



---

## 2 Kerberos SAML Profiles

### 3 Working Draft 01, 9 January 2004

#### 4 Document identifier:

5 draft-sstc-solution-profile-kerberos-01

#### 6 Location:

7 [http://www.oasis-open.org/committees/documents.php?wg\\_abbrev=security](http://www.oasis-open.org/committees/documents.php?wg_abbrev=security)

#### 8 Editors:

9 John Hughes, Entegriety Solutions([john.hughes@entegriety.com](mailto:john.hughes@entegriety.com))

10 Tim Alsop, CyberSafe Ltd([Tim.Alsop@CyberSafe.Ltd.UK](mailto:Tim.Alsop@CyberSafe.Ltd.UK))

#### 11 Contributors:

12 TBD

#### 13 Abstract:

14 This document describes a number of use cases and profiles pertaining to the application of  
15 SAML with Kerberos, DCE and Windows.

#### 16 Status:

17 Committee members should send comments on this specification to the  
18 [securityservices@lists.oasis-open.org](mailto:securityservices@lists.oasis-open.org) list. Others should subscribe to and send comments to the  
19 [security-services-comment@lists.oasis-open.org](mailto:security-services-comment@lists.oasis-open.org) list. To subscribe, send an email message to  
20 [security-services-comment-request@lists.oasis-open.org](mailto:security-services-comment-request@lists.oasis-open.org) with the word "subscribe" as the body of  
21 the message.

22  
23 For information on whether any patents have been disclosed that may be essential to  
24 implementing this specification, and any offers of patent licensing terms, please refer to the  
25 Intellectual Property Rights section of the Security Services TC web page  
26 (<http://www.oasisopen.org/committees/security/>).

27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57

---

## Table of Contents

1 Introduction.....	3
1.1 Terminology.....	3
2 Use Case.....	4
3 Use Case Scenarios.....	5
3.1 Browser client – Browser requesting SAML assertion.....	5
3.2 Browser client – Application requesting SAML assertion.....	6
3.3 Non-Browser client – Client requesting SAML assertion.....	8
3.4 Non-Browser client – Application requesting SAML assertion.....	9
4 Solution Components.....	11
4.1 SAML Service.....	11
4.1.1 SOAP binding.....	11
4.1.1.1 Element <SubjectRequestArtifact>.....	11
4.1.1.2 Element <SubjectRequestAssertion>r.....	11
4.1.1.3 Element <ArtiFactResponse>.....	11
4.1.2 Non-HTTP binding.....	11
4.2 Authorization Data.....	11
5 Normalization.....	12
5.1 Introduction.....	12
5.2 Kerberos.....	12
5.3 Distributed Computing Environment (DCE).....	12
5.4 Windows.....	12
6 SAML-Defined Identifiers.....	13
6.1 Authentication Method Identifiers.....	13
6.1.1 Kerberos.....	13
6.2 NameIdentifier Format Identifiers.....	13
6.2.1 Kerberos Principal Name.....	13
6.2.2 DCE Principal Name.....	13
7 References.....	14
7.1 Normative References.....	14

---

# 58 1 Introduction

59 This document proposes a number of use cases and profile where SAML can be used in conjunction with  
60 Kerberos based technology. This includes:

- 61 • Kerberos v5 (as defined in RFC 1510)
- 62 • Kerberos GSS-API mechanism (as defined in RFC 1964)
- 63 • DCE (Distributed Computing Environment)
- 64 • Windows 2000/2003

65 In particular of interest are the last two technologies. Both of these leverage the ability of Kerberos to  
66 transport authorization data.

67 Note that whilst the use cases contained within this document build upon the Browser profiles defined else  
68 where in SAML 2.0, new components are introduced.

## 69 1.1 Terminology

70 The key words *must*, *must not*, *required*, *shall*, *shall not*, *should*, *should not*, *recommended*, *may*, and  
71 *optional* in this document are to be interpreted as described in IETF RFC 2119 [RFC2119].

## 2 Use Case

72

73 The high-level Kerberos use case has a user of a client workstation logging on to the local Key Distribution  
74 Centre (KDC), the Kerberos client software being present on the workstation. Following successful  
75 authentication the client credentials reside on the workstation.

