



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

2 **Test Criteria**

3 **SAML 2.0**

4 **Version 3.2**

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.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 are 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, C, D, 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 **Conformance Modes: IdP, SP, IdP Lite, SP Lite, eGov**

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

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

312 IdP CONFIRM: SP successfully communicated SAML Authentication Request through
313 HTTP Redirect binding.

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

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

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

319 SP CONFIRM: Artifact is sent by IdP.

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

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

322 Description: SP sends ArtifactResolve message to IdP referencing artifact through synchronous
323 SOAP binding. IdP confirms artifact and returns <Response> message to SP in ArtifactResponse
324 message.

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

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

328 IdP CONFIRM: Receives ArtifactResolve message.

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

330 Description: IdP logs out User session. IdP sends a signed LogoutRequest message to SP using
331 synchronous SOAP binding. SP logs out User session. SP returns a signed LogoutResponse message
332 to IdP using synchronous SOAP binding.

333 IdP CONFIRM: User logged out at IdP.

334 SP CONFIRM: Receives signed LogoutRequest through SOAP binding.

335 SP CONFIRM: User logged out at SP.

336 IdP CONFIRM: Receives signed LogoutResponse through SOAP binding.

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

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

341 IdP CONFIRM: SP successfully communicated SAML Authentication Request through
342 HTTP Redirect binding.

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

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

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

348 SP CONFIRM: Artifact is sent by IdP.

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

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

351 Description: SP sends ArtifactResolve message to IdP referencing artifact through synchronous
352 SOAP binding. IdP confirms artifact and returns <Response> message to SP in ArtifactResponse
353 message.

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

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

357 IdP CONFIRM: Receives ArtifactResolve message.

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

359 Description: SP logs out User session. SP sends a signed LogoutRequest message to IdP using
360 synchronous SOAP binding. IdP logs out User session. IdP returns a signed LogoutResponse
361 message to SP using synchronous SOAP binding.

362 SP CONFIRM: User logged out at SP.

363 IdP CONFIRM: Receives signed LogoutRequest through SOAP binding.

364 IdP CONFIRM: User logged out at IdP.

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

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

367 **Preconditions:**

368 **Metadata exchanged and loaded**

369 **Encryption disabled**

370 **User Identities Not Federated**

371 **Conformance Modes: IdP, SP, eGov**

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

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

376 IdP CONFIRM: SP successfully communicated SAML Authentication Request through
377 HTTP Redirect binding.

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

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

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

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

383 SP CONFIRM: Receives signed assertion is returned from IdP.

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

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

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

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

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

392 SP CONFIRM: New NameID is accepted.

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

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

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

398 SP CONFIRM: User logged out at SP.

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

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

401 IdP CONFIRM: User logged out at IdP.

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

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

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

407 IdP CONFIRM: SP successfully communicated SAML Authentication Request through
408 HTTP Redirect binding.

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

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

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

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

414 SP CONFIRM: Valid assertion is returned from IdP.

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

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

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

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

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

423 IdP CONFIRM: New NameID is accepted.

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

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

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

429 IdP CONFIRM: User logged out at IdP.

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

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

432 SP CONFIRM: User logged out at SP.

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

434 **Step 9: AuthnRequest, Redirect Binding, Federate**

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

438 IdP CONFIRM: SP successfully communicated SAML Authentication Request through
439 HTTP Redirect binding.

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

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

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

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

445 SP CONFIRM: Valid assertion is returned from IdP.

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

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

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

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

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

454 IdP CONFIRM: Federation of User is terminated.

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

456 SP CONFIRM: Federation of User is terminated.

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

458 **Preconditions:**

- 459 **Metadata exchanged and loaded**
- 460 **Encryption enabled for Assertions**
- 461 **Encryption enabled for NameIDs in MNI messages**
- 462 **Encryption enabled for NameIDs in SLO messages**
- 463 **User Identities Not Federated**

464 **Conformance Modes: IdP, SP, eGov**

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

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

469 IdP CONFIRM: SP successfully communicated SAML Authentication Request through
470 HTTP Redirect binding.

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

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

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

477 SP CONFIRM: Artifact is sent by IdP.

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

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

480 Description:

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

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

484 IdP CONFIRM: Receives ArtifactResolve message.

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

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

490 IdP CONFIRM: Receives signed ManageNameIdRequest on SOAP binding.

491 IdP CONFIRM: New NameID is accepted.

492 SP CONFIRM: Receives signed ManageNameIdResponse on SOAP binding.

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

494 Description: IdP logs out User session. IdP sends a signed LogoutRequest message to SP using
495 synchronous SOAP binding. SP logs out User session. SP returns a signed LogoutResponse message
496 to IdP using synchronous SOAP binding.

497 IdP CONFIRM: User logged out at IdP.

498 SP CONFIRM: Receives signed LogoutRequest through SOAP binding.

499 SP CONFIRM: User logged out at SP.

500 IdP CONFIRM: Receives signed LogoutResponse through SOAP binding.

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

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

505 IdP CONFIRM: SP successfully communicated SAML Authentication Request through
506 HTTP Redirect binding.

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

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

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

512 SP CONFIRM: Artifact is sent by IdP.

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

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

515 Description: SP sends ArtifactResolve message to IdP referencing artifact through synchronous
516 SOAP binding. IdP confirms artifact and returns <Response> message to SP in ArtifactResponse
517 message.

