindings SHOULD (in the RFC sense) provide a means to
123   protect SAML data from observation by third parties. Each protocol binding must include
124   a description of how applications can make use of this protection. Examples: S/MIME for
125   MIME, HTTP/S for HTTP.

126   **Non-Goals**

127   •   SAML will not propose any new cryptographic technologies or models for security;
128   instead, the emphasis is on description and use of well-known security technologies
129   utilizing a standard syntax (markup language) in the context of the Internet.

130   •   Non-repudiation services and markup are outside the scope of SAML.

131   •   SAML does not provide for negotiation between authorities about trust between domains
132   and realms or the inclusion of optional data. Trust negotiations must be made out-of-
133   band.

134   •   SAML does not define a data format for expressing authorization policies.

135   •   SAML does not need to specify a mechanism for additions, deletions or modifications to
136   be made to assertions.

137   •   SAML does not define a data format for encrypting assertions or messages independent
138   of binding protocol. However, this non-goal will be revisited in a future version of the
139   SAML spec after XML Encryption is published.

140 Use Cases And Scenarios

141   This section provides a set of high-level use cases for SAML and use case scenarios that
142   illustrate the use case. They give a very abstract view of the intended use of the SAML format.
143   Each use case has a short description, a use case diagram in UML format, and a list of the steps
144   involved in the case.

145   Note that, for each use case, the mechanics of how the actions are performed is not described.
146   More detail provided in the detailed use case scenarios. Each of these high-level use cases has
147   one or more specializations in the detailed use-case scenarios.

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

151 # Use Case 1: Single Sign-On

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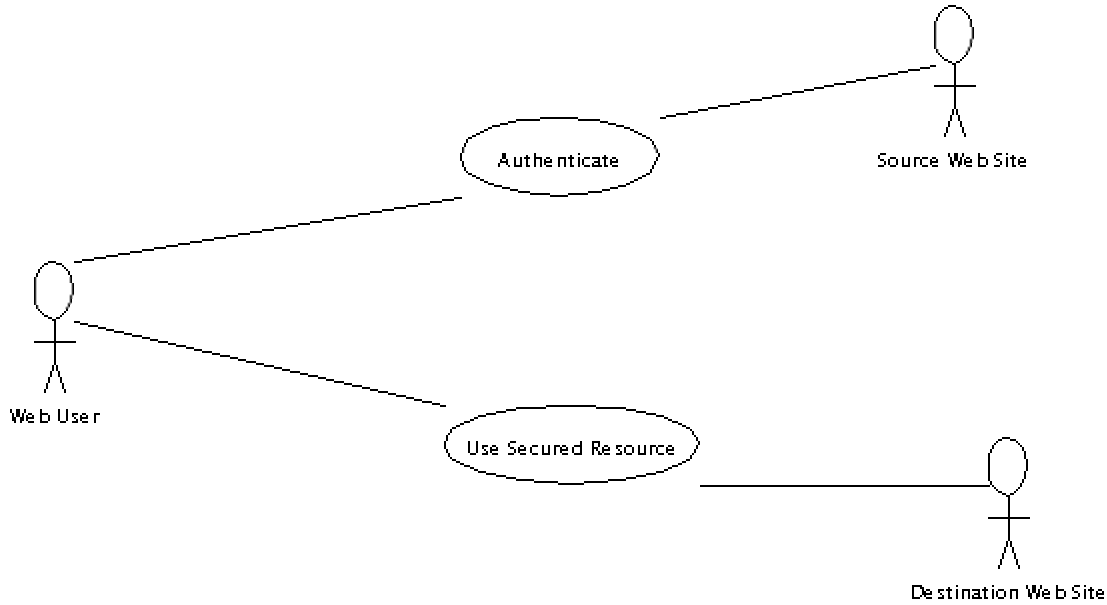153 resource at another Web site, without directly authenticating to that Web site.

154
155 Fig 1. Single Sign-on.

156 Steps:

157   1. Web user authenticates to the source Web site.

158   2. Web user uses a secured resource at the destination Web site.