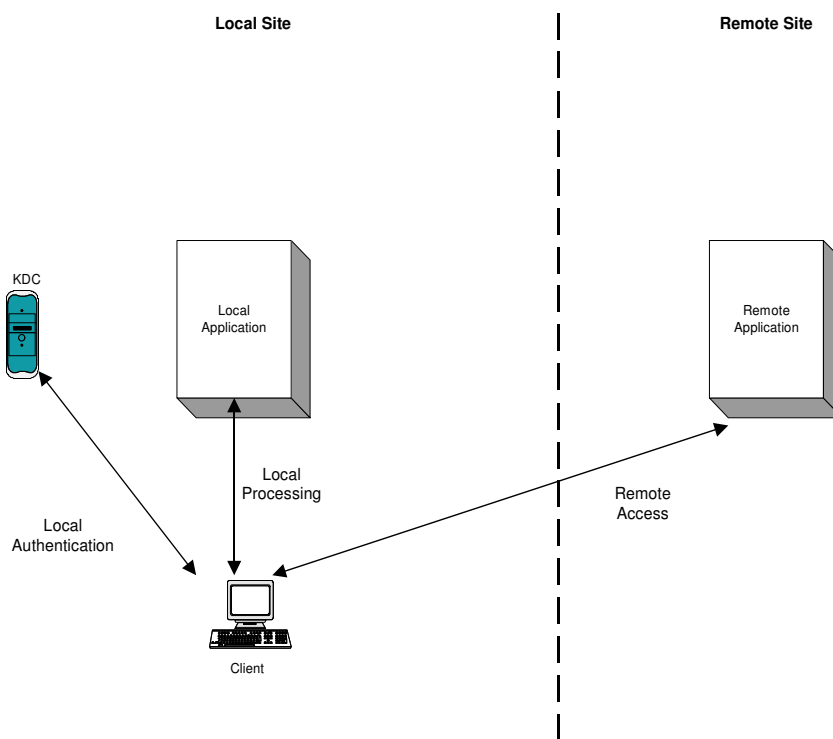
76 The workstation user then wishes to gain access to resources on a remote site in another management  
77 domain, so that:

- 78 • No further authentication is required
- 79 • Session attributes are transferred seamlessly over to the remote application so that it makes  
80 appropriate authorization decisions

81 Note that the local site may, or may not have a web server

82 Figure 1 illustrates the high level use case.

83



84

Figure 1– Overall Kerberos Use case

85

86 This use case can be further broken down into a number of use case scenarios, each involving a different  
87 mix of SAML components and protocols. The following sections articulates the use case scenarios, with  
88 further sections describing the components being used.

89

## 3 Use Case Scenarios

90

### 3.1 Browser client – Browser requesting SAML assertion

91

In this scenario the user on the client workstation has authenticated itself to the local KDC and obtained a TGT. The TGT is stored locally in the workstations credentials cache. The user may or may not interact with a local kerberosised application. At some point the user wishes to use a web browser on the client workstation to gain access on a remote web server in a different management domain.