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

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

521 IdP CONFIRM: Receives ArtifactResolve message.

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

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

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

528 SP CONFIRM: New NameID is accepted.

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

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

531 Description: SP logs out User session. SP sends a signed LogoutRequest message to IdP using
532 synchronous SOAP binding. IdP logs out User session. IdP returns a signed LogoutResponse
533 message to SP using synchronous SOAP binding.

534 SP CONFIRM: User logged out at SP.

535 IdP CONFIRM: Receives signed LogoutRequest through SOAP binding.

536 IdP CONFIRM: User logged out at IdP.

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

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

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

542 IdP CONFIRM: SP successfully communicated SAML Authentication Request through
543 HTTP Redirect binding.

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

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

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

549 SP CONFIRM: Artifact is sent by IdP.

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

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

552 Description: SP sends ArtifactResolve message to IdP referencing artifact through synchronous
553 SOAP binding. IdP confirms artifact and returns <Response> message to SP in ArtifactResponse
554 message.

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

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

558 IdP CONFIRM: Receives ArtifactResolve message.

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

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

563 SP CONFIRM: Receives signed ManageNameIdRequest with <Terminate> element on
564 SOAP binding.

565 SP CONFIRM: Federation of User is terminated.

566 IdP CONFIRM: Receives signed ManageNameIdResponse on SOAP binding.

567 IdP CONFIRM: Federation of User is terminated.

568 **Test Case E – POST Binding**

569 **Preconditions:**

570 **Metadata exchanged and loaded**

571 **Encryption disabled**

572 **User Identities Not Federated**

573 **Conformance Modes: POST Binding**

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

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

579 IdP CONFIRM: SP successfully communicated SAML Authentication Request through
580 HTTP POST binding.

581 IdP CONFIRM: User has been federated

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

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

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

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

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

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

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

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

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

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

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

597 IdP CONFIRM: SP successfully communicated SAML Authentication Request through
598 HTTP POST binding.

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

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

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

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

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

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

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

609 IdP CONFIRM: SP successfully communicated SAML Authentication Request through
610 HTTP POST binding.

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

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

613 Description: IdP sends signed ManageNameIdRequest message with the Terminate element to the
614 SP using HTTP POST binding. User session is terminated. SP returns signed
615 ManageNameIdResponse message using HTTP POST binding.

616 SP CONFIRM: Receives signed ManageNameIdRequest with Terminate flag on HTTP
617 POST binding.

618 SP CONFIRM: User session is terminated.

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

620 IdP CONFIRM: User session is terminated.

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

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

626 IdP CONFIRM: SP successfully communicated SAML Authentication Request through
627 HTTP POST binding.

628 IdP CONFIRM: User has been federated

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

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

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

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

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

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

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

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

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

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

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

644 IdP CONFIRM: SP successfully communicated SAML Authentication Request through
645 HTTP POST binding.

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

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

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

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

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

652 **Test Case F – IdP Proxy**

653 **Preconditions:**

654 **Metadata exchanged and loaded**

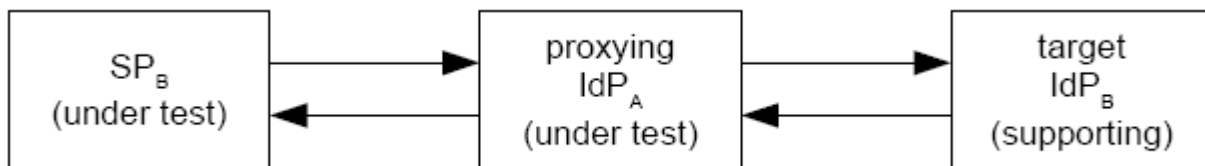
655 **Encryption disabled**

656 **User Identities Not Federated**

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

658 **Background on IdP Proxy**

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



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

666 **Step 1: ProxyCount=0**

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

668 SP CONFIRM: SP has disallowed proxy.

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

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

673 IdP_A CONFIRM: ProxyCount is set to 0.

674 IdP_A CONFIRM: User is not authenticated.

675 **Step 3: Response Failure**

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

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

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

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

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

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

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

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

689 IdP_A CONFIRM: ProxyCount is not set.

690 IdP_A CONFIRM: User is not authenticated.

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

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

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

694 IdP_B CONFIRM: Receives AuthnRequest from IdP_A.

695 IdP_B CONFIRM: ProxyCount is set to 0.

696 IdP_B CONFIRM: <IdPList> includes IdP_B.

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

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

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

701 IdP_A CONFIRM: Valid assertion is returned from IdP_B.

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

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

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

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

707 SP CONFIRM: Valid assertion is returned from IdP_A.

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

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

710 Description: IdP_A logs out User session. IdP_A sends a signed LogoutRequest message to SP using
711 HTTP Redirect binding. SP logs out User session. SP returns a signed LogoutResponse message to
712 IdP_A using HTTP Redirect binding.

713 IdP_A CONFIRM: User logged out at IdP_A.

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

715 SP CONFIRM: User logged out at SP.

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

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

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

719 SP CONFIRM: SP sets ProxyCount to 1.

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

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

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

726 IdP_A CONFIRM: ProxyCount is set to 1.

727 IdP_A CONFIRM: User is not authenticated.

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

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

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

731 IdP_B CONFIRM: Receives AuthnRequest from IdP_A.

732 IdP_B CONFIRM: ProxyCount is set to 0.

733 IdP_B CONFIRM: <IdPList> includes IdP_B.

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

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

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

738 IdP_A CONFIRM: Valid assertion is returned from IdP_B.

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

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

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

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

744 SP CONFIRM: Valid assertion is returned from IdP_A.

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

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

747 Description: IdP_A logs out User session. IdP_A sends a signed LogoutRequest message to SP using
748 HTTP Redirect binding. SP logs out User session. SP returns a signed LogoutResponse message to
749 IdP_A using HTTP Redirect binding.

