



1 **Test Plan for Liberty Alliance SAML Test Event**

2 **Test Criteria**

3 **SAML 2.0**

4 **Version 3.2.2**

5 **Editor:**

6 Kyle Meadors, Drummond Group Inc.

7 **Abstract:**

8 This document describes the test steps to achieve the Liberty Interoperable™ designation for various
9 SAML 2.0 modes and profiles.

10 **Filename:**

11 Liberty_Interoperability_SAML_Test_Plan_v3.2.2.odt



12	Contents	
13	Introduction.....	3
14	Overview of Test Plan.....	3
15	Test Plan History.....	3
16	SAML Conformance Modes.....	3
17	eGov 1.5 Profile.....	4
18	POST Binding.....	4
19	Technical Requirements.....	5
20	Metadata.....	5
21	IdP Authentication.....	5
22	Trivial Processing.....	5
23	Authentication Contexts.....	5
24	Name Identifier Formats.....	6
25	XML Signatures.....	6
26	XML Encryption.....	7
27	Attribute Profiles.....	7
28	Consensus Items.....	7
29	Test Cases.....	9
30	Overview of Test Case Description.....	9
31	Test Cases Associated with Conformance Modes.....	9
32	Test Case A: Web SSO and SLO – Redirect Binding.....	10
33	Test Case B: Web SSO – Artifact Binding and SLO – SOAP Binding.....	12
34	Test Case C – NameID Management – Redirect Binding.....	14
35	Test Case D – NameID Management – SOAP Binding.....	17
36	Test Case E – POST Binding.....	21
37	Test Case F – IdP Proxy.....	24
38	Test Case G – Name Identifier Mapping.....	27
39	Test Case H – IDP Introduction.....	29
40	Test Case I – Single Session Logout.....	31
41	Test Case J – Unsolicited <Response> and “Transient” NameID.....	33
42	Test Case K – Multiple SP Logout.....	34
43	Test Case L – Force Authentication and Passive Authentication.....	37
44	Test Case M – SAML Authentication Authority.....	39
45	Test Case N – SAML Attribute Authority.....	41
46	Test Case O – SAML Authorization Decision Authority.....	43
47	Test Case P – Error Testing.....	45
48	Test Case Q – Requested AuthnContext.....	47
49	Test Case R – User Consent.....	48
50	Test Case S – Assertion Attribute.....	49
51	Test Case T – Unspecified Format.....	50
52	References.....	51

53 Introduction

54 Overview of Test Plan

55 This document is the Liberty SAML 2.0 Test Criteria Test Plan, which contains the scope of the
56 technical requirements for Liberty certification of SAML 2.0. This document is intended to be
57 publicly viewable through the Liberty Alliance website as well as prospective test participants. The
58 document is reviewed and authored by the Technology Expert Group (TEG)

59 The contents of this document include the test cases for Liberty SAML 2.0 certification as well as
60 additional technical information relevant to testing. The test cases include different test steps, which
61 as a whole cover the requirements of the SAML profiles [SAMLProf] and SAML conformance
62 modes [SAMLConf].

63 Another document, Liberty SAML 2.0 Process Test Plan, contains the detailed testing process and
64 test administration requirements for the SAML 2.0 certification test. The Liberty SAML 2.0 Process
65 Test Plan is available only to registered test participants. While the Process Test Plan is used in
66 completing a certification event, it is not needed to understand the technical expectation for
67 completing SAML 2.0 certification.

68 Test Plan History

69 This test plan replaces SAML 2.0 Interoperability Testing Procedure (vs. 3.1) test plan
70 [SAMLTP31]. The major changes to this version are modifications to the eGov profile and removing
71 the ECP Conformance mode testing requirements. Also, consensus items reached from the last
72 interoperability test event have been included here.

73 SAML 2.0 Interoperability Testing Procedure, vs. 3.1 (07/15/2008)

74 SAML 2.0 Interoperability Testing Procedure, vs. 3.0.J (11/20/2007)

75 SAML 2.0 Interoperability Testing Procedure, vs. 2.0 (07/07/2006)

76 SAML 2.0 Interoperability Testing Procedure, vs. 1.0 (2005)

77 SAML Conformance Modes

78 This test plan document contains test cases that cover the many of the operational conformance
79 modes of SAML 2.0 and the specific features that are required or optional for each mode. The details
80 of each mode are provided in [SAMLConf], and the conformance modes a listed here:

81 IdP – Identity Provider

82 IdP Lite – Identity Provider Lite

83 SP – Service Provider

84 SP Lite – Service Provider Lite

85 IdP Extended – Identify Provider Extended

86 SP Extended – Service Provider Extended

87 SAML Attribute Authority (Requester/Responder)

88 SAML Authorization Decision Authority (Requester/Responder)

89 SAML Authentication Authority (Requester/Responder)

90 Each conformance mode requires different test cases, but some test cases cover multiple
91 conformance modes. The required test cases for each conformance mode are noted in the Test Case
92 section of this document.

93 Certification in conformance modes IdP Extended and SP Extended can only be given if a
94 participant has met the certification requirements of IdP mod and SP mode, respectively.

95 **eGov 1.5 Profile**

96 The eGov 1.5 Profile is a conformance profile developed by Liberty eGovernment SIG . The test
97 cases within this test plan to achieve eGov certification are based on the requirements stated in the
98 eGov 1.5 profile. The eGov 1.5 profile and other associated documents should be consulted for
99 further explanation of the eGov requirements.

100 http://www.projectliberty.org/liberty/strategic_initiatives/egovernment

101 **POST Binding**

102 Although the POST binding is not included in the SAML SCR, it is permitted with the SAML
103 specification and has some user deployment. POST Binding is an optional Liberty designation
104 conformance mode. It involves use of POST binding for AuthnRequest, Name ID Management and
105 SLO. Certification in the POST Binding mode is done through successfully completing this [Test](#)
106 [Case E – POST Binding](#).

107 Technical Requirements

108 Metadata

109 There are no normative requirements in [SAMLConf] regarding the content or processing of
110 metadata as described in [SAMLMeta]. However, for purposes of this certification event,
111 implementations are required to:

112 Furnish correct metadata, and

113 Process metadata furnished by other testing partners

114 While metadata is not specified for SAML Attribute Requesters, interoperability with SAML
115 Authorities is very difficult without it, and for this certification event it is required that SAML
116 Attribute Requesters provide metadata as described in the draft metadata extension specification
117 [SAMLMetaExt].

118 IdP Authentication

119 SAML does not normatively specify any requirements for user authentication at IdP for Web SSO.
120 In fact, user authentication is explicitly described as “out of scope” [SAMLProf]. However, for
121 purposes of interoperability testing, it is required that IdP implementations offer at least one of these
122 authentication methods:

123 1. HTTP Basic Auth

124 2. HTTP Form Post

125 3. HTTP Get

126 Similarly, it is required that user agents be able to authenticate using at least one of these methods.

127 Trivial Processing

128 Several features specified by SAML (e.g., IdP Proxy) can be implemented such that any request
129 simply returns an error response. While this trivial behavior is, strictly speaking, in conformance
130 with the specifications, it is not meaningful in the context of interoperability testing. Except where
131 explicitly indicated (e.g., for certain Name Identifier formats) all testing steps will require non-trivial
132 responses in order to be deemed successful.

133 Authentication Contexts

134 Some of the SAML Modes rely on a well-defined ordering of authentication contexts. The SAML
135 specifications do not normatively specify an ordering [SAMLAuthnCxt] and leave the comparison
136 decisions up to the implementation [SAMLCore]. However, for purposes of testing we will
137 arbitrarily define an ordering of authentication contexts to be used in the tests. This arbitrary listing
138 of authentication class URIs, in order of increasing strength, is:

139 1. any defined authentication context not listed below

140 2. urn:oasis:names:tc:SAML:2.0:ac:classes:PreviousSession

141 3. urn:oasis:names:tc:SAML:2.0:ac:classes:InternetProtocol

142 4. urn:oasis:names:tc:SAML:2.0:ac:classes>Password

143 This ordering should be observed by all implementations testing SAML modes where authentication
144 contexts must be compared. The overall concept of the testing of the Authentication Authority is to
145 create several different assertions using different authentication contexts. Then these are queried
146 using the query terms (“exact”, “better”, “maximum”, “minimum”) and a reference authentication
147 context.

148 NOTE: Complete implementation of these authentication contexts is not required. These
149 authentication context URIs should simply be asserted in requests and responses to demonstrate
150 interoperability of authentication context processing rules.

151 **Name Identifier Formats**

152 The following Name Identifier Formats are defined by [SAMLCore]:

- 153 1. Unspecified
- 154 2. Email
- 155 3. X.509 Subject
- 156 4. Windows
- 157 5. Kerberos
- 158 6. Entity
- 159 7. Persistent
- 160 8. Transient

161 Every implementation is required to accept messages containing any of these formats, but
162 [SAMLCore] only requires that the last two be processed.

163 **XML Signatures**

164 The [SAMLConf] does not specifically indicate where XML Signatures are required, but the
165 underlying specifications in [SAMLProf] make signing required for certain profiles. Specifically,
166 these are:

- 167 1. Web SSO: The assertion element(s) in the <Response> MUST be signed for the HTTP POST
168 binding.
- 169 2. Single Logout: The <LogoutRequest> and <LogoutResponse> MUST be signed for the
170 HTTP redirect binding.
- 171 3. Name Identifier Management: The <ManageNameIDRequest> and
172 <ManageNameIDResponse> MUST be signed for the HTTP redirect binding.

173 Note that when a test step refers to a “signed SAML Response message” this implies the assertion
174 element itself is signed per the requirements in [SAMLProf].

175 SP and IdP implementations may indicate via metadata a desire for requests or responses to be
176 signed for other bindings than those indicated above. While such stipulations in metadata may not be
177 binding, participants are strongly encouraged to adhere to these requests and may be required to do
178 so to insure interoperability.

179 XML Encryption

180 [SAMLConf] stipulates several different encryption algorithms and key transport mechanisms that
181 MUST be implemented. However, these testing procedures do not require demonstration of support
182 for all these combinations and instead rely on successful interoperability as a measure of
183 conformance. Implementations should take care to ensure that elements to be encrypted include any
184 XML namespace prefix declarations so that, when decrypted, the element will remain valid
185 independent of context. One method for achieving this is described in [ExcXMLCan], but other
186 approaches will work.

187 Note that while the <ds:KeyInfo> and <xenc:EncryptedKey> elements are not required in the SAML
188 specifications or related schemas, these elements MUST be included in messages for interoperability
189 testing. There is no normative mechanism for exchanging these keys out-of-band. The precise
190 location of these elements in the message is underspecified; the most common practice among
191 interoperable SAML implementations is that in each encrypted element there be one
192 <xenc:EncryptedKey> element in parallel with the <xenc:EncryptedData>, and that this
193 <xenc:EncryptedKey> be inferred as the relevant key information for decryption without relying on
194 any references within the subelements. An erratum has been created to clarify this; see PE43 in
195 [SAMLErrata]. For this certification event, this most common practice stated above SHOULD be
196 done.

197 Finally, encryption coupled with deflation and URL encoding may create URLs that exceed the
198 maximum length supported by some browsers. Consequently, encryption is contraindicated for the
199 MNI HTTP-Redirect testing steps.

200 Attribute Profiles

201 [SAMLConf] makes no normative statements about which Attribute Profiles in [SAMLProf] are
202 required to be supported by SAML Attribute Authority or a SAML Requestor. These are the profiles
203 described in [SAMLProf] except for X.500/LDAP, which is described in [SAMLLDAP]:

- 204 1. Basic
- 205 2. X.500/LDAP
- 206 3. UUID
- 207 4. DCE PAC
- 208 5. XACML

209 Of these, this document only describes testing procedures for the Basic profile, and does not describe
210 any testing procedures regarding the other profiles.

211 Consensus Items

212 Consensus Items contains standards/implementation issues the product test group reached consensus
213 on in previous Liberty test events in order to achieve interoperability among those product test
214 groups. In order to maintain interoperability with previously tested versions, the consensus items
215 will be observed in this test event.