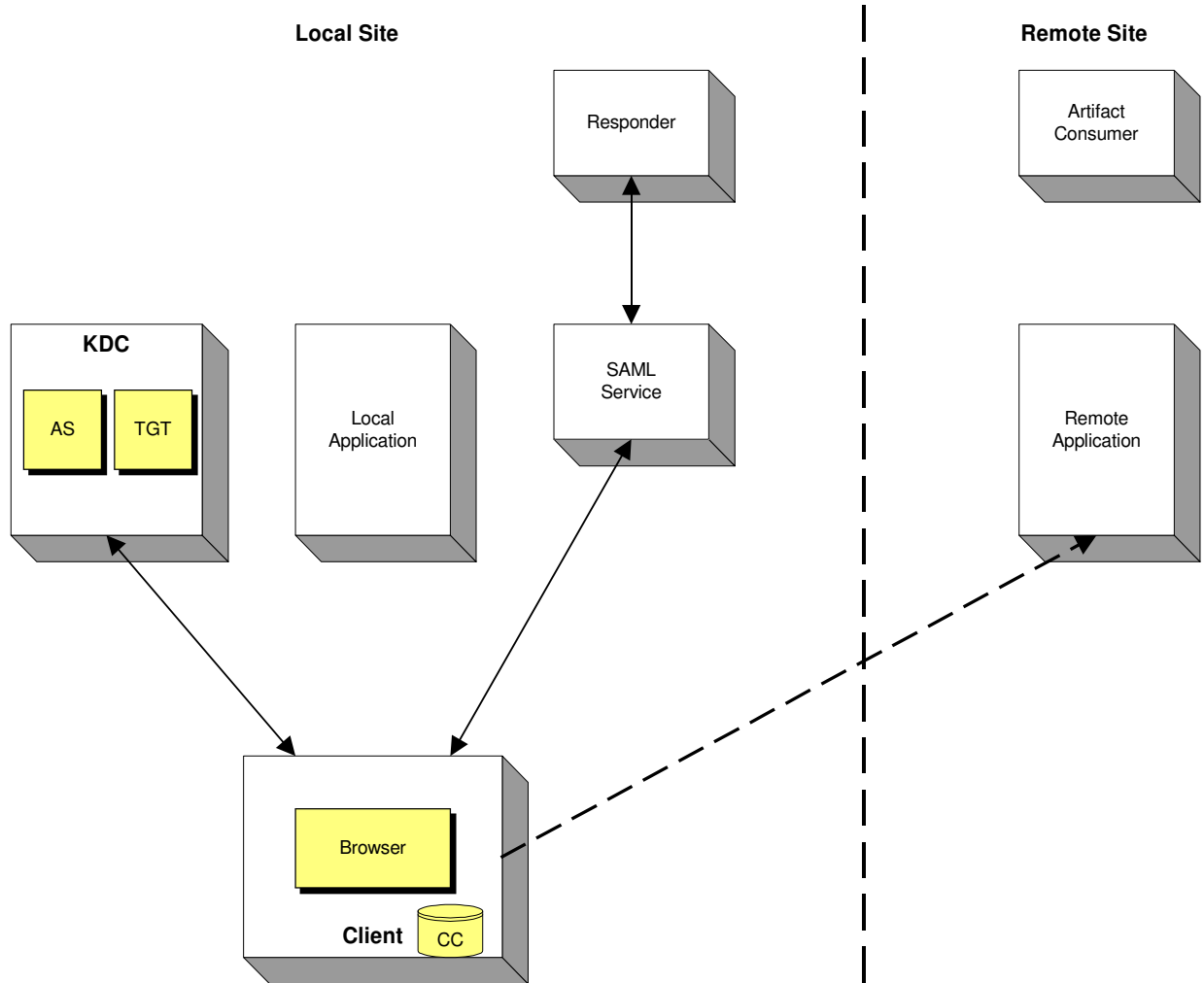
93

At some point the user wishes to use a web browser on the client workstation to gain access on a remote web server in a different management domain.

95

Figure 3 provides an overview of the components.

96



98

Figure 2– Browser Client – Browser Requesting SAML Assertion

99

100 The proposed processing is as follows:

101

1. The user on the workstation authenticates into the local domain/cell/realm, using client software resident on the workstation. Successful authentication results in an appropriate TGT being provided back to the workstation. The user can then access resources in the local site

102

2. The user then wishes to gain access to a remote site. In this scenario the remote site is a web server (or Portal). The first step in the process is to obtain a Service Ticket (ST) for a SAML Service within the local site. The workstation could initiate this either from a signed applet or an application

103

104

105

106

107 running on the workstation.

108 3. Having obtained the SAML Service ST, the application (or applet) then sends a request to the  
109 SAML service to generate a SAML Assertion. The SAML Service front ends the local site's SAML  
110 Responder. Using the SAML Service ST to gain access to the SAML Service means that the  
111 workstation has authenticated itself to the SAML Service and the user identity has also been passed  
112 to it. The request defines whether the response back to the workstation contains either an artifact  
113 (which references a SAML assertion) or a SAML assertion. Refer to Section 4.1.1 for the definition  
114 of the protocol.

115 4. The SAML Service obtains the user identification from the ST used to authenticate the connection  
116 between the workstation and the SAML Service. However if any PAC attributes exists within the ST  
117 then these are used to create AttributeStatements within the SAML assertion. Refer to Section 5.

118 5. The SAML Service responds back to the application/applet with either

- 119 • an artifact that references the generated SAML assertion
- 120 • the generated SAML Assertion within a SAML Response message

121 6. The Browser on the workstation then attempts to gain access to a resource on the remote web site.  
122 Either the Browser/POST or Browser/Artifact profile processing can then be used to process the  
123 request.

124 As for the Browser/POST and Browser/Artifact profiles confidentiality and message integrity MUST be  
125 maintained. If the artifact is passed to the remote site then it is RECOMMENDED that either SSL 3.0 or  
126 TLS 1.0 is used to protect the connection. If a SAML assertion is passed to the remote site then the  
127 SAML Response MUST be digitally signed following the guidelines given in [SAMLCore].

## 128 **3.2 Browser client – Application requesting SAML assertion**

129 In this use case scenario it's the local application (for instance a web server or Portal) that requests and  
130 issues the SAML Assertion. This scenario would be typical of an environment where Kerberos is used to  
131 perform authentication within the local site and a Portal is used as an intermediary to gain access to  
132 remote resources.