750 IdP_A CONFIRM: User logged out at IdP_A.

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

752 SP CONFIRM: User logged out at SP.

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

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

755 **Preconditions:**

756 **Metadata exchanged and loaded**

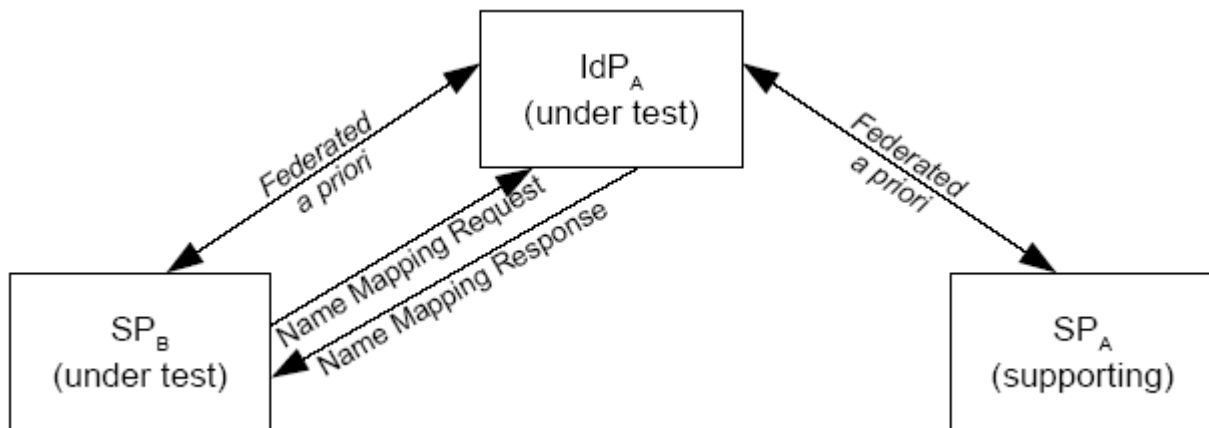
757 **Encryption disabled**

758 **User Identities Not Federated**

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

760 **Background on Name Identifier Mapping Feature**

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



765 **Step 1: SSO at SP_A**

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

769 IdP CONFIRM: SP_A successfully communicated SAML Authentication Request through
770 HTTP Redirect binding.

771 IdP CONFIRM: User has been federated with SP_A.

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

773 SP_A CONFIRM: User has been federated with IdP.

774 **Step 2: SSO at SP_B**

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

778 IdP CONFIRM: SP_B successfully communicated SAML Authentication Request through
779 HTTP Redirect binding.

780 IdP CONFIRM: User has been federated with SP_B.

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

782 SP_B CONFIRM: User has been federated with IdP.

783 **Step 3: NameIDMappingRequest from SP_B**

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

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

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

790 SP_B CONFIRM: Receives Encrypted NameID.

791 **Test Case H – IDP Introduction**

792 **Preconditions:**

793 **Metadata exchanged and loaded**

794 **Encryption disabled**

795 **User Identities Not Federated**

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

797 **Background**

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

800 **Step 1: Clear Cookies**

801 Description: Cookies are cleared from User Browser
802 IdP_A CONFIRM: IdP_A has enabled encryption.

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

804 Description: User logs in at IdP_A. Cookie is set in common domain with IdP_A appended to list of IdPs.
805 IdP_A CONFIRM: User logged in, cookie is set in common domain and IdP_A appended to end
806 of IdP list in cookie.

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

808 Description: User logs in at IdP_B. IdP_B appended to list of IdPs in CDC.
809 IdP_B CONFIRM: User logged in and IdP_B appended to end of IdP list in CDC.

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

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

818 IdP_A CONFIRM: SP successfully communicated signed SAML Authentication Request
819 through HTTP Redirect binding.

820 SP CONFIRM: Cookie was read and IdP_A and IdP_B were present in CDC.

821 SP CONFIRM: IdP_A returns signed assertion through HTTP POST binding.

822 SP CONFIRM: For eGov profile, SP presents list of IdPs for authentication and IdP_A and
823 IdP_B must be present on list.