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

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

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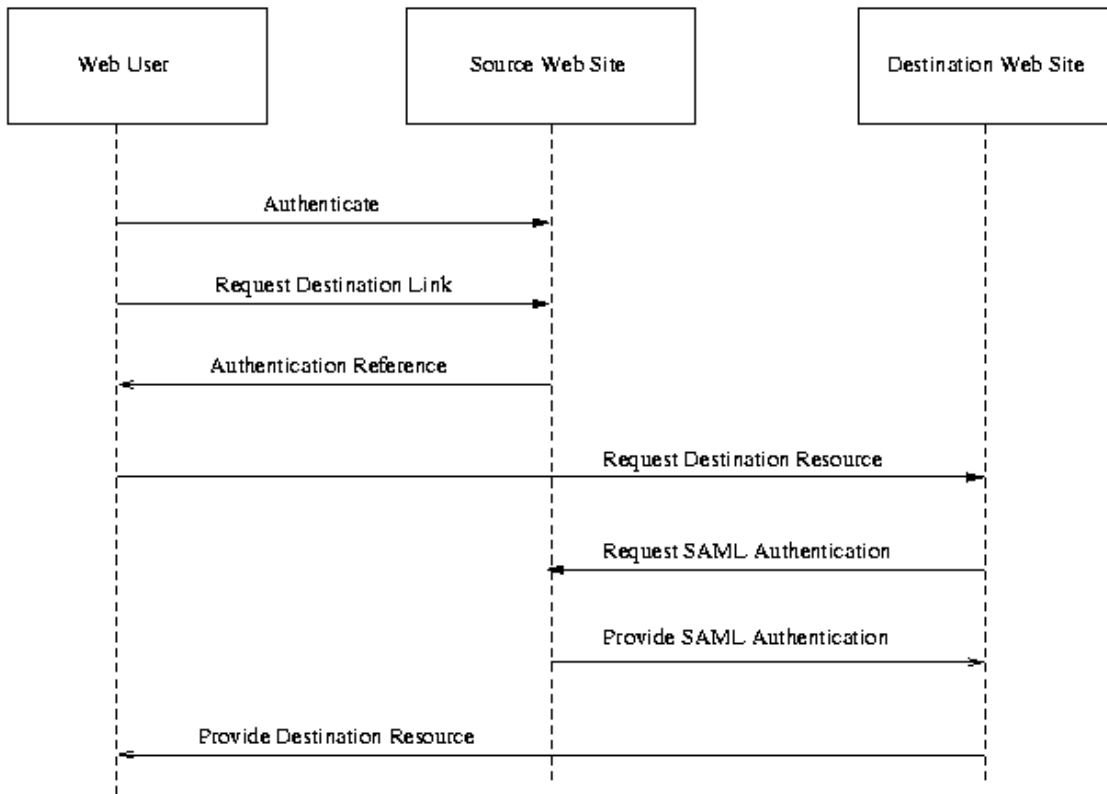165 Decision Point and Policy Enforcement Point.

| 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

166
167   Fig 2. Single Sign-on, Pull Model.

168   Steps:

169   1. Web user authenticates with source Web site.

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

171   3. Source Web site provides user with authentication reference (AKA "name assertion
172   reference"), and redirects user to destination Web site.

173   4. Web user requests destination Web site resource, providing authentication reference.

174   5. Destination Web site requests authentication document ("name assertion") from source
175   Web site, passing authentication reference.

176   6. Source Web site returns authentication document. This document includes authn event
177   description and authz attributions about the Web user.

178   7. Destination Web site provides resource to Web user.

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

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

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

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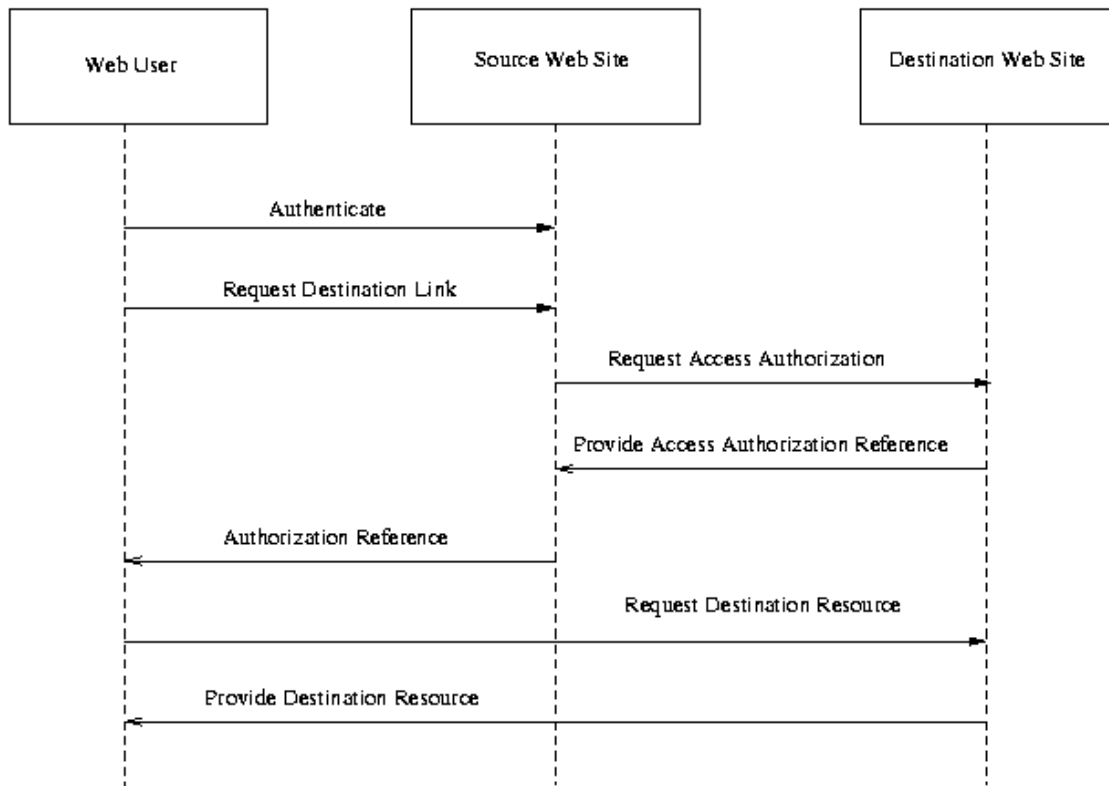

187
188   Fig 3. Single Sign-on, Push Model.

189   Steps:

190   1. Web user authenticates with source Web site.

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

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

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

197   5. Source Web site provides user with authz decision reference and redirects user to
198   destination Web site.

199   6. User requests destination resource from destination Web site, providing authz decision
200   reference.

201   7. Destination Web site provides resource to Web user.