216 In an authentication request message, an interoperable implementation must accept a
217 requested authentication context listed in the <RequestedAuthnContext> element if it can

-
- 218 meet the authentication context requirements of the specified element and not require that
219 such information be specified out-of-band.
- 220 DSAwithSHA1 signature algorithm not supported. Section 4.1 of [SAMLConf] states that
221 the DSAwithSHA1 signature algorithm, while recommended, is not required by SAML 2.0.
222 Participants are only to use digital certificates with the required RSAwithSHA1 signature
223 algorithm.
- 224 Ignore EncryptionMethod elements in metadata. There is some confusion of interpretation
225 implementation of the EncryptionMethod metadata elements described in Section 2.4.1.1 of
226 [SAMLMeta]. After confirming with OASIS SSTC, EncryptionMethod is to be ignored.
- 227 Encryption with NameIDPolicy and ID Encryption. A question had arisen on interpreting
228 NameIDPolicy from [SAMLCore] in lines 2136-2142. It was decided that if NameIDPolicy
229 of AuthnRequest says ID is to be encrypted, it must be encrypted in the assertion and if
230 NameIDPolicy of AuthnRequest does not state the ID is to be encrypted, the IDP MAY still
231 encrypt the ID based on its policy, specifically its policy with the SP.
- 232 SSL Server-side Authentication Only for SOAP connections. To insure all participants used
233 the same security settings, it was agreed to only use SSL server-side authentication for SOAP
234 connections and not to use SSL client-side authentication.

235 Test Cases

236 Overview of Test Case Description

237 Each test case is setup with the first part listing an overview of the test steps in the test case. The
 238 second part describes the details of the individual test steps to carry out the test case. The test step
 239 overview lists the sequence of test steps along with a general description of the message or action or
 240 configuration setting required. The test step details provide more information on the expected test
 241 steps.

242 Test Cases Associated with Conformance Modes

243 In order to achieve certification in one or more of the Liberty SAML Conformance Modes, the
 244 associated test cases must be completed with all test participants with aligning modes. For example,
 245 a product testing for an IdP conformance mode must complete Test Cases A, B, C, D, H, I, J, K, L
 246 and P against all products testing for a SP conformance mode and SP Lite conformance mode. The
 247 specific pairing among participants will be given at the beginning of the certification event. A
 248 conformance mode may not require completion of all the test steps in the associated test cases. The
 249 individual test cases provide details of test steps that may or must be omitted depending on the
 250 conformance mode.

Conformance Mode	Test Cases
IdP	A, B, C, D, H, I, J, K, L, P
IdP Extended	F, G
IdP Lite	A, B, H, I, J, K, L, P
SP	A, B, C, D, H, I, J, K, L, P
SP Extended	F, G
SP Lite	A, B, H, I, J, K, L, P
POST	E, P
SAML Attribute Authority (Requester/Responder)	N
SAML Authorization Decision Authority (Requester/Responder)	O
SAML Authentication Authority (Requester/Responder)	M
eGov 1.5 profile	A, B, H, I, J, K, L, P, Q, R, S, T

251 **Test Case A: Web SSO and SLO – Redirect Binding**

252 **Preconditions:**

253 **Metadata exchanged and loaded**

254 **Encryption disabled**

255 **User Identities Not Federated**

256 **Conformance Modes: IdP, SP, IdP Lite, SP Lite, eGov**

257 **Step 1: AuthnRequest, Redirect Binding, Federate**

258 Description: User/SP does Single Sign-On with Persistent Name Identifier to Federate with
259 AllowCreate is set to TRUE. SP communication to the IdP for the SAML Authentication Request is
260 through HTTP Redirect binding.

261 IdP CONFIRM: SP successfully communicated SAML Authentication Request through
262 HTTP Redirect binding.

263 IdP CONFIRM: Name ID format is 'persistent'.

264 **Step 2: Assertion Response, POST binding**

265 Description: User provides assigned credentials for authentication. IdP provides assertion of User
266 and IdP returns a signed SAML Response message through HTTP POST binding.

267 SP CONFIRM: IdP returns signed SAML Response through HTTP POST binding.

268 SP CONFIRM: Valid assertion is returned from IdP.

269 SP CONFIRM: User identity has been federated with IdP.

270 IdP CONFIRM: User identity has been federated with SP.

271 **Step 3: SLO Request, IdP-Initiated, Redirect Binding**

272 Description: IdP logs out User session. IdP sends a signed LogoutRequest message to SP using
273 HTTP Redirect binding. SP logs out User session. SP returns a signed LogoutResponse message to
274 IdP using HTTP Redirect binding.

275 SP CONFIRM: Receives signed LogoutRequest through HTTP Redirect binding.

276 SP CONFIRM: User logged out at SP.

277 IdP CONFIRM: Receives signed LogoutResponse through HTTP Redirect binding.

278 IdP CONFIRM: User logged out at IdP.

279 **Step 4: AuthnRequest, Redirect Binding, Already Federated**

280 Description: User/SP does Single Sign-On with Persistent Name Identifier to Federate with
281 AllowCreate is set to FALSE. SP communication to the IdP for the SAML Authentication Request is
282 through HTTP Redirect binding.

283 IdP CONFIRM: SP successfully communicated SAML Authentication Request through
284 HTTP Redirect binding.

285 IdP CONFIRM: Name ID format is 'persistent'.

286 **Step 5: Assertion Response, POST binding**

287 Description: User provides assigned credentials for authentication. IdP provides assertion of User
288 and IdP returns a signed SAML Response message through HTTP POST binding.

289 SP CONFIRM: IdP returns signed SAML Response through HTTP POST binding.

290 SP CONFIRM: Valid assertion is returned from IdP.

291 SP CONFIRM: User identity has been federated with IdP.

292 IdP CONFIRM: User identity has been federated with SP.

293 **Step 6: SLO Request, SP-Initiated, Redirect Binding**

294 Description: SP logs out User session. SP sends a signed LogoutRequest message to IdP using HTTP
295 Redirect binding. IdP logs out User session. IdP returns a signed LogoutResponse message to SP
296 using HTTP Redirect binding.

297 SP CONFIRM: User logged out at SP.

298 IdP CONFIRM: Receives signed LogoutRequest through HTTP Redirect binding.

299 IdP CONFIRM: User logged out at IdP.

300 SP CONFIRM: Receives signed on LogoutResponse through HTTP Redirect binding.

301 **Test Case B: Web SSO – Artifact Binding and SLO – SOAP Binding**

302 **Preconditions:**

303 **Metadata exchanged and loaded**

304 **Encryption enabled for Assertions**

305 **Encryption enabled for NameIDs in SLO messages**

306 **User Identities Not Federated**

307 **NOTE: The SAML Conformance specification states that SOAP Binding for SLO is**
308 **optional for SP Lite and IdP Lite applications. SP Lite and IdP Lite participants may**
309 **choose to use Redirect Binding for test steps performing SLO actions instead of SOAP**
310 **Binding.**

311 **Conformance Modes: IdP, SP, IdP Lite, SP Lite, eGov**

312 **Step 1: AuthnRequest, Redirect Binding, Federate**

313 Description: User/SP does Single Sign-On with Persistent Name Identifier to Federate with
314 AllowCreate is set to TRUE. SP communication to the IdP for the SAML Authentication Request is
315 through HTTP Redirect binding.

316 IdP CONFIRM: SP successfully communicated SAML Authentication Request through
317 HTTP Redirect binding.

318 IdP CONFIRM: Name ID format is 'persistent'.

319 **Step 2: Assertion Response, HTTP Artifact**

320 Description: User provides assigned credentials for authentication. IdP creates assertion of User.
321 <Response> message is associated with an artifact. IdP returns artifact in a through HTTP Redirect
322 binding.

323 SP CONFIRM: Artifact is sent by IdP.

324 IdP CONFIRM: User identity has been federated with SP.

325 **Step 3: Artifact Resolution, SOAP Binding**

326 Description: SP sends ArtifactResolve message to IdP referencing artifact through synchronous
327 SOAP binding. IdP confirms artifact and returns <Response> message to SP in ArtifactResponse
328 message.

329 SP CONFIRM: Receives ArtifactResponse message containing <Response> message with
330 signed assertion of User.

331 SP CONFIRM: User identity has been federated with IdP.

332 IdP CONFIRM: Receives ArtifactResolve message.

333 **Step 4: SLO Request, IdP-Initiated, SOAP Binding**

334 Description: IdP logs out User session. IdP sends a signed LogoutRequest message to SP using
335 synchronous SOAP binding. SP logs out User session. SP returns a signed LogoutResponse message
336 to IdP using synchronous SOAP binding.

337 IdP CONFIRM: User logged out at IdP.

338 SP CONFIRM: Receives signed LogoutRequest through SOAP binding.

339 SP CONFIRM: User logged out at SP.

340 IdP CONFIRM: Receives signed LogoutResponse through SOAP binding.

341 **Step 5: Redirect Binding, Already Federated**

342 Description: User/SP does Single Sign-On with Persistent Name Identifier to Federate with
343 AllowCreate is set to FALSE. SP communication to the IdP for the SAML Authentication Request is
344 through HTTP Redirect binding.

345 IdP CONFIRM: SP successfully communicated SAML Authentication Request through
346 HTTP Redirect binding.

347 IdP CONFIRM: Name ID format is 'persistent'.

348 **Step 6: Assertion Response, HTTP Artifact**

349 Description: User provides assigned credentials for authentication. IdP creates assertion of User.
350 <Response> message is associated with an artifact. IdP returns artifact in a through HTTP Redirect
351 binding.

352 SP CONFIRM: Artifact is sent by IdP.

353 IdP CONFIRM: User identity has been federated with SP.

354 **Step 7: Artifact Resolution, SOAP Binding**

355 Description: SP sends ArtifactResolve message to IdP referencing artifact through synchronous
356 SOAP binding. IdP confirms artifact and returns <Response> message to SP in ArtifactResponse
357 message.

358 SP CONFIRM: Receives ArtifactResponse message containing <Response> message with
359 signed assertion of User.

360 SP CONFIRM: User identity has been federated with IdP.

361 IdP CONFIRM: Receives ArtifactResolve message.

362 **Step 8: SLO Request, SP-Initiated, SOAP Binding**

363 Description: SP logs out User session. SP sends a signed LogoutRequest message to IdP using
364 synchronous SOAP binding. IdP logs out User session. IdP returns a signed LogoutResponse
365 message to SP using synchronous SOAP binding.

366 SP CONFIRM: User logged out at SP.

367 IdP CONFIRM: Receives signed LogoutRequest through SOAP binding.

368 IdP CONFIRM: User logged out at IdP.

369 SP CONFIRM: Receives signed on LogoutResponse through SOAP binding.

370 **Test Case C – NameID Management – Redirect Binding**

371 **Preconditions:**

372 **Metadata exchanged and loaded**

373 **Encryption disabled**

374 **User Identities Not Federated**

375 **Conformance Modes: IdP, SP**

376 **Step 1: AuthnRequest, Redirect Binding, Federate**

377 Description: User/SP does Single Sign-On with Persistent Name Identifier to Federate with
378 AllowCreate is set to TRUE. SP communication to the IdP for the SAML Authentication Request is
379 through HTTP Redirect binding.

380 IdP CONFIRM: SP successfully communicated SAML Authentication Request through
381 HTTP Redirect binding.

382 IdP CONFIRM: Name ID format is 'persistent'.

383 **Step 2: Assertion Response, POST binding**

384 Description: User provides assigned credentials for authentication. IdP provides assertion of User
385 and IdP returns a SAML Response message through HTTP POST binding.

386 SP CONFIRM: IdP returns SAML Response through HTTP POST binding.

387 SP CONFIRM: Signed assertion is returned from IdP.

388 SP CONFIRM: User identity has been federated with IdP.

389 IdP CONFIRM: User identity has been federated with SP.

390 **Step 3: MNI Request, IdP-Initiated, Redirect binding**

391 Description: IdP sends signed ManageNameIdRequest message requesting to use a new NameID
392 (value chosen by the IdP at time of test execution) for the User to the SP using HTTP Redirect
393 binding. SP accepts the new NameID for the User. SP returns signed ManageNameIdResponse
394 message using HTTP Redirect binding.

395 SP CONFIRM: Receives signed ManageNameIdRequest on HTTP Redirect binding.

396 SP CONFIRM: New NameID is accepted.

397 IdP CONFIRM: Receives signed ManageNameIdResponse on HTTP Redirect binding.

398 **Step 4: SLO Request, SP-Initiated, Redirect Binding**

399 Description: SP logs out User session. SP sends a signed LogoutRequest message to IdP using HTTP
400 Redirect binding. IdP logs out User session. IdP returns a signed LogoutResponse message to SP
401 using HTTP Redirect binding.