133 Figure 4 provides an overview of the architecture.

134

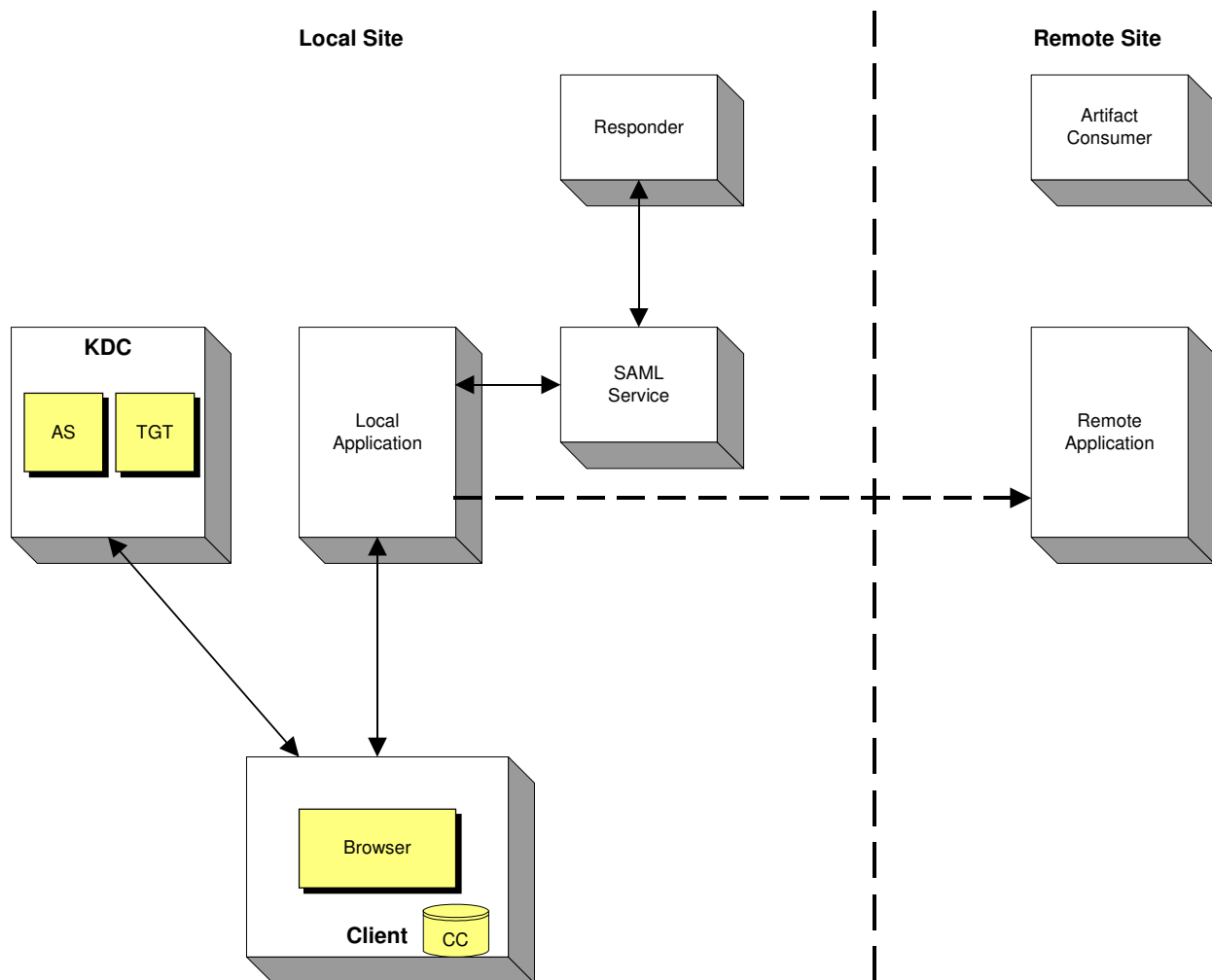


Figure 4– Browser Client – Application Requesting SAML Assertion

135

136

137 The proposed processing is as follows:

- 138 1. The user on the workstation authenticates into the local domain/cell/realm, using client software  
139 resident on the workstation. Successful authentication results in an appropriate TGT being  
140 provided back to the workstation. The user can then access resources in the local site
- 141 2. The user then wishes to gain access to remote resources (probably unbeknown to them) via the  
142 local Portal. The user's credentials on the workstation need to be passed to the Portal, this MUST  
143 be performed in a secure manner. It is RECOMMENDED that either SASL, TLS or SPNEGO are  
144 used to secure transport the user's credentials to the Portal.. The local Portal obtains a Service  
145 Ticket (ST) for the SAML Service using the user's credentials.
- 146 3. Having obtained the SAML Service ST, the portal sends a request to the SAML service to generate  
147 a SAML Assertion. The request defines whether the response back to the workstation contains  
148 either an artifact or an assertion. Refer to Section 4.1.1 for the definition of the protocol.
- 149 4. The SAML Service obtains the user identification from the ST used to authenticate the connection  
150 between the Portal and the SAML Service. However if any PAC attributes exists within the ST then  
151 these are used to create AttributeStatements within the SAML assertion. Refer to Section 5.
- 152 5. The SAML Service responds back to the Portal either
  - 153 • an artifact that references the generated SAML assertion
  - 154 • the generated SAML Assertion within a SAML Response