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

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

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

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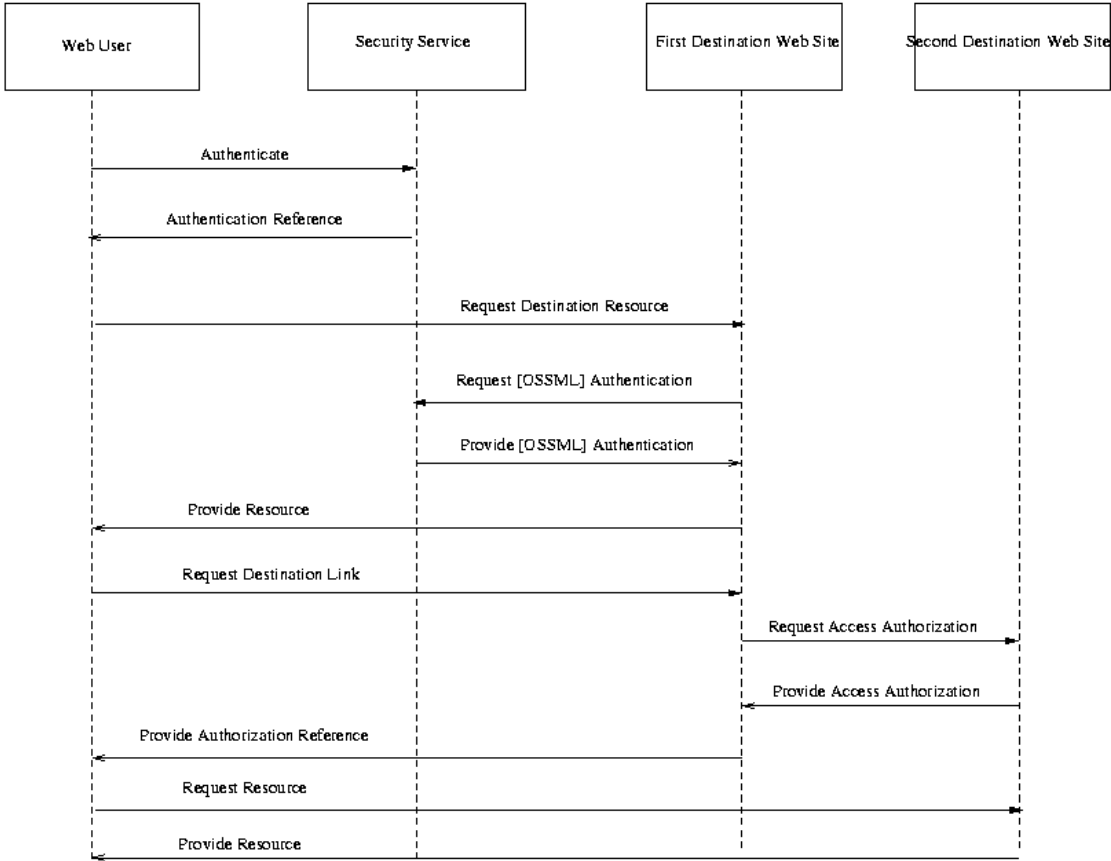212 Decision Point and Policy Enforcement Point.

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.

226   8. First destination Web site requests access authorization from second destination Web site,
227   providing third-party security service authentication document for user.

228   9. Second destination Web site provides access authorization, returning an authz decision
229   reference.

230   10. First destination Web site provides authz decision reference to Web user.

231   11. Web user requests resource from second destination Web site, providing authz decision
232   reference.

233   12. Second destination Web site provides resource.

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

236 Use Case 2: Authorization Service

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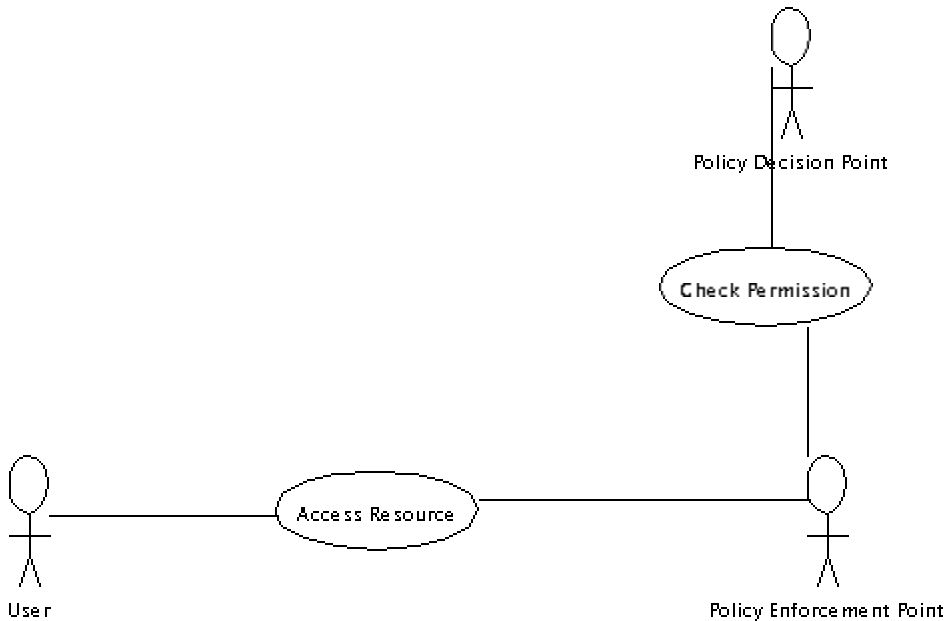
240 The PDP provides an authorization service to the PEP.

