



1 Liberty IGF Client Attribute Requirements 2 Markup Language (“CARML”) Specification

3 Draft Version 1.0-12

4 **Editors:**

5 Phil Hunt, Oracle Corporation

6 Prateek Mishra, Oracle Corporation

8 **Contributors:**

9 Shin Adachi, NTT

10 Conor Cahill, Intel

11 Makoto Hatakeyama, NEC Corporation

12 Paul Madsen, NTT

13 Colin Wallis, New Zealand

14 Peter Davis, Neustar

15 Eric Tiffany, Liberty Alliance

16 Sampo Kellomaki, Symlabs

17 Hubert Le Van Gong, SUN Microsystems

18 George Fletcher, AOL LLC

19 **Abstract:**

20 Client Attribute Requirements Markup (“CARML”) is a declarative format for expressing the requirements for
21 identity-related data of a service, application, device, web site, corporation or other entities. Requirements for
22 identity attributes, predicates, roles and search filters can be expressed using CARML. CARML also supports
23 privacy policies that prescribe constraints on the use of identity data.

24

Notice

25 This document has been prepared by Sponsors of the Liberty Alliance. Permission is hereby granted to use the
26 document solely for the purpose of implementing the Specification. No rights are granted to prepare derivative
27 works of this Specification. Entities seeking permission to reproduce portions of this document for other uses must
28 contact the Liberty Alliance to determine whether an appropriate license for such use is available.

29 Implementation or use of certain elements of this document may require licenses under third party intellectual
30 property rights, including without limitation, patent rights. The Sponsors of and any other contributors to the
31 Specification are not and shall not be held responsible in any manner for identifying or failing to identify any or all
32 such third party intellectual property rights. **This Specification is provided "AS IS," and no participant in the
33 Liberty Alliance makes any warranty of any kind, express or implied, including any implied warranties of
34 merchantability, non-infringement of third party intellectual property rights, and fitness for a particular
35 purpose.** Implementers of this Specification are advised to review the Liberty Alliance Project's website
36 (<http://www.projectliberty.org/>) for information concerning any Necessary Claims Disclosure Notices that have been
37 received by the Liberty Alliance Management Board.

38 Copyright © 2008 AOL LLC; British Telecommunications plc; Computer Associates International, Inc.; Drummond
39 Group, Inc.; Ericsson; France Télécom; Fugen Solutions, Inc.; GSA Office of Governmentwide Policy; Hewlett-
40 Packard Company; Intel Corporation; Luminance Consulting Services; Neustar, Inc.; New Zealand Government
41 State Services Commission; NEC Corporation; NHK (Japan Broadcasting Corporation) Science & Technical
42 Research Laboratories; Nippon Telegraph and Telephone Corporation; Oracle Corporation; Sun Microsystems, Inc.;
43 Symlabs, Inc.; Zenn New Media. All rights reserved.

44 Liberty Alliance Project
45 Licensing Administrator
46 c/o IEEE-ISTO
47 445 Hoes Lane
48 Piscataway, NJ 08855-1331, USA
49 info@projectliberty.org

50 Contents

51	1. Introduction.....	4
52	1.1. Example.....	5
53	1.2. Terminology.....	8
54	1.3. References.....	8
55	1.3.1. Normative References.....	8
56	1.3.2. Non-Normative References.....	8
57	1.4. Notation.....	8
58	2. Foundations.....	9
59	2.1. AttributeOrPredicateSuperType.....	9
60	2.2. CardinalityType.....	9
61	2.3. AttributeType.....	9
62	2.4. PredicateType.....	10
63	2.5. RefType.....	10
64	2.6. FilterRefType.....	10
65	2.7. FilterType.....	11
66	3. Client Attribute Requirements.....	13
67	3.1. DataDefs.....	14
68	3.1.1. ExternalDefsRef.....	15
69	3.1.2. Attributes.....	16
70	3.1.3. Predicates.....	16
71	3.1.4. Roles.....	16
72	3.1.5. Policies.....	16
73	3.2. Interaction.....	16
74	3.2.1. BaseInteractionType.....	17
75	3.2.2. AddInteraction.....	17
76	3.2.3. DeleteInteraction.....	18
77	3.2.4. ModifyInteraction.....	18
78	3.2.5. ReadInteraction.....	19
79	3.2.6. CompareInteraction.....	20
80	3.2.7. FindInteraction.....	21
81	3.2.8. SearchInteraction.....	22
82	Appendix A.	23
83	A.1. DataType URIs.....	23
84	A.2. Comparison Operators.....	24

1. Introduction

85

86 Client Attribute Requirements Markup (“CARML”) is a declarative format for expressing the requirements for
87 identity-related data of a service, application, device, web site, corporation or other entities. By identity-related data
88 we mean information associated with a *digital subject*. The requirements we have in mind primarily concern identity
89 data required by the entity, but support is also provided for expressing the update of identity data and for search of
90 digital subjects meeting certain criteria.