402 SP CONFIRM: User logged out at SP.

403 IdP CONFIRM: Receives signed LogoutRequest through HTTP Redirect binding.

404 IdP CONFIRM: New NameID from Step 3 is used in LogoutRequest.

405 IdP CONFIRM: User logged out at IdP.

406 SP CONFIRM: Receives signed on LogoutResponse through HTTP Redirect binding.

407 **Step 5: AuthnRequest, Redirect Binding, Already Federated**

408 Description: User/SP does Single Sign-On with Persistent Name Identifier to Federate with
409 AllowCreate is set to FALSE. SP communication to the IdP for the SAML Authentication Request is
410 through HTTP Redirect binding.

411 IdP CONFIRM: SP successfully communicated SAML Authentication Request through
412 HTTP Redirect binding.

413 IdP CONFIRM: Name ID format is 'persistent'.

414 **Step 6: Assertion Response, POST binding**

415 Description: User provides assigned credentials for authentication. IdP provides assertion of User
416 and IdP returns a signed SAML Response message through HTTP POST binding.

417 SP CONFIRM: IdP returns signed SAML Response through HTTP POST binding.

418 SP CONFIRM: Valid assertion is returned from IdP.

419 SP CONFIRM: User identity has been federated with IdP.

420 IdP CONFIRM: User identity has been federated with SP.

421 **Step 7: MNI Request, SP-Initiated, Redirect binding**

422 Description: SP sends signed ManageNameIdRequest message requesting to use a new NameID
423 (value chosen by the SP at time of test execution) for the User to the IdP using HTTP Redirect
424 binding. IdP accepts the new NameID for the User. IdP returns signed ManageNameIdResponse
425 message using HTTP Redirect binding.

426 IdP CONFIRM: Receives signed ManageNameIdRequest on HTTP Redirect binding.

427 IdP CONFIRM: New NameID is accepted.

428 SP CONFIRM: Receives signed ManageNameIdResponse on HTTP Redirect binding.

429 **Step 8: SLO Request, IdP-Initiated, Redirect Binding**

430 Description: IdP logs out User session. IdP sends a signed LogoutRequest message to SP using
431 HTTP Redirect binding. SP logs out User session. SP returns a signed LogoutResponse message to
432 IdP using HTTP Redirect binding.

433 IdP CONFIRM: User logged out at IdP.

434 SP CONFIRM: Receives signed LogoutRequest through HTTP Redirect binding.

435 SP CONFIRM: New NameID from Step 7 is used in LogoutRequest.

436 SP CONFIRM: User logged out at SP.

437 IdP CONFIRM: Receives signed LogoutResponse through HTTP Redirect binding.

438 **Step 9: AuthnRequest, Redirect Binding, Already Federated**

439 Description: User/SP does Single Sign-On with Persistent Name Identifier to Federate with
440 AllowCreate is set to FALSE. SP communication to the IdP for the SAML Authentication Request is
441 through HTTP Redirect binding.

442 IdP CONFIRM: SP successfully communicated SAML Authentication Request through
443 HTTP Redirect binding.

444 IdP CONFIRM: Name ID format is 'persistent'.

445 **Step 10: Assertion Response, POST binding**

446 Description: User provides assigned credentials for authentication. IdP provides assertion of User
447 and IdP returns a signed SAML Response message through HTTP POST binding.

448 SP CONFIRM: IdP returns signed SAML Response through HTTP POST binding.

449 SP CONFIRM: Valid assertion is returned from IdP.

450 SP CONFIRM: User identity has been federated with IdP.

451 IdP CONFIRM: User identity has been federated with SP.

452 **Step 11: MNI-Terminate from SP**

453 Description: SP sends signed ManageNameIdRequest message with the <Terminate> element to the
454 IdP using HTTP Redirect binding. Federation for User is terminated. IdP returns signed
455 ManageNameIdResponse message using HTTP Redirect binding.

456 IdP CONFIRM: Receives signed ManageNameIdRequest with <Terminate> element on
457 HTTP Redirect binding.

458 IdP CONFIRM: Federation of User is terminated.

459 SP CONFIRM: Receives signed ManageNameIdResponse on HTTP Redirect binding.

460 SP CONFIRM: Federation of User is terminated.

461 **Test Case D – NameID Management – SOAP Binding**

462 **Preconditions:**

463 **Metadata exchanged and loaded**

464 **Encryption enabled for Assertions**

465 **Encryption enabled for NameIDs in MNI messages**

466 **Encryption enabled for NameIDs in SLO messages**

467 **User Identities Not Federated**

468 **NOTE: The SAML Conformance specification states that SOAP Binding for MNI is**
469 **optional for SP applications. SP participants may choose to use Redirect Binding for**
470 **test steps performing MNI actions instead of SOAP Binding.**

471 **Conformance Modes: IdP, SP**

472 **Step 1: AuthnRequest, Redirect Binding, Federate**

473 Description: User/SP does Single Sign-On with Persistent Name Identifier to Federate with
474 AllowCreate is set to TRUE. SP communication to the IdP for the SAML Authentication Request is
475 through HTTP Redirect binding.

476 IdP CONFIRM: SP successfully communicated SAML Authentication Request through
477 HTTP Redirect binding.

478 IdP CONFIRM: Name ID format is 'persistent'.

479 **Step 2: Assertion Response, HTTP Artifact**

480 Description: User provides assigned credentials for authentication. IdP creates assertion of User.
481 <Response> message is associated with an artifact. IdP returns artifact in a through HTTP Redirect
482 binding. SP sends ArtifactResolve message to IdP referencing artifact through synchronous SOAP
483 binding. IdP confirms artifact and returns <Response> message to SP in ArtifactResponse message.

484 SP CONFIRM: Artifact is sent by IdP.

485 IdP CONFIRM: User identity has been federated with SP.

486 **Step 3: Artifact Resolution, SOAP Binding**

487 Description:

488 SP CONFIRM: Receives ArtifactResponse message containing <Response> message with
489 signed assertion of User.

490 SP CONFIRM: User identity has been federated with IdP.

491 IdP CONFIRM: Receives ArtifactResolve message.

492 **Step 4: MNI Request, SP-Initiated, SOAP binding**

493 Description: SP sends signed ManageNameIdRequest message requesting to use a new NameID
494 (value chosen by the SP at time of test execution) for the User to the IdP using SOAP binding. IdP
495 accepts the new NameID for the User. IdP returns signed ManageNameIdResponse message using
496 same synchronous SOAP binding.

497 IdP CONFIRM: Receives signed ManageNameIdRequest on SOAP binding.

498 IdP CONFIRM: New NameID is accepted.

499 SP CONFIRM: Receives signed ManageNameIdResponse on SOAP binding.

500 **Step 5: SLO Request, IdP-Initiated, SOAP Binding**

501 Description: IdP logs out User session. IdP sends a signed LogoutRequest message to SP using
502 synchronous SOAP binding. SP logs out User session. SP returns a signed LogoutResponse message
503 to IdP using synchronous SOAP binding.

504 IdP CONFIRM: User logged out at IdP.

505 SP CONFIRM: Receives signed LogoutRequest through SOAP binding.

506 SP CONFIRM: User logged out at SP.

507 IdP CONFIRM: Receives signed LogoutResponse through SOAP binding.

508 **Step 6: Redirect Binding, Already Federated**

509 Description: User/SP does Single Sign-On with Persistent Name Identifier to Federate with
510 AllowCreate is set to FALSE. SP communication to the IdP for the SAML Authentication Request is
511 through HTTP Redirect binding.

512 IdP CONFIRM: SP successfully communicated SAML Authentication Request through
513 HTTP Redirect binding.

514 IdP CONFIRM: Name ID format is 'persistent'.

515 **Step 7: Assertion Response, HTTP Artifact**

516 Description: User provides assigned credentials for authentication. IdP creates assertion of User.
517 <Response> message is associated with an artifact. IdP returns artifact in a through HTTP Redirect
518 binding.

519 SP CONFIRM: Artifact is sent by IdP.

520 IdP CONFIRM: User identity has been federated with SP.

521 **Step 8: Artifact Resolution, SOAP Binding**

522 Description: SP sends ArtifactResolve message to IdP referencing artifact through synchronous
523 SOAP binding. IdP confirms artifact and returns <Response> message to SP in ArtifactResponse
524 message.

525 SP CONFIRM: Receives ArtifactResponse message containing <Response> message with
526 signed assertion of User.

527 SP CONFIRM: User identity has been federated with IdP.

528 IdP CONFIRM: Receives ArtifactResolve message.

529 **Step 9: MNI Request, IdP-Initiated, SOAP binding**

530 Description: IdP sends signed ManageNameIdRequest message requesting to use a new NameID
531 (value chosen by the IdP at time of test execution) for the User to the SP using SOAP binding. SP
532 accepts the new NameID for the User. SP returns signed ManageNameIdResponse message using
533 same synchronous SOAP binding.

534 SP CONFIRM: Receives signed ManageNameIdRequest on HTTP Redirect binding.

535 SP CONFIRM: New NameID is accepted.

536 IdP CONFIRM: Receives signed ManageNameIdResponse on HTTP Redirect binding.

537 **Step 10: SLO Request, SP-Initiated, SOAP Binding**

538 Description: SP logs out User session. SP sends a signed LogoutRequest message to IdP using
539 synchronous SOAP binding. IdP logs out User session. IdP returns a signed LogoutResponse
540 message to SP using synchronous SOAP binding.

541 SP CONFIRM: User logged out at SP.

542 IdP CONFIRM: Receives signed LogoutRequest through SOAP binding.

543 IdP CONFIRM: User logged out at IdP.

544 SP CONFIRM: Receives signed on LogoutResponse through SOAP binding.

545 **Step 11: Redirect Binding, Already Federated**

546 Description: User/SP does Single Sign-On with Persistent Name Identifier to Federate with
547 AllowCreate is set to FALSE. SP communication to the IdP for the SAML Authentication Request is
548 through HTTP Redirect binding.

549 IdP CONFIRM: SP successfully communicated SAML Authentication Request through
550 HTTP Redirect binding.

551 IdP CONFIRM: Name ID format is 'persistent'.

552 **Step 12: Assertion Response, HTTP Artifact**

553 Description: User provides assigned credentials for authentication. IdP creates assertion of User.
554 <Response> message is associated with an artifact. IdP returns artifact in a through HTTP Redirect
555 binding.

556 SP CONFIRM: Artifact is sent by IdP.

557 IdP CONFIRM: User identity has been federated with SP.

558 **Step 13: Artifact Resolution, SOAP Binding**

559 Description: SP sends ArtifactResolve message to IdP referencing artifact through synchronous
560 SOAP binding. IdP confirms artifact and returns <Response> message to SP in ArtifactResponse
561 message.

562 SP CONFIRM: Receives ArtifactResponse message containing <Response> message with
563 signed assertion of User.

564 SP CONFIRM: User identity has been federated with IdP.

565 IdP CONFIRM: Receives ArtifactResolve message.

566 **Step 14: MNI-Terminate, IdP-Initiated**

567 Description: IdP sends signed ManageNameIdRequest message with the <Terminate> element to the
568 IdP using SOAP binding. Federation for User is terminated. IdP returns signed
569 ManageNameIdResponse message using same synchronous binding.

570 SP CONFIRM: Receives signed ManageNameIdRequest with <Terminate> element on
571 SOAP binding.

572 SP CONFIRM: Federation of User is terminated.

573 IdP CONFIRM: Receives signed ManageNameIdResponse on SOAP binding.

574 IdP CONFIRM: Federation of User is terminated.

575 **Test Case E – POST Binding**

576 **Preconditions:**

577 **Metadata exchanged and loaded**

578 **Encryption disabled**

579 **User Identities Not Federated**

580 **Conformance Modes: POST Binding**

581 **Step 1: SSO, Federate, POST Binding**

582 Description: User does Single Sign-On at SP with Persistent Name Identifier and AllowCreate set to
583 TRUE. SP communication to the IdP for the SAML Authentication Request is through HTTP POST
584 binding. IdP provides assertion of User and IdP returns a signed SAML Response message through
585 HTTP POST binding.