241
242 Fig 5. Authorization Service.

243 Steps:

244   1. User accesses a resource controlled by PEP.

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

246 Scenario 2-1: Application Chain

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

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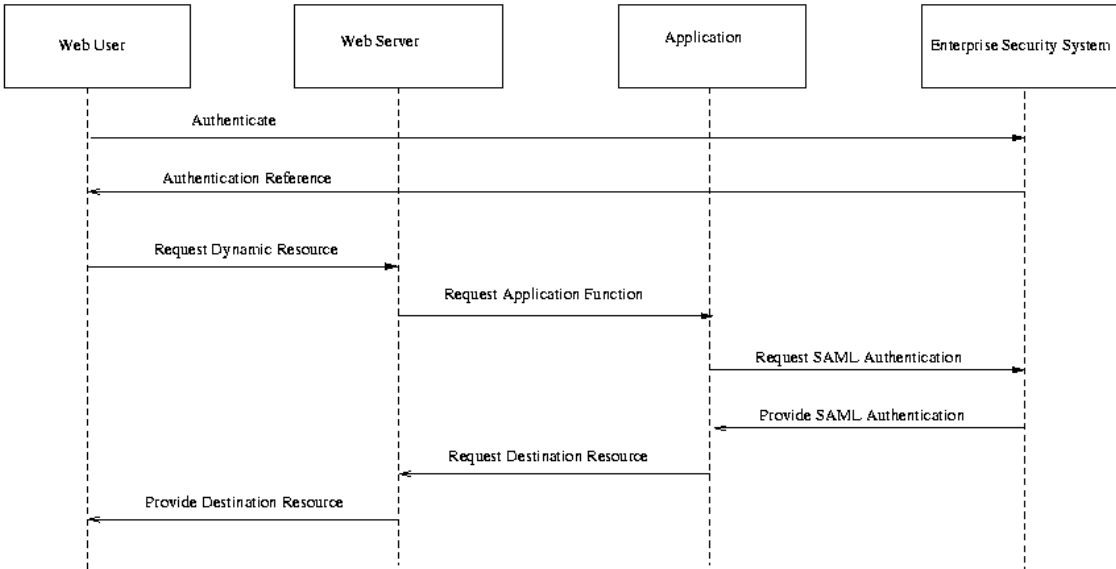252 application acts as a Policy Enforcement Point.

253
254   Fig 6. Application Chain.

255   Steps:

256   1. Web user authenticates with enterprise security system. Note that authentication may be
257   through e.g. the Web server.

258   2. Enterprise security system provides an authentication reference to Web user.

259   3. Web user requests a dynamic resource from Web server, providing authentication
260   reference.

261   4. Web server requests application function from application on behalf of Web user,
262   providing Web user's authentication reference.

263   5. Application requests authentication document from enterprise security system,
264   corresponding to Web user's authentication reference.

265   6. Enterprise security system provides authentication document, including authorization
266   attributes for the Web user, and authn event description.

267   7. Application performs application function for Web server.

268   8. Web server generates dynamic resource for Web user.

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

271 # Use Case 3: Back Office Transaction

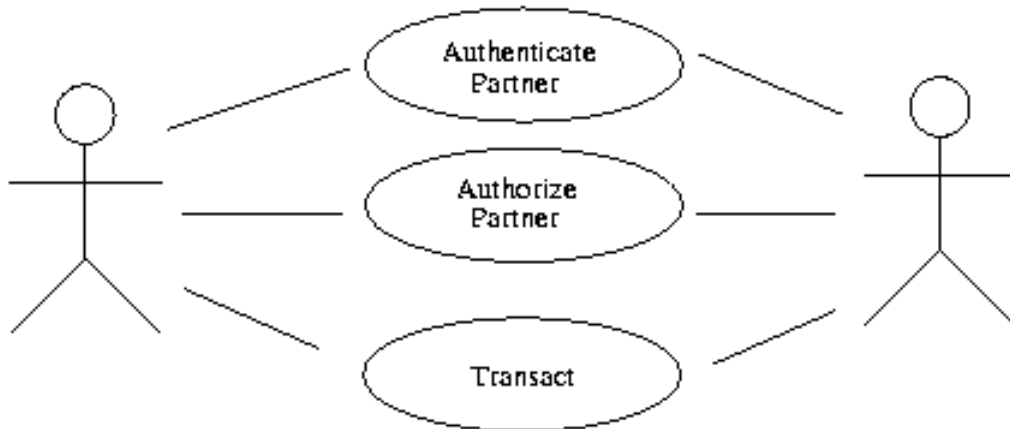272 In this use case, two agents, a buyer and a seller, attempt to execute a business transaction.

273 Fig 7. Back
274 Office Transaction.

275   1. Buyer and seller authenticate that their partner in the transaction is the partner they expect
276   to transact with.

277   2. Buyer and seller check permission of partner to execute transaction.

278   3. Buyer and seller execute the transaction.