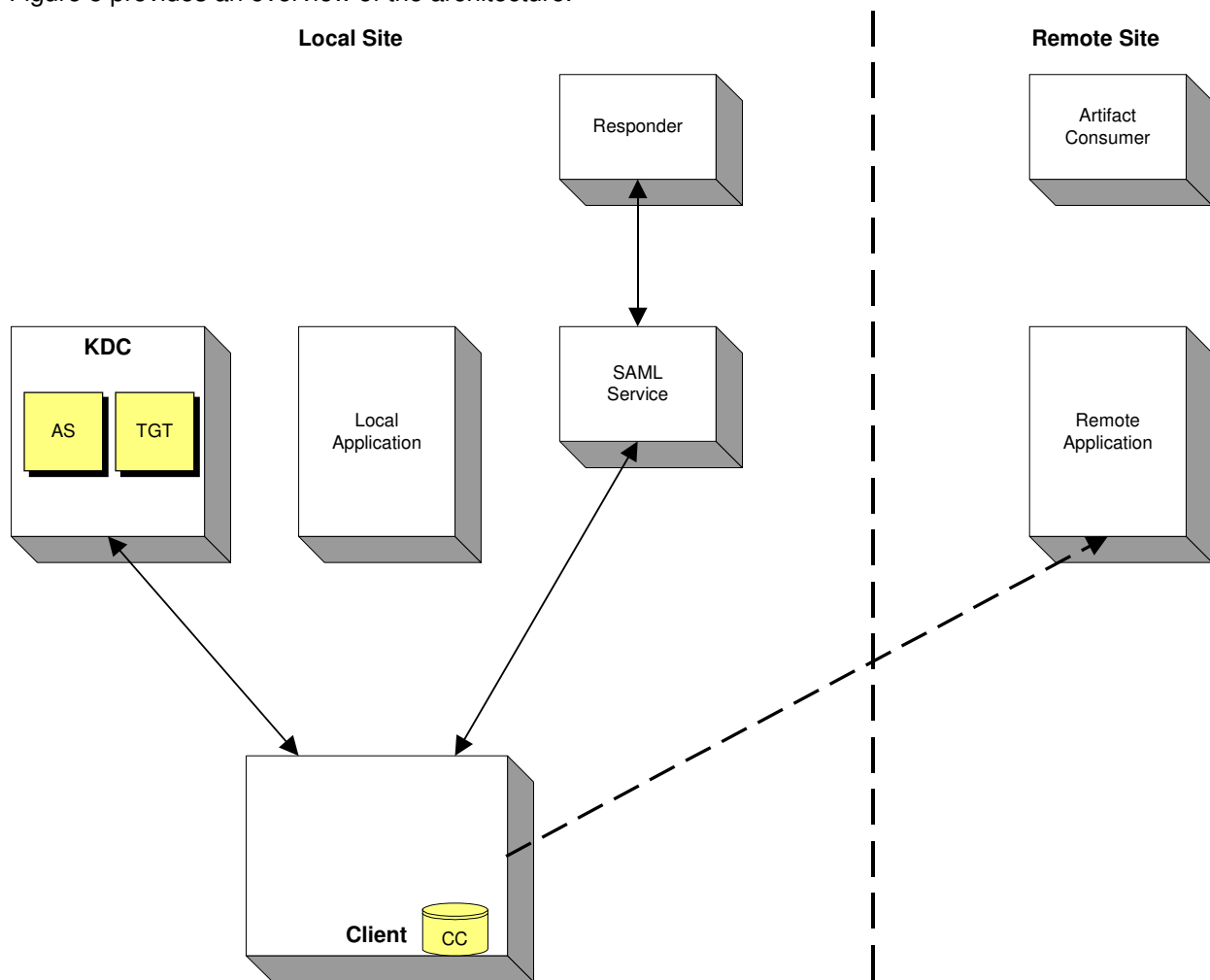
- 155 6. The Portal then gains access to the remote resource either by:
- 156 • Passing the artifact as a query variable in the HTTP URL
- 157 • Passing the artifact as a named custom Header variable
- 158 • The generated SAML Response as POST Data

159 For this last step confidentiality and message integrity MUST be maintained. If the artifact is passed to the  
 160 remote site then it is RECOMMENDED that either SSL 3.0 or TLS 1.0 is used to protect the connection. If  
 161 a SAML Response is passed to the remote site then the SAML Response MUST be digitally signed  
 162 following the guidelines given in [SAMLCore].

### 163 3.3 Non-Browser client – Client requesting SAML assertion

164 In this use case scenario it's the workstation that requests and issues the SAML Assertion, however the  
 165 workstation does not have any HTTP-based components present.