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

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

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

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

829 SP CONFIRM: User is logged out.

830 **Step 6: CDC is removed**

831 Description: User closes browser. CDC is removed.

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

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

834 **Preconditions:**

835 **Metadata exchanged and loaded**

836 **Encryption disabled**

837 **User Identities Not Federated**

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

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

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

844 IdP CONFIRM: SP successfully communicated SAML Authentication Request through
845 HTTP Redirect binding.

846 IdP CONFIRM: User has been federated.

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

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

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

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

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

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

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

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

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

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

864 IdP CONFIRM: User logged out of Session A.

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

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

867 SP CONFIRM: User logged out of Session A.

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

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

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

874 IdP CONFIRM: SP successfully communicated SAML Authentication Request through
875 HTTP Redirect binding.

876 IdP CONFIRM: User has been federated.

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

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

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

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

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

885 SP CONFIRM: User logged out of Session C.

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

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

888 IdP CONFIRM: User logged out of Session C.

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

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

891 **Preconditions:**

892 **Metadata exchanged and loaded**

893 **Encryption disabled**

894 **User Identities Not Federated**

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

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

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

899 IdP CONFIRM: User has been federated.

900 SP CONFIRM: NameID format is “transient”.

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

902 **Step 2: SLO from SP**

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

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

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

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

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

911 IdP CONFIRM: Artifact resolution is properly done.

912 IdP CONFIRM: User has been federated

913 SP CONFIRM: NameID format is “transient”.

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

915 SP CONFIRM: Artifact resolution is properly done.

916 **Step 4: SLO from IdP**

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

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

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

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

922 **Preconditions:**

923 **Metadata exchanged and loaded**

924 **Encryption disabled**

925 **User Identities Not Federated**

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

927 **Step 1: SSO from SP_A**

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

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

931 IdP CONFIRM: User has been federated with SP_A

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

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

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

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

937 IdP CONFIRM: User has been federated with SP_B

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

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

940 Description: User issues SLO from SP_A to IdP.

941 IdP CONFIRM: SP_A sends signed LogoutRequest for User.

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

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

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

946 IdP CONFIRM: Signed LogoutRequest is sent to SP_A and receives back signed
947 LogoutResponse.

948 IdP CONFIRM: No active session for User.

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

951 SP_A CONFIRM: Receives signed LogoutResponse from IdP.

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

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

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

956 IdP CONFIRM: User has active session.

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

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

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

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

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

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

964 Description: User does SLO from IdP to SP_B.

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

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

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

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

970 IdP CONFIRM: Signed LogoutRequest is sent to SP_A and receives back signed
971 LogoutResponse.

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

974 SP CONFIRM: Receives signed LogoutResponse from IdP.

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

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

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

979 IdP CONFIRM: User has active session.

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

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

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

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

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

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

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

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

989 SP_B CONFIRM: User is logged out locally.

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

991 Description: User issues SLO from SP_A to IdP.

992 IdP CONFIRM: SP_A sends signed LogoutRequest for User.

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

994 **Step 13: PartialLogout Error**

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

998 IdP CONFIRM: Signed LogoutRequest is sent to SP_B and receives back signed
999 LogoutResponse.

1000 IdP CONFIRM: Signed LogoutResponse contains status code of
1001 urn:oasis:names:tc:SAML:2.0:status:PartialLogout.

1002 SP_B CONFIRM: IdP sends signed LogoutResponse, unable to perform SLO, and a signed
1003 LogoutResponse is returned indicating “PartialLogout”.

1004 SP_A CONFIRM: Receives signed LogoutResponse from IdP indicating “PartialLogout.”

1005 **Test Case L – Force Authentication and Passive Authentication**

1006 **Preconditions:**

1007 **Metadata exchanged and loaded**

1008 **Encryption disabled**

1009 **User Identities Not Federated**

1010 **Conformance Modes (Required): IdP, SP, IdP Lite, SP Lite, eGov**

1011 **Step 1: User Logins at IdP**

1012 Description: User logs in at IdP and creates and active session

1013 IdP CONFIRM: User logged in.

1014 **Step 2: SP sets IsPassive=TRUE**

1015 Description: SP is configured to make IsPassive set to TRUE.

1016 SP CONFIRM: SP is configured IsPassive=TRUE.

1017 **Step 3: SSO with isPassive=TRUE**

1018 Description: User/SP does Single Sign-On SP communication to the IdP for the SAML

1019 Authentication Request is through HTTP Redirect binding. IdP provides assertion of User without
1020 interacting with the user. IdP returns a signed SAML Response message through HTTP POST
1021 binding.

1022 IdP CONFIRM: SP successfully communicated SAML Authentication Request through
1023 HTTP Redirect binding.

1024 IdP CONFIRM: User does not interact with IdP or IdP must not take control of user
1025 interface.

1026 SP CONFIRM: IdP returns assertion in signed SAML Response through HTTP POST
1027 binding.

1028 **Step 4: SLO from SP**

1029 Description: SP sends a signed LogoutRequest message to IdP using HTTP Redirect binding. IdP
1030 logs out User session. IdP returns a signed LogoutResponse message.

1031 IdP CONFIRM: Receives signed LogoutRequest on HTTP Redirect binding.

1032 SP CONFIRM: Receives signed LogoutResponse on HTTP Redirect binding.

1033 SP CONFIRM: User is logged out.

1034 **Step 5: SP sets IsPassive=FALSE**

1035 Description: SP is configured to make IsPassive set to FALSE.

1036 SP CONFIRM: SP is configured IsPassive=FALSE.

1037 **Step 6: SSO with isPassive=FALSE**

1038 Description: User/SP does Single Sign-On SP communication to the IdP for the SAML

1039 Authentication Request is through HTTP Redirect binding. IdP interacts with and authenticates the
1040 user. IdP returns a signed SAML Response message through HTTP POST binding.

1041 IdP CONFIRM: SP successfully communicated SAML Authentication Request through
1042 HTTP Redirect binding.
1043 IdP CONFIRM: User does interact with IdP.
1044 SP CONFIRM: IdP returns assertion in signed SAML Response through HTTP POST
1045 binding.

1046 **Step 7: SLO from SP**

1047 Description: SP sends a signed LogoutRequest message to IdP using HTTP Redirect binding. IdP
1048 logs out User session. IdP returns a signed LogoutResponse message.