279 ## Scenario 3-1: Back Office Transaction

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

284 In this scenario, the buyer and seller are principals. The buyer and seller security systems act as a
285 Credentials Collector, Authentication Authority, and Attribute Authority, as well as Policy
286 Decision Point. The Web user is the Principal. The buyer acts as a Policy Enforcement Point.
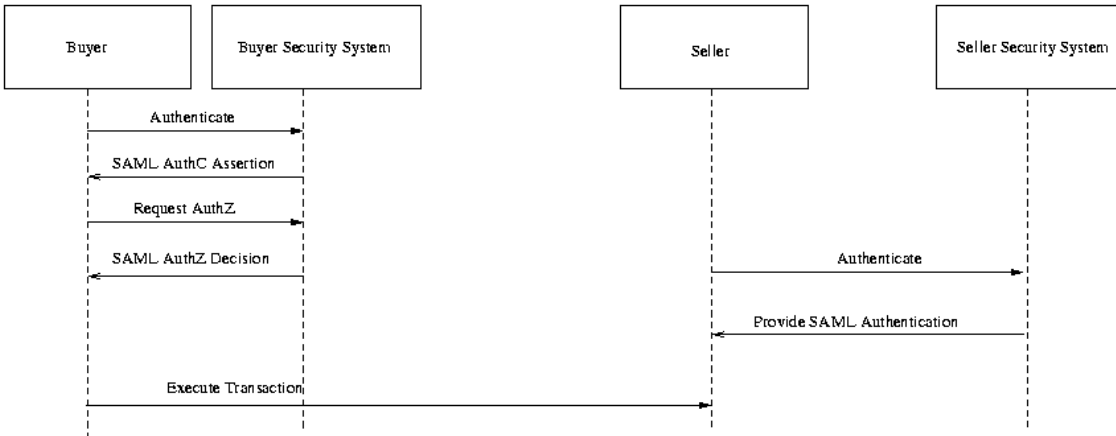
287
288   Fig 8. Back Office Transaction.

289   Steps:

290   1. Buyer authenticates with buyer security system.

291   2. Buyer security system provides authentication document to buyer.

292   3. Seller authenticates with seller security system.

293   4. Seller security system provides authentication document to seller.

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

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

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

298   This scenario is similar to scenario 3-1. The same two parties, buyer and seller, wish to perform a
299   transaction. In this case, however, each authenticates to a third-party security service responsible.
300   The buyer and seller exchange authentication data provided by their security systems to
301   authenticate the transaction.

302   In this scenario, the buyer and seller are Principals. The third-party security service acts as a
303   Credentials Collector, Authentication Authority, and Attribute Authority.
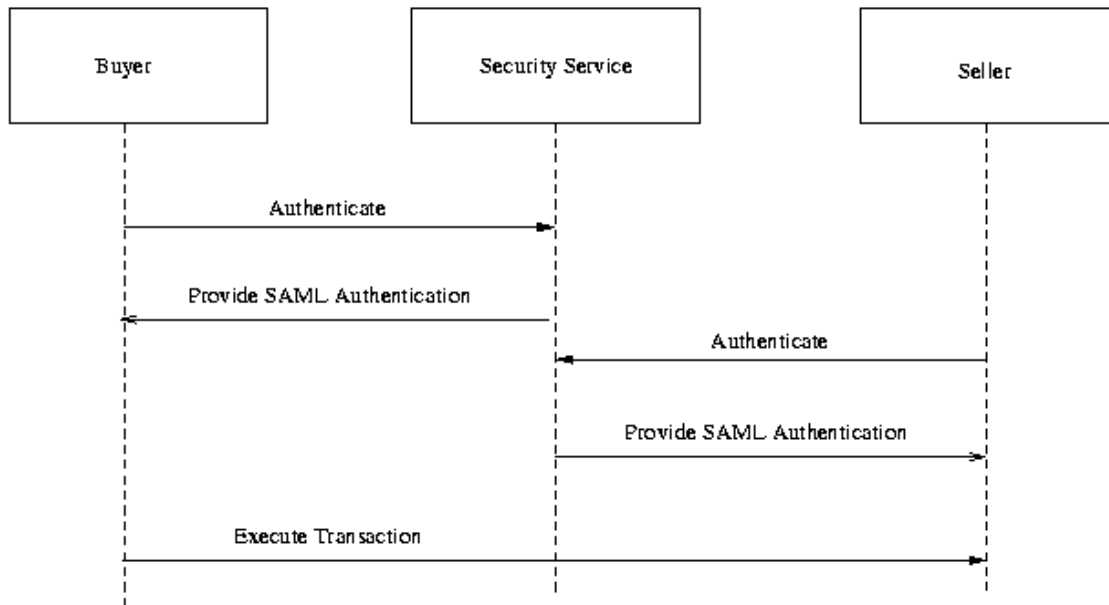
304
305 Fig 9. Back Office Transaction, Third Party Security Service.

306 Steps:

307   1. Buyer authenticates with security service.

308   2. Security service provides authentication document to buyer.

309   3. Seller authenticates with security service.

310   4. Security service provides authentication document to seller.