166 Figure 5 provides an overview of the architecture.



167 Figure 5– Non-Browser Client – Client Requesting SAML Assertion

168

169 The proposed processing is as follows:

- 170 1. The user on the workstation authenticates into the local domain/cell/realm, using client software
- 171 resident on the workstation. Successful authentication results in an appropriate TGT being
- 172 provided back to the workstation. The user can then access resources in the local site



- 173 2. The user then wishes to gain access to a remote site. In this scenario the remote site is an  
174 application that in non-HTTP based. The first step in the process is to obtain a Service Ticket (ST)  
175 for a SAML Service with the local site. The workstation would initiate an application running on the  
176 workstation to perform this.
- 177 3. Having obtained the SAML Service ST, the workstation then sends a request to the SAML service  
178 to generate a SAML Assertion. The SAML Service front ends the SAML Responder. Using the  
179 SAML Service ST to gain access to the SAML Service means that the workstation has  
180 authenticated itself to the SAML Service and the user identity has also been passed to it. The  
181 request defines whether the response back to the workstation contains either an artifact or an  
182 assertion. Refer to Section 4.1.2 for the definition of the protocol.
- 183 4. The SAML Service obtains the user identification from the ST used to authenticate the connection  
184 between the workstation and the SAML Service. However if any PAC attributes exists within the ST  
185 then these are used to create AttributeStatements within the SAML assertion. Refer to Section 5.
- 186 5. The SAML Service responds back to the application either
- 187 • an artifact that references the generated SAML assertion
  - 188 • the generated SAML Assertion within a SAML Response
- 189 6. The workstation then uses the either the Artifact or generation SAML Assertion to communicate  
190 with the remote application. The application protocol used to perform this is not defined.

191 For this last step confidentiality and message integrity MUST be maintained. If the artifact is passed to the  
192 remote site then it is RECOMMENDED that either SSL 3.0 or TLS 1.0 is used to protect the connection. If  
193 a SAML Response is passed to the remote site then the SAML Response MUST be digitally signed  
194 following the guidelines given in [SAMLCore].

### 195 3.4 Non-Browser client – Application requesting SAML assertion

196 In this use case scenario it's the application that requests and issues the SAML Assertion, however the  
197 workstation does not have any HTTP-based components present.

198 Figure 6 provides an overview of the architecture.

199

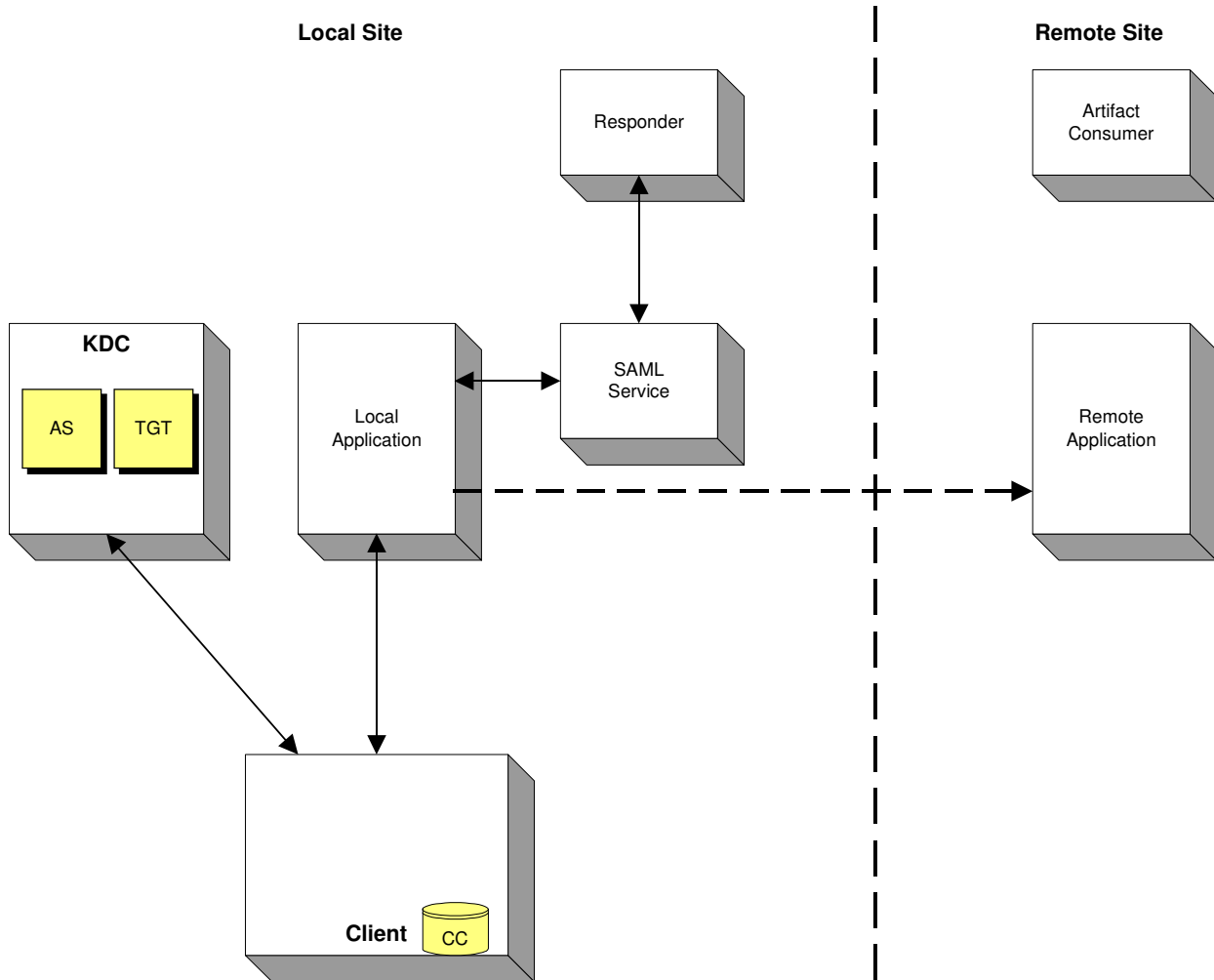
200 The proposed processing is as follows:

- 201 1. The user on the workstation authenticates into the local domain/cell/realm, using client software  
202 resident on the workstation. Successful authentication results in an appropriate TGT being  
203 provided back to the workstation. The user can then access resources in the local site
- 204 2. The application on the local web site then wishes to gain access to resources on the remote site.  
205 The user's credentials on the workstation need to be passed to the local application, this MUST be  
206 performed in a secure manner. It is RECOMMENDED that either SASL, TLS or SPNEGO are  
207 used to secure transport the user's credentials to the Portal.. The local Portal obtains a Service  
208 Ticket (ST) for the SAML Service using the user's credentials.
- 209 3. Having obtained the SAML Service ST, the local application sends a request to the SAML service to  
210 generate a SAML Assertion. The request defines whether the response back to the workstation  
211 contains either an artifact or an assertion. Refer to Section 4.1.2 for the definition of the protocol.
- 212 4. The SAML Service obtains the user identification from the ST used to authenticate the connection  
213 between the application and the SAML Service. However if any PAC attributes exists within the ST  
214 then these are used to create AttributeStatements within the SAML assertion. Refer to Section 5.
- 215 5. The SAML Service responds back to the local application either

  - 216 • an artifact that references the generated SAML assertion
  - 217 • the generated SAML Assertion

- 218 6. The local application then uses the either the Artifact or generated SAML Assertion to communicate  
219 with the remote application. The application protocol used to perform this is not defined.

220 For this last step confidentiality and message integrity MUST be maintained. If the artifact is passed to the  
 221 remote site then it is RECOMMENDED that either SSL 3.0 or TLS 1.0 is used to protect the connection. If  
 222 a SAML Response is passed to the remote site then the SAML Response MUST be digitally signed  
 223 following the guidelines given in [SAMLCore].  
 224



226 Figure 6– Non-Browser Client – Application Requesting SAML Assertion

---

## 227 4 Solution Components

### 228 4.1 SAML Service

229 The SAML Service is a front end to a SAML Responder and is a Kerberos-based service. The SAML  
230 Service can be co-located with the SAML Responder or a simple wrapper. In all cases the connection  
231 between the SAML Service and SAML Responder MUST be secure.

#### 232 4.1.1 SOAP binding

233 This uses the standard SOAP binding for the SAML protocol as defined in TBD. Two types of requests  
234 can be made on the SAML Service, to either request an assertion or an artifact (which refers to a SAML  
235 assertion). In both cases the SAML protocol `<SubjectQuery>` element is extended

##### 236 4.1.1.1 Element `<SubjectRequestArtifact>`

237 This query requests that an artifact is returned for the given subject. The following schema fragment  
238 defines the `<SubjectRequestArtifact>`

239

240 TBD

241

242 The SAML Service MUST validate that the identity supplied in the Service Tick matches that in the  
243 `<Subject>` element.

##### 244 4.1.1.2 Element `<SubjectRequestAssertion>`

245 This query requests that an assertion is returned for the given subject. The following schema fragment  
246 defines the `<SubjectRequestAssertion>`

247

248 TBD

249

250 The SAML Service MUST validate that the identity supplied in the Service Tick matches that in the  
251 `<Subject>` element.

##### 252 4.1.1.3 Element `<ArtifactResponse>`

253 When an Artifact is requested using the query `SubjectRequestArtifact`, the SAML response contains a  
254 `<ArtifactResponse>` element. The following schema fragment defines the `<ArtifactResponse>`  
255 element