1049 IdP CONFIRM: Receives signed LogoutRequest on HTTP Redirect binding.

1050 SP CONFIRM: Receives signed LogoutResponse on HTTP Redirect binding.

1051 SP CONFIRM: User is logged out.

1052 **Step 8: User Logins At IdP**

1053 Description: User logs in at IdP and creates and active session

1054 IdP CONFIRM: User logged in.

1055 **Step 9: SP sets ForceAuthn=TRUE**

1056 Description: SP is configured to make ForceAuthn set to TRUE.

1057 SP CONFIRM: SP is configured ForceAuthn=TRUE.

1058 **Step 10: SSO with ForceAuthn=TRUE**

1059 Description: User/SP does Single Sign-On SP communication to the IdP for the SAML
1060 Authentication Request is through HTTP Redirect binding. IdP interacts with User and authenticates
1061 the User. IdP provides assertion of User. IdP returns a signed SAML Response message through
1062 HTTP POST binding.

1063 IdP CONFIRM: SP successfully communicated SAML Authentication Request through
1064 HTTP Redirect binding.

1065 IdP CONFIRM: User interacts with IdP and is authenticated.

1066 SP CONFIRM: IdP returns assertion in signed SAML Response through HTTP POST
1067 binding.

1068 **Step 11: SLO from SP**

1069 Description: SP sends a signed LogoutRequest message to IdP using HTTP Redirect binding. IdP
1070 logs out User session. IdP returns a signed LogoutResponse message.

1071 IdP CONFIRM: Receives signed LogoutRequest on HTTP Redirect binding.

1072 SP CONFIRM: Receives signed LogoutResponse on HTTP Redirect binding.

1073 SP CONFIRM: User is logged out.

1074 **Test Case M – SAML Authentication Authority**

1075 **Preconditions:**

1076 **Metadata exchanged and loaded**

1077 **Encryption disabled**

1078 **User Identities Not Federated**

1079 **Conformance Modes: SAML Authentication Authority**

1080 **Note: Section [[AuthenticationContexts](#)] within this document describes the strength of**
1081 **the AuthnContext classes used for comparison.**

1082 **Test Steps**

1083 **Step 1:**

1084 Description: User/SP does Single Sign-On with Persistent Name Identifier. SP communication to the
1085 IdP for the SAML Authentication Request is through HTTP POST binding. IdP provides assertion of
1086 User and IdP returns a signed SAML Response message through HTTP POST binding.

1087 IdP CONFIRM: SP successfully communicated SAML Authentication Request through
1088 HTTP POST binding.

1089 IdP CONFIRM: User has been federated

1090 SP CONFIRM: IdP returns signed SAML Response through HTTP POST binding.

1091 **Step 2:**

1092 Description: SAML Requester sets AC comparison to “exact”.

1093 SAML Requester CONFIRM: AC comparison=“exact”.

1094 **Step 3:**

1095 Description: SAML Requester sends Authentication Query to SAML Responder through SOAP
1096 binding. SAML Responder returns SAML Response.

1097 SAML Responder CONFIRM: SAML Requester sent Authentication Query.

1098 SAML Requester CONFIRM: SAML Responder returned the SAML Response.

1099 **Step 4:**

1100 Description: SAML Requester sets AC comparison to “better”.

1101 SAML Requester CONFIRM: AC comparison=“better”.

1102 **Step 5:**

1103 Description: SAML Requester sends Authentication Query to SAML Responder through SOAP
1104 binding. SAML Responder returns SAML Response.

1105 SAML Responder CONFIRM: SAML Requester sent Authentication Query.

1106 SAML Requester CONFIRM: SAML Responder returned the SAML Response.

1107 **Step 6:**

1108 Description: SAML Requester sets AC comparison to “minimum”.

1109 SAML Requester CONFIRM: AC comparison="minimum".

1110 **Step 7:**

1111 Description: SAML Requester sends Authentication Query to SAML Responder through SOAP
1112 binding. SAML Responder returns SAML Response.

1113 SAML Responder CONFIRM: SAML Requester sent Authentication Query.

1114 SAML Requester CONFIRM: SAML Responder returned the SAML Response.

1115 **Step 8:**

1116 Description: SAML Requester sets AC comparison to "maximum".

1117 SAML Requester CONFIRM: AC comparison=" maximum".

1118 **Step 9:**

1119 Description: SAML Requester sends Authentication Query to SAML Responder through SOAP
1120 binding. SAML Responder returns SAML Response.

1121 SAML Responder CONFIRM: SAML Requester sent Authentication Query.

1122 SAML Requester CONFIRM: SAML Responder returned the SAML Response.

1123 **Test Case N – SAML Attribute Authority**

1124 **Preconditions:**

1125 **Metadata exchanged and loaded**

1126 **Encryption disabled**

1127 **User Identities Not Federated**

1128 **Conformance Modes: SAML Attribute Authority**

1129 **Step 1:**

1130 Description: User/SP does Single Sign-On with Persistent Name Identifier. SP communication to the
1131 IdP for the SAML Authentication Request is through HTTP POST binding. IdP provides assertion of
1132 User and IdP returns a signed SAML Response message through HTTP POST binding.