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

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

Scenario 3-3: Intermediary Add

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

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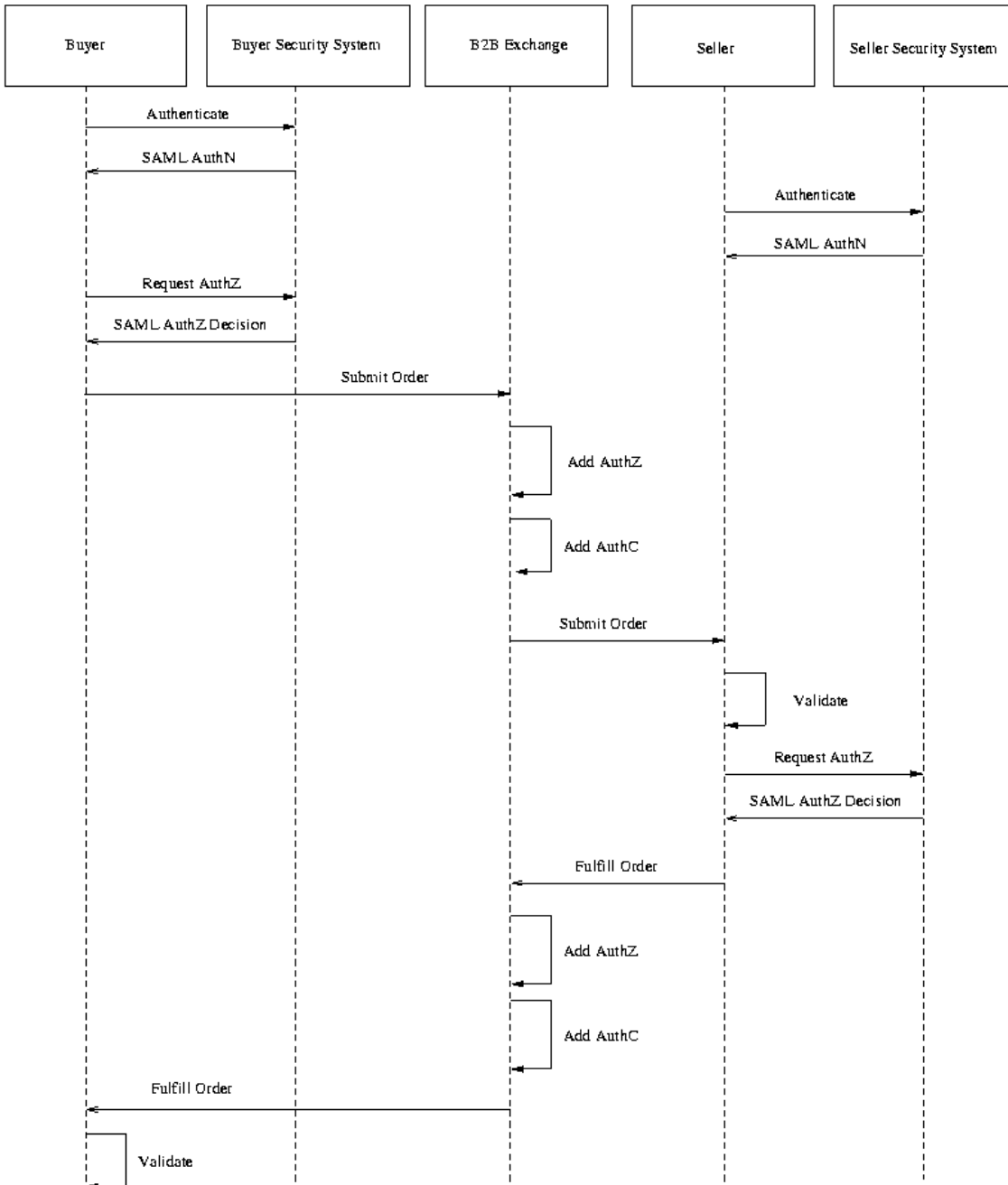321 Authentication Authority and Attribute Authority.

322
323 Fig 10. Intermediary Add.

324 Steps:

325 • Buyer authenticates to Buyer Security System.

326 • Buyer Security System provides a SAML AuthN assertion to Buyer, containing data

327   about the authentication event and authorization attributes about the Buyer.

328   • Seller authenticates to Seller Security System.

329   • Seller Security System provides a SAML AuthN assertion to Seller, containing data
330   about the authentication event and authorization attributes about the Seller.

331   • Buyer requests authorization from Buyer Security System to submit a given order.

332   • Buyer Security System provides a SAML AuthZ Decision assertion to Buyer, stating that
333   Buyer is allowed to submit the order.

334   • Buyer submits order to B2B Exchange, providing AuthN assertion and AuthZ decision
335   assertion.

336   • B2B exchange adds AuthN assertion data, specifying that the exchange authenticated the
337   buyer (using the assertion). The exchange adds its own assertion, and does not modify the
338   Buyer Security System assertion.

339   • B2B exchange adds AuthZ decision assertion data, stating that the Buyer is permitted to
340   use the exchange to make this order. The exchange adds its own assertion, and does not
341   modify the Buyer Security System assertion.

342   • B2B exchange submits order to Seller.

343   • Seller validates the order, using the assertions.

344   • Seller requests authorization from Seller Security System to fulfill a given order.

345   • Seller Security System provides a SAML AuthZ Decision assertion to Seller, stating that
346   Seller is allowed to fulfill the order.

347   • Seller submits intention to fulfill the order to the B2B exchange, including AuthN
348   assertions and AuthZ decision assertions.

349   • B2B exchange adds AuthN data, specifying that it used the original SAML AuthN
350   assertion to authenticate the Seller. The exchange adds its own assertion, and does not
351   modify the Seller Security System assertion.

352   • B2B exchange add AuthZ decision data, specifying that the seller is authorized to fulfill
353   this order through the exchange. The exchange adds its own assertion, and does not
354   modify the Seller Security System assertion.

355   • B2B exchange sends the order fulfillment to the Buyer.

356   • Buyer validates the order fulfillment based on AuthN assertion(s) and AuthZ decision

357   assertion(s).