586 IdP CONFIRM: SP successfully communicated SAML Authentication Request through
587 HTTP POST binding.

588 IdP CONFIRM: User has been federated

589 SP CONFIRM: IdP returns signed SAML Response through HTTP POST binding.

590 **Step 2: MNI Request, IdP-Initiated, POST binding**

591 Description: IdP sends signed ManageNameIdRequest message to the SP using HTTP POST
592 binding. SP returns signed ManageNameIdResponse message using HTTP POST binding.

593 SP CONFIRM: Receives signed ManageNameIdRequest on HTTP POST binding.

594 IdP CONFIRM: Receives signed ManageNameIdResponse on HTTP POST binding.

595 **Step 3: SLO Request, SP-Initiated, POST Binding**

596 Description: SP sends a signed LogoutRequest message to IdP using HTTP POST binding. IdP logs
597 out User session. IdP returns a signed LogoutResponse message.

598 IdP CONFIRM: Receives signed LogoutRequest on HTTP POST binding.

599 SP CONFIRM: Receives signed LogoutResponse on HTTP POST binding.

600 **Step 3: SSO, Already Federated, POST Binding**

601 Description: User does Single Sign-On at SP with AllowCreate set to FALSE. SP communication to
602 the IdP for the SAML Authentication Request is through HTTP POST binding. IdP provides
603 assertion of User and IdP returns a signed SAML Response message through HTTP POST binding.

604 IdP CONFIRM: SP successfully communicated SAML Authentication Request through
605 HTTP POST binding.

606 SP CONFIRM: IdP returns signed SAML Response through HTTP POST binding.

607 **Step 4: SLO Request, IdP-Initiated, POST Binding**

608 Description: IdP logs out User session. IdP sends a signed LogoutRequest message to SP using
609 HTTP POST binding. SP returns a signed LogoutResponse message.

610 IdP CONFIRM: Receives signed LogoutRequest on HTTP POST binding.

611 SP CONFIRM: Receives signed LogoutResponse on HTTP POST binding.

612 **Step 5: SSO, Already Federated, POST Binding**

613 Description: User does Single Sign-On at SP with AllowCreate set to FALSE. SP communication to
614 the IdP for the SAML Authentication Request is through HTTP POST binding. IdP provides
615 assertion of User and IdP returns a signed SAML Response message through HTTP POST binding.

616 IdP CONFIRM: SP successfully communicated SAML Authentication Request through
617 HTTP POST binding.

618 SP CONFIRM: IdP returns signed SAML Response through HTTP POST binding.

619 **Step 6: MNI-Terminate, IdP-Initiated**

620 Description: IdP sends signed ManageNameIdRequest message with the Terminate element to the
621 SP using HTTP POST binding. Federation for User is terminated. SP returns signed
622 ManageNameIdResponse message using HTTP POST binding.

623 SP CONFIRM: Receives signed ManageNameIdRequest with Terminate flag on HTTP
624 POST binding.

625 SP CONFIRM: Federation of User is terminated.

626 IdP CONFIRM: Receives signed ManageNameIdResponse on HTTP POST binding.

627 IdP CONFIRM: Federation of User is terminated.

628 **Step 7: SSO, Federate, POST Binding**

629 Description: User does Single Sign-On at SP with Persistent Name Identifier and AllowCreate set to
630 TRUE. SP communication to the IdP for the SAML Authentication Request is through HTTP POST
631 binding. IdP provides assertion of User and IdP returns a signed SAML Response message through
632 HTTP POST binding.

633 IdP CONFIRM: SP successfully communicated SAML Authentication Request through
634 HTTP POST binding.

635 IdP CONFIRM: User has been federated

636 SP CONFIRM: IdP returns signed SAML Response through HTTP POST binding.

637 **Step 8: MNI Request, SP-Initiated, POST binding**

638 Description: SP sends signed ManageNameIdRequest message to the IdP using HTTP POST
639 binding. IdP returns signed ManageNameIdResponse message using HTTP POST binding.

640 IdP CONFIRM: Receives signed ManageNameIdRequest on HTTP POST binding.

641 SP CONFIRM: Receives signed ManageNameIdResponse on HTTP POST binding.

642 **Step 9: SLO Request, IdP-Initiated, POST Binding**

643 Description: IdP sends a signed LogoutRequest message to SP using HTTP POST binding. SP logs
644 out User session. SP returns a signed LogoutResponse message.

645 SP CONFIRM: Receives signed LogoutRequest on HTTP POST binding.

646 IdP CONFIRM: Receives signed LogoutResponse on HTTP POST binding.

647 **Step 10: SSO, Already Federated, POST Binding**

648 Description: User does Single Sign-On at SP with AllowCreate set to FALSE. SP communication to
649 the IdP for the SAML Authentication Request is through HTTP POST binding. IdP provides
650 assertion of User and IdP returns a signed SAML Response message through HTTP POST binding.

651 IdP CONFIRM: SP successfully communicated SAML Authentication Request through
652 HTTP POST binding.

653 SP CONFIRM: IdP returns signed SAML Response through HTTP POST binding.

654 **Step 11: SLO Request, SP-Initiated, POST Binding**

655 Description: SP sends a signed LogoutRequest message to IdP using HTTP POST binding. IdP logs
656 out User session. IdP returns a signed LogoutResponse message.

657 IdP CONFIRM: Receives signed LogoutRequest on HTTP POST binding.

658 SP CONFIRM: Receives signed LogoutResponse on HTTP POST binding.

659 **Test Case F – IdP Proxy**

660 **Preconditions:**

661 **Metadata exchanged and loaded**

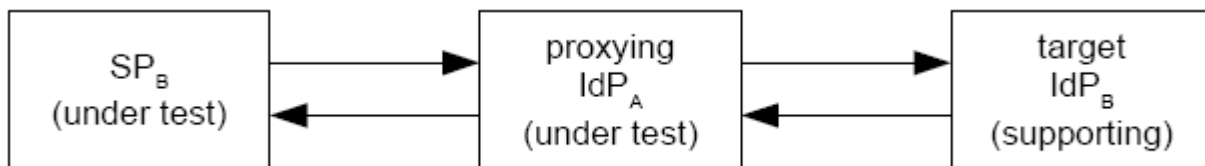
662 **Encryption disabled**

663 **User Identities Not Federated**

664 **Conformance Modes: IdP Extended, SP Extended**

665 **Background on IdP Proxy**

666 Refer to Section 3.4.1.5 of [SAMLCore] for more background. The IdP Proxy feature
667 requires two IdP implementations and one SP implementation. If we have participants A and
668 B, the following diagram depicts the roles of the test participants, assuming that IdP_A and
669 SP_B are the implementations under test:



670 To complete this Test Case, the IdP under test must receive an authentication request for a
671 User it can not authenticate but a User that the supporting IdP can authenticate. This
672 coordination of User accounts must be done prior to executing the test case.

673 **Step 1: ProxyCount=0**

674 Description: SP sets ProxyCount=0 where proxy is disallowed.

675 SP CONFIRM: SP has disallowed proxy.

676 **Step 2: AuthnRequest from SP to IdP_A, Redirect Binding, Federate**

677 Description: User/SP attempts Single Sign-On with Persistent Name Identifier to Federate with
678 AllowCreate is set to TRUE. SP communication to the IdP_A for the SAML Authentication Request is
679 through HTTP Redirect binding. IdP_A does not recognize User and thus can not authenticate user.

680 IdP_A CONFIRM: ProxyCount is set to 0.

681 IdP_A CONFIRM: User is not authenticated.

682 **Step 3: Response Failure**

683 Description: Being unable to authenticate User, IdP_A returns SAML Response with error indicating
684 AuthnRequest failed.

685 SP CONFIRM: IdP_A returns SAML Response indicating authentication error.

686 **Step 4: ProxyCount is Removed and IdP List is set**

687 Description: SP removes ProxyCount where proxy is allowed. SP configures <IdPList> to include
688 IdP_B.

689 SP CONFIRM: SP has removed ProxyCount to allow proxy.

690 SP CONFIRM: SP has set <IdPList> to include IdP_B.

691 **Step 5: AuthnRequest from SP to IdP_A, Redirect Binding, Federate**

692 Description: User/SP does Single Sign-On with Persistent Name Identifier to Federate with
693 AllowCreate is set to TRUE. SP communication to the IdP_A for the SAML Authentication Request is
694 through HTTP Redirect binding. IdP_A does not recognize User but recognizes it can proxy the
695 AuthnRequest to IdP_B.

696 IdP_A CONFIRM: ProxyCount is not set.

697 IdP_A CONFIRM: User is not authenticated.

698 IdP_A CONFIRM: AuthnRequest contains <IdPList> which includes IdP_B.

699 **Step 6: AuthnRequest from IdP_A to IdP_B, Redirect Binding, Federate**

700 Description: IdP_A proxies AuthnRequest to IdP_B through HTTP Redirect binding.

701 IdP_B CONFIRM: Receives AuthnRequest from IdP_A.

702 IdP_B CONFIRM: ProxyCount is set to 0.

703 IdP_B CONFIRM: <IdPList> includes IdP_B.

704 **Step 7: Assertion Response from IdP_B to IdP_A, POST binding**

705 Description: User provides assigned credentials to IdP_B for authentication. IdP_B provides assertion of
706 User and returns a signed SAML Response message to IdP_A through HTTP POST binding.

707 IdP_A CONFIRM: Receives SAML Response through HTTP POST binding.

708 IdP_A CONFIRM: Valid assertion is returned from IdP_B.

709 IdP_A CONFIRM: <AuthnStatement> contains <AuthenticatingAuthority> referencing IdP_B.

710 **Step 8: Assertion Response from IdP_A to SP, POST binding**

711 Description: IdP_A inserts assertion of User it received from IdP_B and returns a signed SAML
712 Response message to SP through HTTP POST binding.

713 SP CONFIRM: Receives SAML Response through HTTP POST binding.

714 SP CONFIRM: Valid assertion is returned from IdP_A.

715 SP CONFIRM: <AuthnStatement> contains <AuthenticatingAuthority> referencing IdP_B.

716 **Step 9: SLO Request, IdP-Initiated, Redirect Binding**

717 Description: IdP_A logs out User session. IdP_A sends a signed LogoutRequest message to SP using
718 HTTP Redirect binding. SP logs out User session. SP returns a signed LogoutResponse message to
719 IdP_A using HTTP Redirect binding.

720 IdP_A CONFIRM: User logged out at IdP_A.

721 SP CONFIRM: Receives signed LogoutRequest through HTTP Redirect binding.

722 SP CONFIRM: User logged out at SP.

723 IdP_A CONFIRM: Receives signed LogoutResponse through HTTP Redirect binding.

724 **Step 10: ProxyCount=1 and IdP List is set**

725 Description: SP makes ProxyCount set to 1. SP configures <IdPList> to include IdP_B.

726 SP CONFIRM: SP sets ProxyCount to 1.

727 SP CONFIRM: SP has set <IdPList> to include IdP_B.

728 **Step 11: AuthnRequest from SP to IdP_A, Redirect Binding, Federate**

729 Description: User/SP does Single Sign-On with Persistent Name Identifier to Federate with
730 AllowCreate is set to TRUE. SP communication to the IdP_A for the SAML Authentication Request is
731 through HTTP Redirect binding. IdP_A does not recognize User but recognizes it can proxy the
732 AuthnRequest to IdP_B.

733 IdP_A CONFIRM: ProxyCount is set to 1.

734 IdP_A CONFIRM: User is not authenticated.

735 IdP_A CONFIRM: AuthnRequest contains <IdPList> which includes IdP_B.

736 **Step 12: AuthnRequest from IdP_A to IdP_B, Redirect Binding, Federate**

737 Description: IdP_A proxies AuthnRequest to IdP_B through HTTP Redirect binding.

738 IdP_B CONFIRM: Receives AuthnRequest from IdP_A.

739 IdP_B CONFIRM: ProxyCount is set to 0.

740 IdP_B CONFIRM: <IdPList> includes IdP_B.

741 **Step 13: Assertion Response from IdP_B to IdP_A, POST binding**

742 Description: User provides assigned credentials to IdP_B for authentication. IdP_B provides assertion of
743 User and returns a signed SAML Response message to IdP_A through HTTP POST binding.