256

257 TBD

#### 258 4.1.2 Non-HTTP binding

259 TBD

### 260 4.2 Authorization Data

261 TBD – refer to following section

---

## 262 5 Normalization

### 263 5.1 Introduction

264 TBD

### 265 5.2 Kerberos

266 TBD

267 Example of how details of a Kerberos principal are carried within a SAML Assertion.

268

```
269 <saml:Assertion xmlns:saml="urn:oasis:names:tc:SAML:1.0:assertion"  
270   MajorVersion="1"  
271   MinorVersion="1"  
272   AssertionID="P1YaAztP6UfswxAjax5TPxQ"  
273   Issuer="www.entegrity.com"  
274   IssueInstant="2002-06-19T17:05:37.795Z"  
275   <saml:Conditions NotBefore="2002-06-19T17:00:37.795Z"  
276     NotOnOrAfter="2002-06-19T17:10:37.795Z"/>  
277   <saml:AuthenticationStatement  
278     AuthenticationMethod="urn:ietf:rfc:1510"  
279     AuthenticationInstant="2002-06-19T17:05:17.706Z">  
280     <saml:Subject>  
281       <saml:NameIdentifier  
282         NameQualifier="http://www.entegrity.com/"  
283         Format="urn:oasis:names:tc:SAML:2.0:nameid-format:kerberos">  
284         talsop@CYBERSAFE.LTD.UK  
285       </saml:NameIdentifier>  
286       <saml:SubjectConfirmation>  
287         <saml:ConfirmationMethod>  
288           urn:oasis:names:tc:SAML:1.0:cm:artifact  
289         </saml:ConfirmationMethod>  
290         <saml:SubjectConfirmationData  
291           AAGZE1RNQJEFzYNGAGPjWvtDIRSZ4lWDqBphqA  
292         </saml:SubjectConfirmationData>  
293       </saml:SubjectConfirmation>  
294     </saml:Subject>  
295   </saml:AuthenticationStatement>  
296 </saml:Assertion>  
297
```

298

299

### 300 5.3 Distributed Computing Environment (DCE)

301 TBD. How DCE attributes are mapped into SAML Attribute Statements.

### 302 5.4 Windows

303 TBD. How Windows PAC attributes are mapped into SAML Attribute Statements.

---

304 **6 SAML-Defined Identifiers**

305 **6.1 Authentication Method Identifiers**

306 **6.1.1 Kerberos**

307 **URI:** urn:ietf:rfc:1510

308 The authentication was performed by means of the Kerberos protocol [**RFC 1510**], an instantiation of the  
309 Needham-Schroeder symmetric key authentication mechanism [**Needham78**]

310 **6.2 NameIdentifier Format Identifiers**

311 **6.2.1 Kerberos Principal Name**

312 **URI:** urn:oasis:names:tc:SAML:2.0:nameid-format:kerberos

313 Indicates that the content of the <NameIdentifier> element is in the form of a Kerberos principal  
314 name.

315 **6.2.2 DCE Principal Name**

316 **URI:** urn:oasis:names:tc:SAML:2.0:nameid-format:DCE

317 Indicates that the content of the <NameIdentifier> element is in the form of a DCE principal name.

---

318 **7 References**

319

320 **7.1 Normative References**

321 **[RFC2119]** S. Bradner, *Key words for use in RFCs to Indicate Requirement Levels*, IETF  
322 RFC 2119, March 1997, <http://www.ietf.org/rfc/rfc2119.txt>.

323

324 TBD

---

325

## A. Acknowledgments

326

The editors would like to acknowledge the contributions of the OASIS Security Services Technical Committee, whose voting members at the time of publication were:

327

328

- TBD

329

---

## B. Revision History

330

Rev	Date	By Whom	What
01	8 Jan 2004	John Hughes	Initial Version

331



---

## C. Notices

333 OASIS takes no position regarding the validity or scope of any intellectual property or other rights that  
334 might be claimed to pertain to the implementation or use of the technology described in this document or  
335 the extent to which any license under such rights might or might not be available; neither does it represent  
336 that it has made any effort to identify any such rights. Information on OASIS's procedures with respect to  
337 rights in OASIS specifications can be found at the OASIS website. Copies of claims of rights made  
338 available for publication and any assurances of licenses to be made available, or the result of an attempt  
339 made to obtain a general license or permission for the use of such proprietary rights by implementors or  
340 users of this specification, can be obtained from the OASIS Executive Director.

341 OASIS invites any interested party to bring to its attention any copyrights, patents or patent applications, or  
342 other proprietary rights which may cover technology that may be required to implement this specification.  
343 Please address the information to the OASIS Executive Director.

344 **Copyright © OASIS Open 2004. All Rights Reserved.**

345 This document and translations of it may be copied and furnished to others, and derivative works that  
346 comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and  
347 distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and  
348 this paragraph are included on all such copies and derivative works. However, this document itself does  
349 not be modified in any way, such as by removing the copyright notice or references to OASIS, except as  
350 needed for the purpose of developing OASIS specifications, in which case the procedures for copyrights  
351 defined in the OASIS Intellectual Property Rights document must be followed, or as required to translate it  
352 into languages other than English.

353 The limited permissions granted above are perpetual and will not be revoked by OASIS or its successors  
354 or assigns.

355 This document and the information contained herein is provided on an "AS IS" basis and OASIS  
356 DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY  
357 WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR  
358 ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.