1133 IdP CONFIRM: SP successfully communicated SAML Authentication Request through
1134 HTTP POST binding.

1135 IdP CONFIRM: User has been federated

1136 SP CONFIRM: IdP returns signed SAML Response through HTTP POST binding.

1137 **Step 2:**

1138 Description: SAML Responder sets attribute query to no attributes.

1139 SAML Responder CONFIRM: Attribute Query No Attributes.

1140 **Step 3:**

1141 Description: SAML Requester sends Attribute Query to SAML Responder through SOAP binding.

1142 SAML Responder returns SAML Response.

1143 SAML Responder CONFIRM: SAML Requester sent Attribute Query.

1144 SAML Requester CONFIRM: SAML Responder returned the SAML Response.

1145 **Step 4:**

1146 Description: SAML Responder sets attribute query to attribute named.

1147 SAML Responder CONFIRM: Attribute Query Attribute Named.

1148 **Step 5:**

1149 Description: SAML Requester sends Attribute Query to SAML Responder through SOAP binding.

1150 SAML Responder returns SAML Response.

1151 SAML Responder CONFIRM: SAML Requester sent Attribute Query.

1152 SAML Requester CONFIRM: SAML Responder returned the SAML Response.

1153 **Step 6:**

1154 Description: SAML Responder sets attribute query to attribute value.

1155 SAML Responder CONFIRM: Attribute Query Attribute Value.

1156 **Step 7:**

1157 Description: SAML Requester sends Attribute Query to SAML Responder through SOAP binding.

1158 SAML Responder returns SAML Response.

1159 SAML Responder CONFIRM: SAML Requester sent Attribute Query.
1160 SAML Requester CONFIRM: SAML Responder returned the SAML Response.

1161 **Step 8:**

1162 Description: SAML Responder sets attribute query to attribute named. SAML Responder enables
1163 attribute for encryption.

1164 SAML Responder CONFIRM: Attribute Query Attribute Named.

1165 SAML Responder CONFIRM: Encryption assertion enabled.

1166 **Step 9:**

1167 Description: SAML Requester sends Attribute Query to SAML Responder through SOAP binding.

1168 SAML Responder returns SAML Response.

1169 SAML Responder CONFIRM: SAML Requester sent Attribute Query.

1170 SAML Requester CONFIRM: SAML Responder returned the SAML Response.

1171 **Test Case O – SAML Authorization Decision Authority**

1172 **Preconditions:**

1173 **Metadata exchanged and loaded**

1174 **Encryption disabled**

1175 **User Identities Not Federated**

1176 **Conformance Modes: SAML Authorization Decision Authority**

1177 **Step 1:**

1178 Description: User/SP does Single Sign-On with Persistent Name Identifier. SP communication to the
1179 IdP for the SAML Authentication Request is through HTTP POST binding. IdP provides assertion of
1180 User and IdP returns a signed SAML Response message through HTTP POST binding.

1181 IdP CONFIRM: SP successfully communicated SAML Authentication Request through
1182 HTTP POST binding.

1183 IdP CONFIRM: User has been federated

1184 SP CONFIRM: IdP returns signed SAML Response through HTTP POST binding.

1185 **Step 2:**

1186 Description: SAML Requester enables HTTP Basic Authentication.

1187 SAML Requester CONFIRM: HTTP Basic Authentication enabled.

1188 **Step 3:**

1189 Description: SAML Responder sets Authorization Query to never permitted which means subject is
1190 never authorized for access.

1191 SAML Responder CONFIRM: AuthzQuery Resource=never

1192 **Step 4:**

1193 Description: SAML Requester sends Authorization Query to SAML Responder through SOAP
1194 binding. SAML Responder returns SAML Response.

1195 SAML Responder CONFIRM: SAML Requester sent Authorization Query.

1196 SAML Requester CONFIRM: SAML Responder returned the SAML Response.

1197 **Step 5:**

1198 Description: SAML Responder sets authorization query to maybe permitted if authentication is
1199 matched which means subject is authorized if it is a “particular” subject.

1200 SAML Responder CONFIRM: AuthzQuery Resource=maybe

1201 **Step 6:**

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 7:**

1207 Description: SAML Responder sets Authorization Query to always permitted which means subject is
1208 always authorized.

1209 SAML Responder CONFIRM: AuthzQuery Resource=always

1210 **Step 8:**

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 **Test Case P – Error Testing**

1216 **Preconditions:**

1217 **Metadata exchanged and loaded**

1218 **Encryption disabled**

1219 **User Identities Not Federated**

1220 **Conformance Modes: IdP, SP, SP Lite, eGov, POST**

1221 **NOTE – Test Steps 2-11 involve the Liberty Error Test Tool. Metadata for conducting these**
1222 **tests will be exchanged.**

1223 **Step 1:**

1224 Description: Successful SSO using Artifact Resolution as described in Steps 1-3 of Test Case B are
1225 done. Once those steps are complete, the SP reissues the same <Artifact> in a new
1226 <ArtifactResolve> message. The IdP recognizes the reissued <Artifact> and refuses it.
1227 <ArtifactResponse> is returned with no embedded message.

1228 IdP CONFIRM: Successful SSO using Artifact Binding.

1229 IdP CONFIRM: Second <ArtifactResolve> message received using same <Artifact> and
1230 refused.

1231 SP CONFIRM: <ArtifactResponse> is returned with no embedded message.

1232 **Step 2:**

1233 Description: Test Harness POSTs an unsolicited SAML Response message containing a valid
1234 assertion.

1235 SP CONFIRM: SAML Response was received and assertion accepted.

1236 **Step 3:**

1237 Description: Test Harness re-POSTs the assertion that was successful during the initialization of this
1238 test sequence.

1239 SP CONFIRM: Assertions are not replayed within the validity period of the assertion.

1240 **Step 4:**

1241 Description: The assertion of the SAML Response from Step 2 is altered and sent without re-signing
1242 in a HTTP POST from Test Harness.

1243 SP CONFIRM: SP rejects the message.

1244 **Step 5:**

1245 Description: The assertion of the SAML Response from Step 2 is sent but signed with the wrong
1246 signing key in a HTTP POST from Test Harness.