744 IdP_A CONFIRM: Receives SAML Response through HTTP POST binding.

745 IdP_A CONFIRM: Valid assertion is returned from IdP_B.

746 IdP_A CONFIRM: <AuthnStatement> contains <AuthenticatingAuthority> referencing IdP_B.

747 **Step 14: Assertion Response from IdP_A to SP, POST binding**

748 Description: IdP_A inserts assertion of User it received from IdP_B and returns a signed SAML
749 Response message to SP through HTTP POST binding.

750 SP CONFIRM: Receives SAML Response through HTTP POST binding.

751 SP CONFIRM: Valid assertion is returned from IdP_A.

752 SP CONFIRM: <AuthnStatement> contains <AuthenticatingAuthority> referencing IdP_B.

753 **Step 15: SLO Request, IdP-Initiated, Redirect Binding**

754 Description: IdP_A logs out User session. IdP_A sends a signed LogoutRequest message to SP using
755 HTTP Redirect binding. SP logs out User session. SP returns a signed LogoutResponse message to
756 IdP_A using HTTP Redirect binding.

757 IdP_A CONFIRM: User logged out at IdP_A.

758 SP CONFIRM: Receives signed LogoutRequest through HTTP Redirect binding.

759 SP CONFIRM: User logged out at SP.

760 IdP_A CONFIRM: Receives signed LogoutResponse through HTTP Redirect binding.

761 **Test Case G – Name Identifier Mapping**

762 **Preconditions:**

763 **Metadata exchanged and loaded**

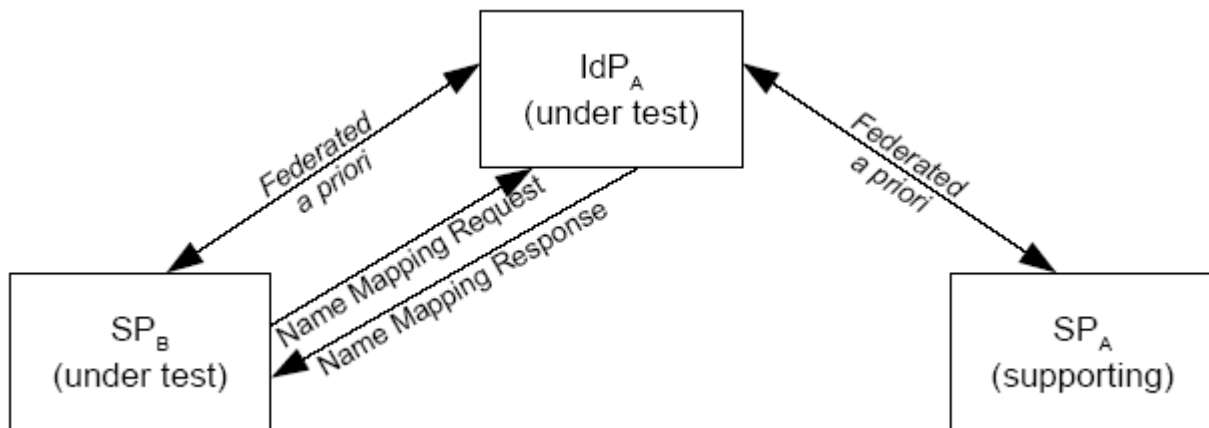
764 **Encryption disabled**

765 **User Identities Not Federated**

766 **Conformance Modes: IdP Extended, SP Extended**

767 **Background on Name Identifier Mapping Feature**

768 The name identifier mapping feature requires that an IdP provide an indirect reference for a
769 principal at SP_A in response to a request from SP_B. Assuming again that teams A and B are
770 testing IdP_A and SP_B, it is necessary for the principal to federate her identity at both SP_B and
771 SP_A with IdP_A. This can be depicted as follows:



772 **Step 1: SSO at SP_A**

773 Description: User does Single Sign-On at SP_A with Persistent Name Identifier. SP_A communicates
774 Authentication Request through HTTP Redirect binding. IdP provides assertion of User and IdP
775 returns a signed SAML Response message through HTTP POST binding.

776 IdP CONFIRM: SP_A successfully communicated SAML Authentication Request through
777 HTTP Redirect binding.

778 IdP CONFIRM: User has been federated with SP_A.

779 SP_A CONFIRM: IdP returns signed SAML Response through HTTP POST binding.

780 SP_A CONFIRM: User has been federated with IdP.

781 **Step 2: SSO at SP_B**

782 Description: User does Single Sign-On at SP_B with Persistent Name Identifier. SP_B communicates
783 Authentication Request through HTTP Redirect binding. IdP provides assertion of User and IdP
784 returns a signed SAML Response message through HTTP POST binding.

785 IdP CONFIRM: SP_B successfully communicated SAML Authentication Request through
786 HTTP Redirect binding.

787 IdP CONFIRM: User has been federated with SP_B.

788 SP_B CONFIRM: IdP returns signed SAML Response through HTTP POST binding.

789 SP_B CONFIRM: User has been federated with IdP.

790 **Step 3: NameIDMappingRequest from SP_B**

791 13. SP_B sends signed NameIDMappingRequest message over a SOAP binding to the IdP requesting
792 an alternative name identifier for User. IdP maps the request to the User name ID federated with
793 SP_A. IdP returns the encrypted name ID federated with SP_A in a signed NameIDMappingResponse
794 message using a SOAP binding.

795 IdP CONFIRM: Receives signed NameIDMappingRequest for name ID federated with SP_B.

796 SP_B CONFIRM: Receives NameIDMappingResponse for for name ID federated with SP_A.

797 SP_B CONFIRM: Receives Encrypted NameID.

798 **Test Case H – IDP Introduction**

799 **Preconditions:**

800 **Metadata exchanged and loaded**

801 **Encryption disabled**

802 **User Identities Not Federated**

803 **NOTE: The SAML Conformance specification states that IdP Discovery is optional for**
804 **SP and SP Lite applications. SP and SP Lite participants may option out of this test**
805 **case.**

806 **Conformance Modes: IdP, SP, IdP Lite, SP Lite, eGov**

807 **Background**

808 Two IdP actors are needed to execute this test case. Test administrator will provide specific
809 instructions on setup and actor roles at time of test case execution.

810 **Step 1: Clear Cookies**

811 Description: Cookies are cleared from User Browser

812 USER_CONFIRM: User has cleared cookies from browser.

813 **Step 2: IdP_A is added to CDC**

814 Description: User logs in at IdP_A. Cookie is set in common domain with IdP_A appended to list of IdPs.

815 IdP_A_CONFIRM: User logged in, cookie is set in common domain and IdP_A appended to end
816 of IdP list in cookie.

817 **Step 3: IdP_B is added to CDC**

818 Description: User logs in at IdP_B. IdP_B appended to list of IdPs in CDC.

819 IdP_B_CONFIRM: User logged in and IdP_B appended to end of IdP list in CDC.

820 **Step 4: SSO to IdP_A using CDC, HTTP Redirect**

821 Description: User/SP does Single Sign-On using a common domain cookie. SP reads cookie. For
822 eGov profile testing, SP must present to the User a list of IdPs and allow User to select IdP_A for
823 authentication. For non-eGov profile testing, depending on SP implementation, either the User is
824 presented list of IDPs and selects IdP_A for authentication or SP redirects User to IdP_A for
825 authentication. SP communication to the IdP_A for the signed authentication request is through HTTP
826 Redirect binding. IdP_A provides signed assertion of User and IdP returns a SAML Response message
827 through HTTP POST binding.

828 IdP_A_CONFIRM: SP successfully communicated signed SAML Authentication Request
829 through HTTP Redirect binding.

830 SP_CONFIRM: Cookie was read and IdP_A and IdP_B were present in CDC.

831 SP_CONFIRM: IdP_A returns signed assertion through HTTP POST binding.

832 SP_CONFIRM: For eGov profile, SP presents list of IdPs for authentication and IdP_A and
833 IdP_B must be present on list.

834 **Step 5: SLO, SP-Initiated, HTTP Redirect**

835 Description: SP does SLO. SP sends a signed LogoutRequest message to IdP_A using HTTP Redirect
836 binding. IdP_A returns a signed LogoutResponse message. User is logged out.

837 IdP_A CONFIRM: Receives signed LogoutRequest on HTTP Redirect binding.

838 SP CONFIRM: Receives signed LogoutResponse on HTTP Redirect binding.

839 SP CONFIRM: User is logged out.

840 **Step 6: CDC is removed**

841 Description: User closes browser. CDC is removed.

842 User CONFIRM: CDC is removed once browser is closed.

843 **Test Case I – Single Session Logout**

844 **Preconditions:**

- 845 **Metadata exchanged and loaded**
- 846 **Encryption disabled**
- 847 **User Identities Not Federated**

848 **Conformance Modes: IdP, SP, IdP Lite, SP Lite, eGov**

849 **Step 1: SSO creates Session A for User**

850 Description: User creates Session A through Single Sign-On with Federate where AllowCreate is set
851 to TRUE. SP communication to the IdP for the SAML Authentication Request is through HTTP
852 Redirect binding. IdP provides assertion of User and IdP returns a signed SAML Response message
853 through HTTP POST binding.

854 IdP CONFIRM: SP successfully communicated SAML Authentication Request through
855 HTTP Redirect binding.

856 IdP CONFIRM: User has been federated.

857 IdP CONFIRM: User has been logged in through Session A.

858 SP CONFIRM: IdP returns signed SAML Response through HTTP POST binding.

859 **Step 2: SSO creates Session B for User**

860 Description: User creates new Session B, generally through second browser instances, through
861 Single Sign-On. SP communication to the IdP for the SAML Authentication Request is through
862 HTTP Redirect binding. IdP provides assertion of User and IdP returns a signed SAML Response
863 message through HTTP POST binding.

864 IdP CONFIRM: SP successfully communicated SAML Authentication Request through
865 HTTP Redirect binding.

866 IdP CONFIRM: User has been logged in through Session B.

867 SP CONFIRM: IdP returns signed SAML Response through HTTP POST binding.

868 **Step 3: SLO from SP for Session A**

869 Description: User logs off of Session A at the SP. SP sends a signed LogoutRequest message to IdP
870 for Session A using HTTP Redirect binding. IdP examines <SessionIndex> and determines the logout
871 request is for Session A. User is logged out of Session A, but User remains logged in through
872 Session B. IdP returns a signed LogoutResponse message for Session A.

873 IdP CONFIRM: Receives signed LogoutRequest on HTTP Redirect binding.

874 IdP CONFIRM: User logged out of Session A.

875 IdP CONFIRM: User remains logged in through Session B.

876 SP CONFIRM: Receives signed LogoutResponse on HTTP Redirect binding.

877 SP CONFIRM: User logged out of Session A.

878 SP CONFIRM: User remains logged in through Session B.

879 **Step 4: SSO creates Session C for User**

880 Description: User creates Session C through Single Sign-On with Federate where AllowCreate is set
881 to TRUE. SP communication to the IdP for the SAML Authentication Request is through HTTP
882 Redirect binding. IdP provides assertion of User and IdP returns a signed SAML Response message
883 through HTTP POST binding.

884 IdP CONFIRM: SP successfully communicated SAML Authentication Request through
885 HTTP Redirect binding.

886 IdP CONFIRM: User has been federated.

887 IdP CONFIRM: User has been logged in through Session C.

888 SP CONFIRM: IdP returns signed SAML Response through HTTP POST binding.

889 **Step 5: SLO from IdP for Session C**

890 Description: User logs off of Session C at the IdP. IdP sends a signed LogoutRequest message to SP
891 for Session C using HTTP Redirect binding. SP examines <SessionIndex> and determines the logout
892 request is for Session C. User is logged out of Session C, but User remains logged in through
893 Session B. SP returns a signed LogoutResponse message for Session C.

894 SP CONFIRM: Receives signed LogoutRequest on HTTP Redirect binding.

895 SP CONFIRM: User logged out of Session C.

896 SP CONFIRM: User remains logged in through Session B.

897 IdP CONFIRM: Receives signed LogoutResponse on HTTP Redirect binding.

898 IdP CONFIRM: User logged out of Session C.

899 IdP CONFIRM: User remains logged in through Session B.

900 **Test Case J – Unsolicited <Response> and “Transient” NameID**

901 **Preconditions:**

902 **Metadata exchanged and loaded**

903 **Encryption disabled**

904 **User Identities Not Federated**

905 **Conformance Modes: IdP, SP, IdP Lite, SP Lite, eGov**

906 **Step 1: Unsolicited <Response>, HTTP Post Binding, “transient” NameID**

907 Description: User does Single Sign-On at IdP. IdP provides assertion of User and makes Name ID
908 format “transient”. IdP sends a signed SAML Response message through HTTP POST binding.