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

360 # Use Case 4: User Session

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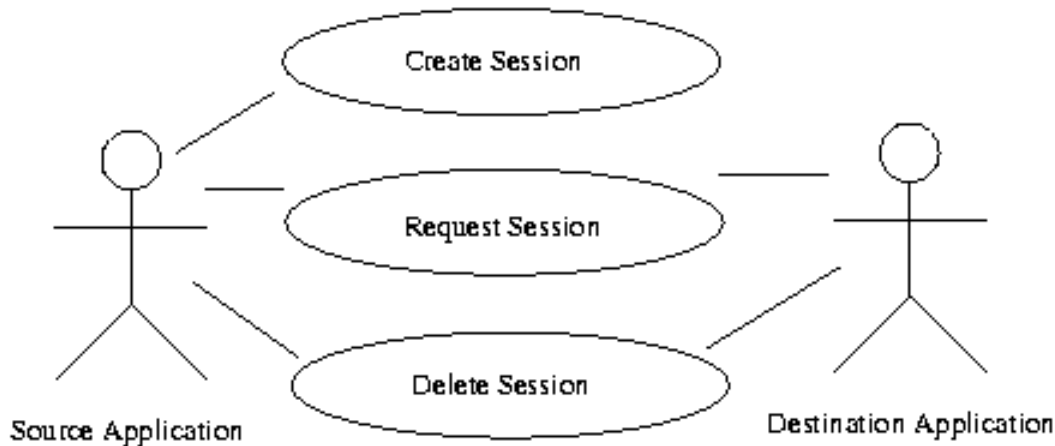

362 Fig 11.
363 User Session.

364   1. Source application creates a session.

365   2. Source and/or destination application request the session.

366   3. Source and/or destination application delete the session.

367 ## Scenario 4-1: Single Sign-on, User Session

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

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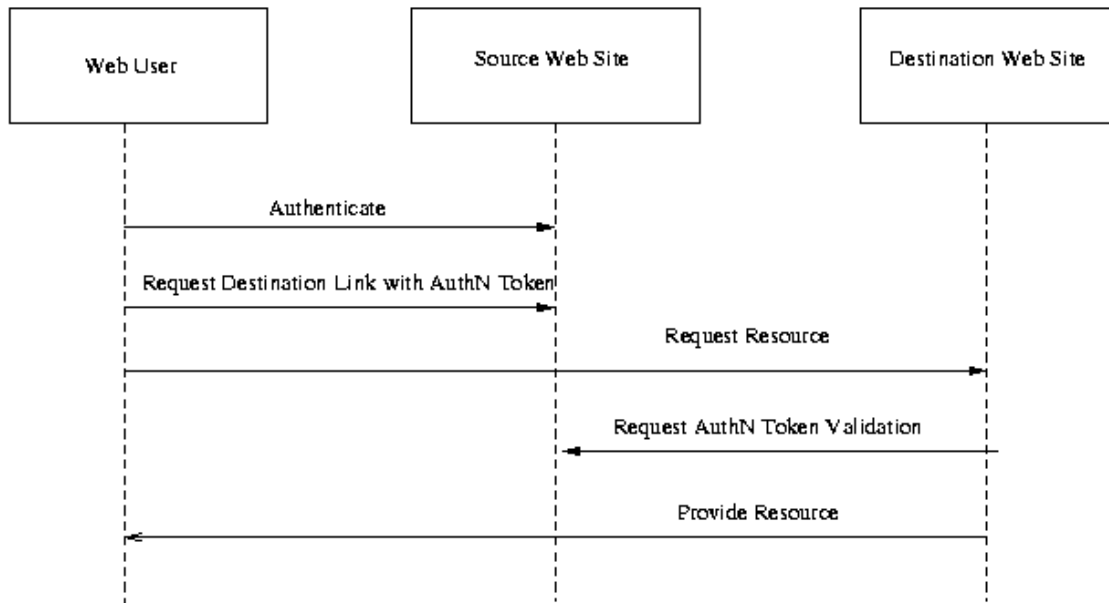372 site acts as a Policy Decision Point and Policy Enforcement Point.

373   Fig.
374   12. Single Sign-on, User Session

375   Steps:

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

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

380   3. User requests resource represented by link on destination Web site, including reference.

381   4. Destination Web site requests validation of authentication reference from source Web
382   site.

383   5. Source Web site returns success or failure, optionally additional session information.

384   6. Destination Web site returns Web site to user.
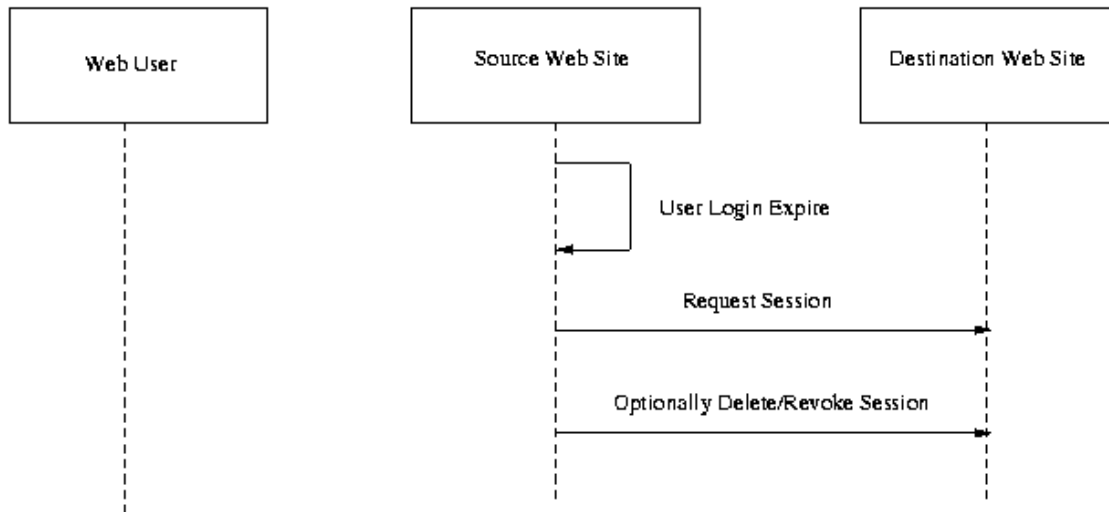
385   Fig.
386   13. User Session Timeout