1247 SP CONFIRM: SP rejects the message.

1248 **Step 6:**

1249 Description: The Test Harness constructs a SAML Response message with an incorrect Recipient
1250 attribute. Recipient attribute is in the <SubjectConfirmationData> element.

1251 SP CONFIRM: SP detects and rejects the message.

1252 **Step 7:**

1253 Description: The Test Harness sends an altered assertion in the SAML Response. A different
1254 Method URN is substituted in the assertion's <SubjectConfirmation> element other than the
1255 required Method of urn:oasis:names:tc:SAML:2.0:cm:bearer.

1256 SP CONFIRM: SP detects and rejects the message.

1257 **Step 8:**

1258 Description: The Test Harness POSTs a SAML Response containing an assertion which does not
1259 contain an <AudienceRestriction> including the SP's unique identifier as an <Audience>.

1260 SP CONFIRM: SP rejects the assertion.

1261 **Step 9:**

1262 Description: The Test Harness sets the *NotOnOrAfter* attribute to a date/time that has occurred in
1263 past with respect the date/time of executing this test step.

1264 SP CONFIRM: The SP to reject the assertion because of the *NotOnOrAfter* attribute.

1265 **Step 10:**

1266 Description: The Test Harness sets the *NotBefore* attribute to a date/time in the future with respect to
1267 the date/time of executing this test step.

1268 SP CONFIRM: The SP to reject the assertion because of the *NotBefore* attribute.

1269 **Step 11:**

1270 Description: The Test Harness includes a <Condition> extension element in the <Conditions>
1271 element of the assertion that cannot be understood.

1272 SP CONFIRM: The SP rejects the assertion.

1273 **Test Case Q – Requested AuthnContext**

1274 **Preconditions:**

- 1275 **Metadata exchanged and loaded**
- 1276 **Encryption disabled**
- 1277 **User Identities Not Federated**

1278 **Conformance Modes: eGov Profile**

1279 **Note:** Section [[AuthenticationContexts](#)] within this document describes the strength of
1280 **the AuthnContext classes used for comparison used in this test case.**

1281 **Step 1: Issue <AuthnRequest> with Assigned <RequestedAuthnContext>**

1282 Description: For each iteration in Table Q.1, SP sends an <AuthnRequest> to the IdP. Within
1283 <NameIDPolicy>, AllowCreate is set to “true”, and the with format is set to 'persistent'. The
1284 *ForceAuthn* attribute is set to “true”. SP communication to the IdP for the SAML Authentication
1285 Request is through HTTP Redirect binding.

1286 For each iteration, the SP inserts a <RequestedAuthnContext> element into the <AuthnRequest>
1287 message. The authentication context requested and the *Comparison* attribute setting is defined in
1288 Table Q.1. Prior to each iteration, the IdP enables its authenticating context for the User as defined in
1289 the table. The expected Status value for the <Response> message is also listed in the table.

1290 **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

1291 SP CONFIRM: Every iteration from Table Q.1 is executed, and all messages, actions and
1292 responses match the results assigned by the table.

1293 IdP CONFIRM: Every iteration from Table Q.1 is executed, and all messages, actions and
1294 responses match the results assigned by the table.

1295 **Test Case R – User Consent**

1296 **Preconditions:**

- 1297 **Metadata exchanged and loaded**
- 1298 **Encryption disabled**
- 1299 **User Identities Not Federated**

1300 **Conformance Modes: eGov**

1301 **Step 1: User Consent StatusResponse**

1302 Description: IdP must provide means for User to provide authentication consent per the different
 1303 consent values listed in Table R.1. Consent conditions are listed in section 8.4 of [SAMLCore]. The
 1304 exact means used is left to the individual IdP. After user provides assigned credentials for
 1305 authentication, IdP provides assertion of User and returns <Assertion> in an unsolicited signed
 1306 SAML Response message through HTTP POST binding. The *Consent* attribute is included in the
 1307 StatusResponse. The test step is repeated through each iteration in Table R.1

1308 **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:uspecified

- 1309 SP CONFIRM: IdP sends signed SAML Response through HTTP POST binding.
- 1310 SP CONFIRM: Valid assertion is returned from IdP.
- 1311 SP CONFIRM: *Consent* attribute match values in Table R.1
- 1312 SP CONFIRM: User A identity has been federated with IdP.
- 1313 IdP CONFIRM: User A identity has been federated with SP.

1314 **Test Case S – Assertion Attribute**

1315 **Preconditions:**

- 1316 **Metadata exchanged and loaded**
- 1317 **Encryption disabled**
- 1318 **User Identities Not Federated**

1319 **Conformance Modes: eGov**

1320 **Step 1: User A, AttributeStatement in Assertion Response**

1321 Description: User A requires authentication. SP sends <AuthnRequest> with AllowCreate is set to
 1322 TRUE. SP communication to the IdP for the SAML Authentication Request is through HTTP
 1323 Redirect binding. User A provides assigned credentials for authentication. IdP provides assertion of
 1324 User A. The attributes in the table below are assigned to User A and are to be returned in a single
 1325 <AttributeStatement> in the assertion. IdP returns <Assertion> in a signed SAML Response message
 1326 through HTTP POST binding.

1327 **TABLE S.1**

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

- 1328 SP CONFIRM: IdP returns signed SAML Response through HTTP POST binding.
- 1329 SP CONFIRM: Valid assertion is returned from IdP.
- 1330 SP CONFIRM: Returned attributes match values in Table S.1
- 1331 SP CONFIRM: User A identity has been federated with IdP.
- 1332 IdP CONFIRM: User A identity has been federated with SP.