909 IdP CONFIRM: User has been federated.

910 SP CONFIRM: NameID format is “transient”.

911 SP CONFIRM: IdP sends signed SAML Response through HTTP POST binding.

912 **Step 2: SLO from SP**

913 Description: SP sends a signed LogoutRequest message to IdP using HTTP Redirect binding. IdP
914 logs out User session. IdP returns a signed LogoutResponse message.

915 IdP CONFIRM: Receives signed LogoutRequest on HTTP Redirect binding.

916 SP CONFIRM: Receives signed LogoutResponse on HTTP Redirect binding.

917 **Step 3: Unsolicited <Response>, Artifact Binding, “transient” NameID**

918 Description: User does Single Sign-On at IdP. IdP provides assertion of User and makes Name ID is
919 format “transient”. <Response> message is communicated through Artifact binding. The IdP and SP
920 resolve the artifact via a SOAP binding. SP consumes the <Response> message.

921 IdP CONFIRM: Artifact resolution is properly done.

922 IdP CONFIRM: User has been federated

923 SP CONFIRM: NameID format is “transient”.

924 SP CONFIRM: IdP sends signed SAML Response through HTTP Artifact.

925 SP CONFIRM: Artifact resolution is properly done.

926 **Step 4: SLO from IdP**

927 Description: IdP sends a signed LogoutRequest message to SP using HTTP Redirect binding. SP
928 logs out User session. SP returns a signed LogoutResponse message.

929 SP CONFIRM: Receives signed LogoutRequest on HTTP Redirect binding.

930 IdP CONFIRM: Receives signed LogoutResponse on HTTP Redirect binding.

931 **Test Case K – Multiple SP Logout**

932 **Preconditions:**

933 **Metadata exchanged and loaded**

934 **Encryption disabled**

935 **User Identities Not Federated**

936 **Conformance Modes: IdP, SP, IdP Lite, SP Lite, eGov**

937 **Step 1: SSO from SP_A**

938 Description: User at SP_A performs Single Sign-On (any profile) to IdP.

939 IdP CONFIRM: SP_A successfully communicated SAML Authentication Request and IdP sent
940 back Assertion for User.

941 IdP CONFIRM: User has been federated with SP_A

942 SP_A CONFIRM: IdP returns signed SAML Response and User is authenticated.

943 **Step 2: SSO from SP_B using same Session ID**

944 Description: User logs in to SP_B and is authenticated by IdP with same session id.

945 IdP CONFIRM: SP_B successfully communicated SAML Authentication Request and IdP sent
946 back Assertion for User and maintained same session id as in Step 1.

947 IdP CONFIRM: User has been federated with SP_B

948 SP_B CONFIRM: IdP returns signed SAML Response and User is authenticated.

949 **Step 3: SLO from SP_A to IdP**

950 Description: User issues SLO from SP_A to IdP.

951 IdP CONFIRM: SP_A sends signed LogoutRequest for User.

952 SP_A CONFIRM: A signed LogoutRequest is sent to IdP.

953 **Step 4: LogoutRequest from IdP to SP_B**

954 Description: Signed LogoutRequest is sent from IdP to SP_B. User is logged out of SP_B. After
955 receiving the LogoutResponse from SP_B, IdP sends LogoutResponse to SP_A.

956 IdP CONFIRM: Signed LogoutRequest is sent to SP_A and receives back signed
957 LogoutResponse.

958 IdP CONFIRM: No active session for User.

959 SP_B CONFIRM: IdP sends signed LogoutResponse, a signed LogoutResponse is returned and
960 User is logged out.

961 SP_A CONFIRM: Receives signed LogoutResponse from IdP.

962 **Step 5: SSO from SP_B to IdP**

963 Description: User at SP_B performs Single Sign-On (any profile) to IdP.

964 IdP CONFIRM: SP_B successfully communicated SAML Authentication Request and IdP sent
965 back Assertion for User.

966 IdP CONFIRM: User has active session.

967 SP_B CONFIRM: IdP returns signed SAML Response and User is authenticated.

968 **Step 6: SSO from SP_A using same Session ID**

969 Description: User logs in to SP_A and is authenticated by IdP with same session id.

970 IdP CONFIRM: SP_A successfully communicated SAML Authentication Request and IdP sent
971 back Assertion for User and maintained same session id as in Step 5.

972 SP_A CONFIRM: IdP returns signed SAML Response and User is authenticated.

973 **Step 7: SLO from SP_B to IdP**

974 Description: User does SLO from IdP to SP_B.

975 IdP CONFIRM: SP_B is sent signed LogoutRequest for User.

976 SP_B CONFIRM: IdP sends a signed LogoutRequest and User is logged out.

977 **Step 8: LogoutRequest from IdP to SP_A**

978 Description: Signed LogoutRequest is sent to SP_A from IdP. User is logged out of SP_A. After
979 receiving the LogoutResponse from SP_A, IdP sends LogoutResponse to SP_B.

980 IdP CONFIRM: Signed LogoutRequest is sent to SP_A and receives back signed
981 LogoutResponse.

982 SP_A CONFIRM: IdP sends signed LogoutResponse, a signed LogoutResponse is returned
983 and User is logged out.

984 SP CONFIRM: Receives signed LogoutResponse from IdP.

985 **Step 9: SSO from SP_B to IdP**

986 Description: User at SP_B performs Single Sign-On (any profile) to IdP.

987 IdP CONFIRM: SP_B successfully communicated SAML Authentication Request and IdP sent
988 back Assertion for User.

989 IdP CONFIRM: User has active session.

990 SP_B CONFIRM: IdP returns signed SAML Response and User is authenticated.

991 **Step 10: SSO from SP_A using same Session ID**

992 Description: User logs in to SP_A and is authenticated by IdP with same session id.

993 IdP CONFIRM: SP_A successfully communicated SAML Authentication Request and IdP sent
994 back Assertion for User and maintained same session id as in Step 5.

995 SP_A CONFIRM: IdP returns signed SAML Response and User is authenticated.

996 **Step 11: Local logout at SP_B**

997 Description: User does local logout (not SLO) at SP_B.

998 IdP CONFIRM: LogoutRequest for User is not received at this time.

999 SP_B CONFIRM: User is logged out locally.

1000 **Step 12: SLO from SP_A to IdP**

1001 Description: User issues SLO from SP_A to IdP.

1002 IdP CONFIRM: SP_A sends signed LogoutRequest for User.

1003 SP_A CONFIRM: A signed LogoutRequest is sent to IdP. User is logged out.

1004 **Step 13: PartialLogout Error**

1005 Description: Signed LogoutRequest is sent from IdP to SP_B. Because User is already logged out of
1006 SP_B, a status code of “PartialLogout” is returned in the to the Signed LogoutResponse. IdP sends
1007 LogoutResponse to SP_A.

1008 IdP CONFIRM: Signed LogoutRequest is sent to SP_B and receives back signed
1009 LogoutResponse.

1010 IdP CONFIRM: Signed LogoutResponse contains status code of
1011 urn:oasis:names:tc:SAML:2.0:status:PartialLogout.

1012 SP_B CONFIRM: IdP sends signed LogoutResponse, unable to perform SLO, and a signed
1013 LogoutResponse is returned indicating “PartialLogout”.

1014 SP_A CONFIRM: Receives signed LogoutResponse from IdP indicating “PartialLogout.”

1015 **Test Case L – Force Authentication and Passive Authentication**

1016 **Preconditions:**

1017 **Metadata exchanged and loaded**

1018 **Encryption disabled**

1019 **Conformance Modes (Required): IdP, SP, IdP Lite, SP Lite, eGov**

1020 **Step 1: User Logins at IdP**

1021 Description: User logs in at IdP and creates and active session

1022 IdP CONFIRM: User logged in.

1023 **Step 2: SP sets IsPassive=TRUE**

1024 Description: SP is configured to make IsPassive set to TRUE.

1025 SP CONFIRM: SP is configured IsPassive=TRUE.

1026 **Step 3: SSO with isPassive=TRUE**

1027 Description: User/SP does Single Sign-On SP communication to the IdP for the SAML
1028 Authentication Request is through HTTP Redirect binding. IdP provides assertion of User without
1029 interacting with the user. IdP returns a signed SAML Response message through HTTP POST
1030 binding.

1031 IdP CONFIRM: SP successfully communicated SAML Authentication Request through
1032 HTTP Redirect binding.

1033 IdP CONFIRM: User does not interact with IdP or IdP must not take control of user
1034 interface.

1035 SP CONFIRM: IdP returns assertion in signed SAML Response through HTTP POST
1036 binding.

1037 **Step 4: SLO from SP**

1038 Description: SP sends a signed LogoutRequest message to IdP using HTTP Redirect binding. IdP
1039 logs out User session. IdP returns a signed LogoutResponse message.

1040 IdP CONFIRM: Receives signed LogoutRequest on HTTP Redirect binding.

1041 SP CONFIRM: Receives signed LogoutResponse on HTTP Redirect binding.

1042 SP CONFIRM: User is logged out.

1043 **Step 5: SP sets IsPassive=FALSE**

1044 Description: SP is configured to make IsPassive set to FALSE.

1045 SP CONFIRM: SP is configured IsPassive=FALSE.

1046 **Step 6: SSO with isPassive=FALSE**

1047 Description: User/SP does Single Sign-On SP communication to the IdP for the SAML
1048 Authentication Request is through HTTP Redirect binding. IdP interacts with and authenticates the
1049 user. IdP returns a signed SAML Response message through HTTP POST binding.

1050 IdP CONFIRM: SP successfully communicated SAML Authentication Request through
1051 HTTP Redirect binding.
1052 IdP CONFIRM: User does interact with IdP.
1053 SP CONFIRM: IdP returns assertion in signed SAML Response through HTTP POST
1054 binding.

1055 **Step 7: SLO from SP**

1056 Description: SP sends a signed LogoutRequest message to IdP using HTTP Redirect binding. IdP
1057 logs out User session. IdP returns a signed LogoutResponse message.

1058 IdP CONFIRM: Receives signed LogoutRequest on HTTP Redirect binding.

1059 SP CONFIRM: Receives signed LogoutResponse on HTTP Redirect binding.

1060 SP CONFIRM: User is logged out.

1061 **Step 8: User Logins At IdP**

1062 Description: User logs in at IdP and creates and active session

1063 IdP CONFIRM: User logged in.

1064 **Step 9: SP sets ForceAuthn=TRUE**

1065 Description: SP is configured to make ForceAuthn set to TRUE.

1066 SP CONFIRM: SP is configured ForceAuthn=TRUE.

1067 **Step 10: SSO with ForceAuthn=TRUE**

1068 Description: User/SP does Single Sign-On SP communication to the IdP for the SAML
1069 Authentication Request is through HTTP Redirect binding. IdP interacts with User and authenticates
1070 the User. IdP provides assertion of User. IdP returns a signed SAML Response message through
1071 HTTP POST binding.

1072 IdP CONFIRM: SP successfully communicated SAML Authentication Request through
1073 HTTP Redirect binding.

1074 IdP CONFIRM: User interacts with IdP and is authenticated.

1075 SP CONFIRM: IdP returns assertion in signed SAML Response through HTTP POST
1076 binding.

1077 **Step 11: SLO from SP**

1078 Description: SP sends a signed LogoutRequest message to IdP using HTTP Redirect binding. IdP
1079 logs out User session. IdP returns a signed LogoutResponse message.

1080 IdP CONFIRM: Receives signed LogoutRequest on HTTP Redirect binding.

1081 SP CONFIRM: Receives signed LogoutResponse on HTTP Redirect binding.

1082 SP CONFIRM: User is logged out.

1083 **Test Case M – SAML Authentication Authority**

1084 **Preconditions:**

1085 **Metadata exchanged and loaded**

1086 **Encryption disabled**

1087 **User Identities Not Federated**

1088 **Conformance Modes: SAML Authentication Authority**

1089 **Note: Section [[AuthenticationContexts](#)] within this document describes the strength of**
1090 **the AuthnContext classes used for comparison.**

1091 **Test Steps**

1092 **Step 1:**

1093 Description: User/SP does Single Sign-On with Persistent Name Identifier. SP communication to the
1094 IdP for the SAML Authentication Request is through HTTP POST binding. IdP provides assertion of
1095 User and IdP returns a signed SAML Response message through HTTP POST binding.