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

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

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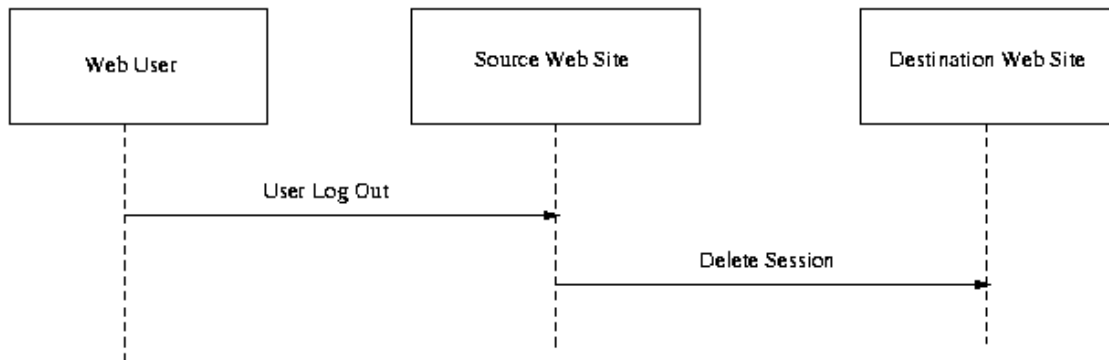391   period, the destination Web sites are instructed to delete the session.

392   Fig.
393   14. User Session Logout

394   Logout

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

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

397   Associated requirements: [R-AuthN], [R-AuthZ], [R-PullMessage], [R-PushMessage], [R-

398   MultiDomain], [R-Bindings] (standard commercial browsers), [R-Reference], [R-UserSession].

399 References

400   This document is derived from the following sources:

401   •   Security Services Markup Language v0.8a, Prateek Mishra et. al.

402   •   AuthXML: A Specification for Authentication Information In XML v0.3, Evan
403   Prodromou et. al.

404   Other references that may be useful:

405   •   Oasis Open Security Services Technical Committee, http://www.oasis-
406   open.org/committees/security/index.shtml.

407   •   Unified Modeling Language (UML), http://www.omg.org/uml/.

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

Document History

409

- 25 Jan 2001 -- First draft derived from merge of S2ML and AuthXML specs.

410

- 9 Feb 2001 -- Second draft.

411

  - Incorporated comments from Use Case subcommittee of Oasis Security Services TC.

412
413

  - Added set of high-level use cases.

414

  - Changed diagrams of detailed use case scenarios to use sequence diagrams instead of use case diagrams.

415
416

  - Added description of each use-case scenario and list of requirements flowing from the scenario.

417
418

  - Added draft glossary (as link).

419

  - Added issues list (as link).

420

  - Gave requirements labelled names for easier reference.

421

  - Incorporated and merged requirements list from Core Assertions subcommittee of Oasis Security Services TC (by Philip Hallam-Baker).

422
423

  - Corrected various editorial mistakes.

424

- 26 Feb 2001 -- Third draft.

425

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

426

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

427
428

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

429

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

430

  - Added [R-Anonymity].

431

  - Added [R-Pseudonymity].

432

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

433

  - Added [R-AuthZDecision] and noted exchange of authz decisions, per UC-1-

434

435   09:AuthZDecisions.

436   • Edited [R-AuthN] and noted exchange of authn event data, per UC-1-
437   10:AuthNEvent.

438   • Added user session use case, per UC-3-1.

439   • 10 Apr 2001 -- Fourth draft.

440   • Changed placeholder "[OSSML]" to new, official "SAML" in diagrams.

441   • Removed non-goal for challenge-response protocol based on TC motion.

442   • Modified non-goal for policies based on TC motion.

443   • Removed non-goal for protection from third parties based on TC motion.

444   • Added new use case for user sessions per TC motion, and moved session scenario
445   there.

446   • Added [R-Logout] per [UC-3-03:Logout].

447   • Added [R-SessionTermination] per [UC-3-05:SessionTermination].

448   • Added [R-BackwardCompatibleExtensions] per [UC-10-
449   06:BackwardCompatibleExtensions].

450   • Added [R-Confidentiality] per [UC-12-01:Confidentiality].

451   • Added [R-BindingConfidentiality] per [UC-12-03:BindingConfidentiality].

452   • Added non-goal for assertion and message encryption, per [UC-12-
453   03:EncryptionMethod], and reference to XML-Encryption site.

454   • Added [R-Enveloped] per [UC-7-02:Enveloped].

455   • Added [R-Intermediaries] per [UC-8-01:Intermediaries].

456   • Added [R-Intermediaries] per [UC-8-01:Intermediaries].

457   • Added Use Case Scenario 3-3 per [UC-8-02:IntermediaryAdd].

458   • Added non-goal for atomic assertions per [UC-8-05:AtomicAssertion].