1333 **Step 2: User B, No AttributeStatement in Assertion Response**

1334 Description: User B requires authentication. SP sends <AuthnRequest> with AllowCreate is set to
 1335 TRUE. SP communication to the IdP for the SAML Authentication Request is through HTTP
 1336 Redirect binding. User B provides assigned credentials for authentication. IdP provides assertion of
 1337 User B. No <AttributeStatement> is returned in the <Assertion>.

- 1338 SP CONFIRM: IdP returns signed SAML Response through HTTP POST binding.
- 1339 SP CONFIRM: Valid assertion is returned from IdP.
- 1340 SP CONFIRM: No <AttributeStatement> is returned in <Assertion>.
- 1341 SP CONFIRM: User B identity has been federated with IdP.
- 1342 IdP CONFIRM: User B identity has been federated with SP.

1343 **Test Case T – Unspecified Format**

1344 **Preconditions:**

1345 **Metadata exchanged and loaded**

1346 **Encryption disabled**

1347 **User Identities Not Federated**

1348 **Conformance Modes: eGov**

1349 **Step 1: AuthnRequest, 'Unspecified' NameID format, Redirect Binding, Federate**

1350 Description: User/SP does Single Sign-On with AllowCreate is set to TRUE. The with Name
1351 Identifier format is set to 'unspecified'. SP communication to the IdP for the SAML Authentication
1352 Request is through HTTP Redirect binding.

1353 IdP CONFIRM: SP successfully communicated SAML Authentication Request through
1354 HTTP Redirect binding.

1355 IdP CONFIRM: Name ID format is 'unspecified'.

1356 **Step 2: Assertion Response, POST binding**

1357 Description: User provides assigned credentials for authentication. IdP provides assertion of User.
1358 NameID format is set to 'persistent'. In <Assertion>, *SessionIndex* attribute must be present but
1359 *SessionNotOnOrAfter* must not be present. IdP returns <Assertion> in a signed SAML Response
1360 message through HTTP POST binding.

1361 SP CONFIRM: IdP returns signed SAML Response through HTTP POST binding.

1362 SP CONFIRM: Valid assertion is returned from IdP.

1363 SP CONFIRM: NameID format is 'persistent'.

1364 SP CONFIRM: *SessionIndex* is present.

1365 SP CONFIRM: *SessionNotOnOrAfter* is not present.

1366 SP CONFIRM: User identity has been federated with IdP.

1367 IdP CONFIRM: User identity has been federated with SP.

1368 References

- 1369 [SAMLTP31] Kyle Meadors, et al, "SAML 2.0 Interoperability Testing Procedures, V3.1,
1370 Errata J," Liberty Alliance Project (July 2008),
1371 [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)
1372 [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)
- 1373 [ExcXMLCan] John Boyer et al, "Exclusive XML Canonicalization Version 1.0, W3C
1374 Recommendation", W3C (July 2002), <http://www.w3.org/TR/xml-exc-c14n/>
- 1375 [SAMLAuthnCxt] J. Kemp et al, "Authentication Context for the OASIS Security Assertion
1376 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)
1377 [docs.oasis-](http://docs.oasis-open.org/security/saml/v2.0/saml-authn-context-2.0-os.pdf) open.org/security/saml/v2.0/saml-authn-context-2.0-os.pdf.
- 1378 [SAMLBind] Scott Cantor et al, "Bindings for the OASIS Security Assertion Markup
1379 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)
1380 [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)
- 1381 [SAMLConf] Prateek Mishra et al, "Conformance Requirements for the OASIS Security
1382 Assertion Markup Language (SAML) V2.0," OASIS SSTC (March 2005).
1383 <http://docs.oasis-open.org/security/saml/v2.0/saml-conformance-2.0-os.pdf>.
- 1384 [SAMLCore] S. Cantor et al, "Assertions and Protocols for the OASIS Security Assertion
1385 Markup Language (SAML) V2.0," OASIS SSTC (March 2005),
1386 <http://docs.oasis-open.org/security/saml/v2.0/saml-core-2.0-os.pdf>.
- 1387 [SAMLErrata] Jahan Moreh, "Errata for the OASIS Security 2 Assertion Markup Language
1388 (SAML) V2.0, Working Draft 28," OASIS SSTC (May 8, 2006),
1389 [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)
1390 [2.0-draft-28.pdf](http://www.oasis-open.org/committees/download.php/18070/sstc-saml-errata-2.0-draft-28.pdf)
- 1391 [SAMLLDAP] S. Cantor et al, "SAML V2.0 X.500/LDAP Attribute Profile," OASIS SSTC
1392 (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)
1393 [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)
- 1394 [SAMLMeta] S. Cantor et al, "Metadata for the OASIS Security Assertion Markup
1395 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)
1396 [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).
- 1397 [SAMLMetaExt] Tom Scavo et al, "SAML Metadata Extension for Query Requesters,
1398 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)
1399 [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)
1400 [01.pdf](http://www.oasis-open.org/committees/download.php/18052/sstc-saml-metadata-ext-query-cd-01.pdf)
- 1401 [SAMLProf] S. Cantor et al, "Profiles for the OASIS Security Assertion Markup Language
1402 (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)
1403 [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).
- 1404 [SAMLSec] Frederick Hirsch et al, "Security and Privacy Considerations for the OASIS
1405 Security Assertion Markup Language (SAML) V2.0," OASIS SSTC (March
1406 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)
1407 [os.pdf](http://docs.oasis-open.org/security/saml/v2.0/saml-sec-consider-2.0-os.pdf)
-