1096 IdP CONFIRM: SP successfully communicated SAML Authentication Request through
1097 HTTP POST binding.

1098 IdP CONFIRM: User has been federated

1099 SP CONFIRM: IdP returns signed SAML Response through HTTP POST binding.

1100 **Step 2:**

1101 Description: SAML Requester sets AC comparison to “exact”.

1102 SAML Requester CONFIRM: AC comparison=“exact”.

1103 **Step 3:**

1104 Description: SAML Requester sends Authentication Query to SAML Responder through SOAP
1105 binding. SAML Responder returns SAML Response.

1106 SAML Responder CONFIRM: SAML Requester sent Authentication Query.

1107 SAML Requester CONFIRM: SAML Responder returned the SAML Response.

1108 **Step 4:**

1109 Description: SAML Requester sets AC comparison to “better”.

1110 SAML Requester CONFIRM: AC comparison=“better”.

1111 **Step 5:**

1112 Description: SAML Requester sends Authentication Query to SAML Responder through SOAP
1113 binding. SAML Responder returns SAML Response.

1114 SAML Responder CONFIRM: SAML Requester sent Authentication Query.

1115 SAML Requester CONFIRM: SAML Responder returned the SAML Response.

1116 **Step 6:**

1117 Description: SAML Requester sets AC comparison to “minimum”.

1118 SAML Requester CONFIRM: AC comparison="minimum".

1119 **Step 7:**

1120 Description: SAML Requester sends Authentication Query to SAML Responder through SOAP
1121 binding. SAML Responder returns SAML Response.

1122 SAML Responder CONFIRM: SAML Requester sent Authentication Query.

1123 SAML Requester CONFIRM: SAML Responder returned the SAML Response.

1124 **Step 8:**

1125 Description: SAML Requester sets AC comparison to "maximum".

1126 SAML Requester CONFIRM: AC comparison=" maximum".

1127 **Step 9:**

1128 Description: SAML Requester sends Authentication Query to SAML Responder through SOAP
1129 binding. SAML Responder returns SAML Response.

1130 SAML Responder CONFIRM: SAML Requester sent Authentication Query.

1131 SAML Requester CONFIRM: SAML Responder returned the SAML Response.

1132 **Test Case N – SAML Attribute Authority**

1133 **Preconditions:**

1134 **Metadata exchanged and loaded**

1135 **Encryption disabled**

1136 **User Identities Not Federated**

1137 **Conformance Modes: SAML Attribute Authority**

1138 **Step 1:**

1139 Description: User/SP does Single Sign-On with Persistent Name Identifier. SP communication to the
1140 IdP for the SAML Authentication Request is through HTTP POST binding. IdP provides assertion of
1141 User and IdP returns a signed SAML Response message through HTTP POST binding.

1142 IdP CONFIRM: SP successfully communicated SAML Authentication Request through
1143 HTTP POST binding.

1144 IdP CONFIRM: User has been federated

1145 SP CONFIRM: IdP returns signed SAML Response through HTTP POST binding.

1146 **Step 2:**

1147 Description: SAML Responder sets attribute query to no attributes.

1148 SAML Responder CONFIRM: Attribute Query No Attributes.

1149 **Step 3:**

1150 Description: SAML Requester sends Attribute Query to SAML Responder through SOAP binding.

1151 SAML Responder returns SAML Response.

1152 SAML Responder CONFIRM: SAML Requester sent Attribute Query.

1153 SAML Requester CONFIRM: SAML Responder returned the SAML Response.

1154 **Step 4:**

1155 Description: SAML Responder sets attribute query to attribute named.

1156 SAML Responder CONFIRM: Attribute Query Attribute Named.

1157 **Step 5:**

1158 Description: SAML Requester sends Attribute Query to SAML Responder through SOAP binding.

1159 SAML Responder returns SAML Response.

1160 SAML Responder CONFIRM: SAML Requester sent Attribute Query.

1161 SAML Requester CONFIRM: SAML Responder returned the SAML Response.

1162 **Step 6:**

1163 Description: SAML Responder sets attribute query to attribute value.

1164 SAML Responder CONFIRM: Attribute Query Attribute Value.

1165 **Step 7:**

1166 Description: SAML Requester sends Attribute Query to SAML Responder through SOAP binding.

1167 SAML Responder returns SAML Response.

-
- 1168 SAML Responder CONFIRM: SAML Requester sent Attribute Query.
1169 SAML Requester CONFIRM: SAML Responder returned the SAML Response.

1170 **Step 8:**

1171 Description: SAML Responder sets attribute query to attribute named. SAML Responder enables
1172 attribute for encryption.

1173 SAML Responder CONFIRM: Attribute Query Attribute Named.

1174 SAML Responder CONFIRM: Encryption assertion enabled.

1175 **Step 9:**

1176 Description: SAML Requester sends Attribute Query to SAML Responder through SOAP binding.

1177 SAML Responder returns SAML Response.

1178 SAML Responder CONFIRM: SAML Requester sent Attribute Query.

1179 SAML Requester CONFIRM: SAML Responder returned the SAML Response.

1180 **Test Case O – SAML Authorization Decision Authority**

1181 **Preconditions:**

1182 **Metadata exchanged and loaded**

1183 **Encryption disabled**

1184 **User Identities Not Federated**

1185 **Conformance Modes: SAML Authorization Decision Authority**

1186 **Step 1:**

1187 Description: User/SP does Single Sign-On with Persistent Name Identifier. SP communication to the
1188 IdP for the SAML Authentication Request is through HTTP POST binding. IdP provides assertion of
1189 User and IdP returns a signed SAML Response message through HTTP POST binding.

1190 IdP CONFIRM: SP successfully communicated SAML Authentication Request through
1191 HTTP POST binding.

1192 IdP CONFIRM: User has been federated

1193 SP CONFIRM: IdP returns signed SAML Response through HTTP POST binding.

1194 **Step 2:**

1195 Description: SAML Requester enables HTTP Basic Authentication.

1196 SAML Requester CONFIRM: HTTP Basic Authentication enabled.

1197 **Step 3:**

1198 Description: SAML Responder sets Authorization Query to never permitted which means subject is
1199 never authorized for access.

1200 SAML Responder CONFIRM: AuthzQuery Resource=never

1201 **Step 4:**

1202 Description: SAML Requester sends Authorization Query to SAML Responder through SOAP
1203 binding. SAML Responder returns SAML Response.

1204 SAML Responder CONFIRM: SAML Requester sent Authorization Query.

1205 SAML Requester CONFIRM: SAML Responder returned the SAML Response.

1206 **Step 5:**

1207 Description: SAML Responder sets authorization query to maybe permitted if authentication is
1208 matched which means subject is authorized if it is a “particular” subject.

1209 SAML Responder CONFIRM: AuthzQuery Resource=maybe

1210 **Step 6:**

1211 Description: SAML Requester sends Authorization Query to SAML Responder through SOAP
1212 binding. SAML Responder returns SAML Response.

1213 SAML Responder CONFIRM: SAML Requester sent Authorization Query.

1214 SAML Requester CONFIRM: SAML Responder returned the SAML Response.

1215 **Step 7:**

1216 Description: SAML Responder sets Authorization Query to always permitted which means subject is
1217 always authorized.

1218 SAML Responder CONFIRM: AuthzQuery Resource=always

1219 **Step 8:**

1220 Description: SAML Requester sends Authorization Query to SAML Responder through SOAP
1221 binding. SAML Responder returns SAML Response.

1222 SAML Responder CONFIRM: SAML Requester sent Authorization Query.

1223 SAML Requester CONFIRM: SAML Responder returned the SAML Response.

1224 **Test Case P – Error Testing**

1225 **Preconditions:**

1226 **Metadata exchanged and loaded**

1227 **Encryption disabled**

1228 **User Identities Not Federated**

1229 **Conformance Modes: IdP, SP, SP Lite, eGov, POST**

1230 **NOTE – Test Steps 2-11 involve the Liberty Error Test Tool. Metadata for conducting these**
1231 **tests will be exchanged.**

1232 **Step 1:**

1233 Description: Successful SSO using Artifact Resolution as described in Steps 1-3 of Test Case B are
1234 done. Once those steps are complete, the SP reissues the same <Artifact> in a new
1235 <ArtifactResolve> message. The IdP recognizes the reissued <Artifact> and refuses it.
1236 <ArtifactResponse> is returned with no embedded message.

1237 IdP CONFIRM: Successful SSO using Artifact Binding.

1238 IdP CONFIRM: Second <ArtifactResolve> message received using same <Artifact> and
1239 refused.

1240 SP CONFIRM: <ArtifactResponse> is returned with no embedded message.

1241 **Step 2:**

1242 Description: Test Harness POSTs an unsolicited SAML Response message containing a valid
1243 assertion.

1244 SP CONFIRM: SAML Response was received and assertion accepted.

1245 **Step 3:**

1246 Description: Test Harness re-POSTs the assertion that was successful during the initialization of this
1247 test sequence.

1248 SP CONFIRM: Assertions are not replayed within the validity period of the assertion.

1249 **Step 4:**

1250 Description: The assertion of the SAML Response from Step 2 is altered and sent without re-signing
1251 in a HTTP POST from Test Harness.

1252 SP CONFIRM: SP rejects the message.

1253 **Step 5:**

1254 Description: The assertion of the SAML Response from Step 2 is sent but signed with the wrong
1255 signing key in a HTTP POST from Test Harness.

1256 SP CONFIRM: SP rejects the message.

1257 **Step 6:**

1258 Description: The Test Harness constructs a SAML Response message with an incorrect Recipient
1259 attribute. Recipient attribute is in the <SubjectConfirmationData> element.

1260 SP CONFIRM: SP detects and rejects the message.

1261 **Step 7:**

1262 Description: The Test Harness sends an altered assertion in the SAML Response. A different
1263 Method URN is substituted in the assertion's <SubjectConfirmation> element other than the
1264 required Method of urn:oasis:names:tc:SAML:2.0:cm:bearer.

1265 SP CONFIRM: SP detects and rejects the message.

1266 **Step 8:**

1267 Description: The Test Harness POSTs a SAML Response containing an assertion which does not
1268 contain an <AudienceRestriction> including the SP's unique identifier as an <Audience>.

1269 SP CONFIRM: SP rejects the assertion.

1270 **Step 9:**

1271 Description: The Test Harness sets the *NotOnOrAfter* attribute to a date/time that has occurred in
1272 past with respect the date/time of executing this test step.

1273 SP CONFIRM: The SP to reject the assertion because of the *NotOnOrAfter* attribute.

1274 **Step 10:**

1275 Description: The Test Harness sets the *NotBefore* attribute to a date/time in the future with respect to
1276 the date/time of executing this test step.

1277 SP CONFIRM: The SP to reject the assertion because of the *NotBefore* attribute.

1278 **Step 11:**

1279 Description: The Test Harness includes a <Condition> extension element in the <Conditions>
1280 element of the assertion that cannot be understood.

1281 SP CONFIRM: The SP rejects the assertion.

1282 **Test Case Q – Requested AuthnContext**

1283 **Preconditions:**

- 1284 **Metadata exchanged and loaded**
- 1285 **Encryption disabled**
- 1286 **User Identities Not Federated**

1287 **Conformance Modes: eGov Profile**

1288 **Note: Section [[AuthenticationContexts](#)] within this document describes the strength of**
 1289 **the AuthnContext classes used for comparison used in this test case.**

1290 **Step 1: Issue <AuthnRequest> with Assigned <RequestedAuthnContext>**

1291 Description: For each iteration in Table Q.1, SP sends an <AuthnRequest> to the IdP. Within
 1292 <NameIDPolicy>, AllowCreate is set to “true”, and the with format is set to 'persistent'. The
 1293 *ForceAuthn* attribute is set to “true”. SP communication to the IdP for the SAML Authentication
 1294 Request is through HTTP Redirect binding.

1295 For each iteration, the SP inserts a <RequestedAuthnContext> element into the <AuthnRequest>
 1296 message. The authentication context requested and the *Comparison* attribute setting is defined in
 1297 Table Q.1. Prior to each iteration, the IdP enables its authenticating context for the User as defined in
 1298 the table. The expected Status value for the <Response> message is also listed in the table.

1299 **TABLE Q.1**

Iteration	SP Requested AC	<i>Comparison</i>	IdP Supported AC	Status Response
1	Password	“exact”	InternetProtocol	NoAuthnContext
2	Password	“minimum”	InternetProtocol	NoAuthnContext
3	Password	“better”	InternetProtocol	NoAuthnContext
4	InternetProtocol	“exact”	InternetProtocol	Success
5	InternetProtocol	“minimum”	InternetProtocol	Success
6	InternetProtocol	“maximum”	InternetProtocol	Success
7	InternetProtocol	“maximum”	Password	NoAuthnContext
8	InternetProtocol	“better”	Password	Success

1300 **SP CONFIRM:** Every iteration from Table Q.1 is executed, and all messages, actions and
 1301 responses match the results assigned by the table.

1302 **IdP CONFIRM:** Every iteration from Table Q.1 is executed, and all messages, actions and
 1303 responses match the results assigned by the table.

1304 **Test Case R – User Consent**

1305 **Preconditions:**

1306 **Metadata exchanged and loaded**

1307 **Encryption disabled**

1308 **User Identities Not Federated**

1309 **Conformance Modes: eGov**

1310 **Step 1: User Consent StatusResponse**

1311 Description: IdP must provide means for User to provide authentication consent per the different
1312 consent values listed in Table R.1. Consent conditions are listed in section 8.4 of [SAMLCore]. The
1313 exact means used is left to the individual IdP. After user provides assigned credentials for
1314 authentication, IdP provides assertion of User and returns <Assertion> in an unsolicited signed
1315 SAML Response message through HTTP POST binding. The *Consent* attribute is included in the
1316 StatusResponse. The test step is repeated through each iteration in Table R.1

1317 **TABLE R.1**

Iteration	Consent value
1	urn:oasis:names:tc:SAML:2.0:consent:obtained
2	urn:oasis:names:tc:SAML:2.0:consent:prior
3	urn:oasis:names:tc:SAML:2.0:consent:current-implicit
4	urn:oasis:names:tc:SAML:2.0:consent:current-explicit
5	urn:oasis:names:tc:SAML:2.0:consent:unspecified

1318 SP CONFIRM: IdP sends signed SAML Response through HTTP POST binding.

1319 SP CONFIRM: Valid assertion is returned from IdP.

1320 SP CONFIRM: *Consent* attribute match values in Table R.1

1321 SP CONFIRM: User A identity has been federated with IdP.

1322 IdP CONFIRM: User A identity has been federated with SP.

1323 **Test Case S – Assertion Attribute**

1324 **Preconditions:**

- 1325 **Metadata exchanged and loaded**
- 1326 **Encryption disabled**
- 1327 **User Identities Not Federated**

1328 **Conformance Modes: eGov**

1329 **Step 1: User A, AttributeStatement in Assertion Response**

1330 Description: User A requires authentication. SP sends <AuthnRequest> with AllowCreate is set to
 1331 TRUE. SP communication to the IdP for the SAML Authentication Request is through HTTP
 1332 Redirect binding. User A provides assigned credentials for authentication. IdP provides assertion of
 1333 User A. The attributes in the table below are assigned to User A and are to be returned in a single
 1334 <AttributeStatement> in the assertion. IdP returns <Assertion> in a signed SAML Response message
 1335 through HTTP POST binding.

1336 **TABLE S.1**

Attribute Name	AttributeValue (string)	NameFormat
LastName	Wall	“basic”
urn:oid:2.5.4.40	John	“uri”
Position	PG	“unspecified”

- 1337 SP CONFIRM: IdP returns signed SAML Response through HTTP POST binding.
- 1338 SP CONFIRM: Valid assertion is returned from IdP.
- 1339 SP CONFIRM: Returned attributes match values in Table S.1
- 1340 SP CONFIRM: User A identity has been federated with IdP.
- 1341 IdP CONFIRM: User A identity has been federated with SP.

1342 **Step 2: User B, No AttributeStatement in Assertion Response**

1343 Description: User B requires authentication. SP sends <AuthnRequest> with AllowCreate is set to
 1344 TRUE. SP communication to the IdP for the SAML Authentication Request is through HTTP
 1345 Redirect binding. User B provides assigned credentials for authentication. IdP provides assertion of
 1346 User B. No <AttributeStatement> is returned in the <Assertion>.

- 1347 SP CONFIRM: IdP returns signed SAML Response through HTTP POST binding.
- 1348 SP CONFIRM: Valid assertion is returned from IdP.
- 1349 SP CONFIRM: No <AttributeStatement> is returned in <Assertion>.
- 1350 SP CONFIRM: User B identity has been federated with IdP.
- 1351 IdP CONFIRM: User B identity has been federated with SP.

1352 **Test Case T – Unspecified Format**

1353 **Preconditions:**

1354 **Metadata exchanged and loaded**

1355 **Encryption disabled**

1356 **User Identities Not Federated**

1357 **Conformance Modes: eGov**

1358 **Step 1: AuthnRequest, 'Unspecified' NameID format, Redirect Binding, Federate**

1359 Description: User/SP does Single Sign-On with AllowCreate is set to TRUE. The with Name
1360 Identifier format is set to 'unspecified'. SP communication to the IdP for the SAML Authentication
1361 Request is through HTTP Redirect binding.

1362 IdP CONFIRM: SP successfully communicated SAML Authentication Request through
1363 HTTP Redirect binding.

1364 IdP CONFIRM: Name ID format is 'unspecified'.

1365 **Step 2: Assertion Response, POST binding**

1366 Description: User provides assigned credentials for authentication. IdP provides assertion of User.
1367 NameID format is set to 'persistent'. In <Assertion>, *SessionIndex* attribute must be present but
1368 *SessionNotOnOrAfter* must not be present. IdP returns <Assertion> in a signed SAML Response
1369 message through HTTP POST binding.

1370 SP CONFIRM: IdP returns signed SAML Response through HTTP POST binding.

1371 SP CONFIRM: Valid assertion is returned from IdP.

1372 SP CONFIRM: NameID format is 'persistent'.

1373 SP CONFIRM: *SessionIndex* is present.

1374 SP CONFIRM: *SessionNotOnOrAfter* is not present.

1375 SP CONFIRM: User identity has been federated with IdP.

1376 IdP CONFIRM: User identity has been federated with SP.

1377 References

- 1378 [SAMLTP31] Kyle Meadors, et al, "SAML 2.0 Interoperability Testing Procedures, V3.1,
1379 Errata J," Liberty Alliance Project (July 2008),
1380 [http://www.projectliberty.org/liberty/content/download/4160/27946/file/Liberty](http://www.projectliberty.org/liberty/content/download/4160/27946/file/Liberty_Interoperability_SAML_Test_Plan_v3.1.pdf)
1381 [y_Interoperability_SAML_Test_Plan_v3.1.pdf](http://www.projectliberty.org/liberty/content/download/4160/27946/file/Liberty_Interoperability_SAML_Test_Plan_v3.1.pdf)
- 1382 [ExcXMLCan] John Boyer et al, "Exclusive XML Canonicalization Version 1.0, W3C
1383 Recommendation", W3C (July 2002), <http://www.w3.org/TR/xml-exc-c14n/>
- 1384 [SAMLAuthnCxt] J. Kemp et al, "Authentication Context for the OASIS Security Assertion
1385 Markup Language (SAML) V2.0," OASIS SSTC (March 2005), [http://](http://docs.oasis-open.org/security/saml/v2.0/saml-authn-context-2.0-os.pdf)
1386 [docs.oasis- open.org/security/saml/v2.0/saml-authn-context-2.0-os.pdf](http://docs.oasis-open.org/security/saml/v2.0/saml-authn-context-2.0-os.pdf).
- 1387 [SAMLBind] Scott Cantor et al, "Bindings for the OASIS Security Assertion Markup
1388 Language (SAML) V2.0," OASIS SSTC (March 2005), [http://docs.oasis-](http://docs.oasis-open.org/security/saml/v2.0/saml-bindings-2.0-os.pdf)
1389 [open.org/security/saml/v2.0/saml-bindings-2.0-os.pdf](http://docs.oasis-open.org/security/saml/v2.0/saml-bindings-2.0-os.pdf)
- 1390 [SAMLConf] Prateek Mishra et al, "Conformance Requirements for the OASIS Security
1391 Assertion Markup Language (SAML) V2.0," OASIS SSTC (March 2005).
1392 <http://docs.oasis-open.org/security/saml/v2.0/saml-conformance-2.0-os.pdf>.
- 1393 [SAMLCore] S. Cantor et al, "Assertions and Protocols for the OASIS Security Assertion
1394 Markup Language (SAML) V2.0," OASIS SSTC (March 2005),
1395 <http://docs.oasis-open.org/security/saml/v2.0/saml-core-2.0-os.pdf>.
- 1396 [SAMLErrata] Jahan Moreh, "Errata for the OASIS Security 2 Assertion Markup Language
1397 (SAML) V2.0, Working Draft 28," OASIS SSTC (May 8, 2006),
1398 [http://www.oasis-open.org/committees/download.php/18070/sstc-saml-errata-](http://www.oasis-open.org/committees/download.php/18070/sstc-saml-errata-2.0-draft-28.pdf)
1399 [2.0-draft-28.pdf](http://www.oasis-open.org/committees/download.php/18070/sstc-saml-errata-2.0-draft-28.pdf)
- 1400 [SAMLLDAP] S. Cantor et al, "SAML V2.0 X.500/LDAP Attribute Profile," OASIS SSTC
1401 (December 19, 2006), [http://docs.oasis-open.org/security/saml/SpecDrafts-](http://docs.oasis-open.org/security/saml/SpecDrafts-Post2.0/sstc-saml-attribute-x500-cd-01.pdf)
1402 [Post2.0/sstc-saml-attribute-x500-cd-01.pdf](http://docs.oasis-open.org/security/saml/SpecDrafts-Post2.0/sstc-saml-attribute-x500-cd-01.pdf)
- 1403 [SAMLMeta] S. Cantor et al, "Metadata for the OASIS Security Assertion Markup
1404 Language (SAML) V2.0," OASIS SSTC (March 2005), [http://docs.oasis-](http://docs.oasis-open.org/security/saml/v2.0/saml-metadata-2.0-os.pdf)
1405 [open.org/security/saml/v2.0/saml-metadata-2.0-os.pdf](http://docs.oasis-open.org/security/saml/v2.0/saml-metadata-2.0-os.pdf).
- 1406 [SAMLMetaExt] Tom Scavo et al, "SAML Metadata Extension for Query Requesters,
1407 Committee Draft 01", OASIS SSTC (March 2006), [http://www.oasis-](http://www.oasis-open.org/committees/download.php/18052/sstc-saml-metadata-ext-query-cd-01.pdf)
1408 [open.org/committees/download.php/18052/sstc-saml-metadata-ext-query-cd-](http://www.oasis-open.org/committees/download.php/18052/sstc-saml-metadata-ext-query-cd-01.pdf)
1409 [01.pdf](http://www.oasis-open.org/committees/download.php/18052/sstc-saml-metadata-ext-query-cd-01.pdf)
- 1410 [SAMLProf] S. Cantor et al, "Profiles for the OASIS Security Assertion Markup Language
1411 (SAML) V2.0," OASIS SSTC (March 2005), [http://docs.oasis-](http://docs.oasis-open.org/security/saml/v2.0/saml-profiles-2.0-os.pdf)
1412 [open.org/security/saml/v2.0/saml-profiles-2.0-os.pdf](http://docs.oasis-open.org/security/saml/v2.0/saml-profiles-2.0-os.pdf).
- 1413 [SAMLSec] Frederick Hirsch et al, "Security and Privacy Considerations for the OASIS
1414 Security Assertion Markup Language (SAML) V2.0," OASIS SSTC (March
1415 2005), [http://docs.oasis-open.org/security/saml/v2.0/saml-sec-consider-2.0-](http://docs.oasis-open.org/security/saml/v2.0/saml-sec-consider-2.0-os.pdf)
1416 [os.pdf](http://docs.oasis-open.org/security/saml/v2.0/saml-sec-consider-2.0-os.pdf)
-