91 We will refer to the entity with whom the requirements are associated as the *client* or the *requestor*; we will refer to
92 the entity that satisfies the stated requirements as the *identity service* or the *responder*. No specific realization or
93 form factor is associated with these roles; in many situations a single entity may act as both a client or an identity
94 service.

95 Often, there are *policies* associated with the release of identity data by the identity service, including both *access*
96 policies and *privacy* policies. CARML does not discuss access policies or authentication methods, these have been
97 covered in other works, it deals only with declarations describing *interactions* concerning identity data between the
98 requestor and the responder, as well as privacy policies of the client.

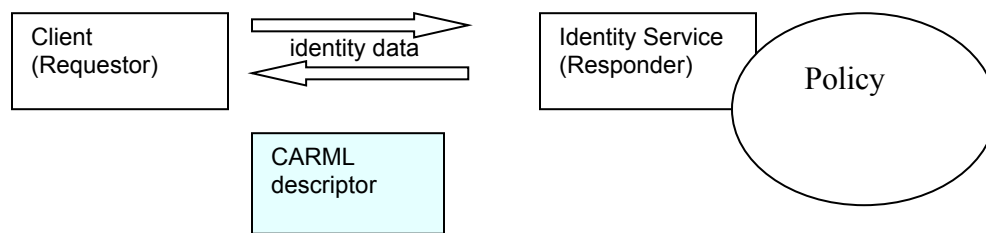


Figure 1

100 No particular protocol binding or message format for the identity service is defined in this specification. The exact
101 format used to identify a digital subject is also left to particular implementations. Depending upon the business
102 context we assume that many different protocols and message formats may utilize the CARML specification. This
103 could take the form of defining specific profiles or bindings that use a CARML elements and provide appropriate
104 access to identity data.

105 We do assume that the identity service supports some of the following operations, each of which is expressed by one
106 or more CARML interaction elements:

- 107 1. Given a digital subject, retrieve or read attributes, roles or predicate values associated with the subject
- 108 2. Given a digital subject, determine if certain predicates, roles, or attribute values are associated with it.
- 109 3. Given attribute values or roles, retrieve digital subjects that possess those values or roles
- 110 4. Given a set of attribute values or roles, request the creation of a digital subject associated with these values
- 111 5. Given a digital subject, request the update of attributes or roles associated with the digital subject
- 112 6. Given a digital subject, request that the digital subject be deleted.

113 These interactions are designed to be flexible enough to meet the types of identity processing requirements of a
114 variety of applications that can be mapped and profiled for a number of information exchange protocols such as
115 LDAP, WS-Trust, ID-WSF, etc. Because the intent of CARML is to allow an application to declare its definition of
116 identity data schema and the operations against that schema, it is important to keep in mind that these interaction
117 declarations are always from the perspective of the requester and may not correspond directly to the steps carried out
118 by the identity service.

119 For example, in a distributed multi-application environment, a single application's "AddInteraction", a request to add
120 a new record, should be considered solely as a request for a certain type of service. The identity service may respond
121 to the request in many different ways – adding a new record in persistent store, or just modifying an existing identity
122 record to add information specific to an application to that record. Likewise, for a DeleteInteraction, it will be policy

123 and context information within the identity service and other infra-structure that determine the actions carried out
124 when the deletion of a digital subject is requested (e.g. delete from persistent store, log and archive request, set flag
125 indicating delete requested).

126 The means by which a CARML descriptor is defined or created is outside the scope of this specification. Depending
127 upon business-context, such a descriptor may be created via automatic or manual negotiation or provided
128 unilaterally by the client or the identity service.

129 1.1. Example

```
130 [a1] - <carml:ClientAttrReq AppName="CARML Example" Description="Demonstrates  
131     features of CARML Schema" xmlns:carml="urn:igf:client:0.9:carml"  
132     xmlns:wsp="http://www.w3.org/ns/ws-policy"  
133     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
134     xsi:schemaLocation="urn:igf:client:0.9:carml igf-carml-09.xsd">  
135 [a2]  
136 [a3] - <DataDefs>  
137 [a4]  
138 [a5] - <Attributes>  
139 [a6]   <Attribute Cardinality="single" DataType="string" DisplayName="Surname"  
140     Name="sn" />  
141 [a7]   <Attribute Cardinality="single" DataType="string" Description="One or  
142     more names that are considered given names. The first name should be the preferred  
143     name." DisplayName="Given names" Name="givenname" />  
144 [a8]   <Attribute Cardinality="single" DataType="urn:oasis:names:tc:xacml:1.0:data-  
145     type:rfc822Name" DisplayName="E-Mail" Name="mail" />  
146 [a9]   <Attribute Cardinality="single" DataType="string" DisplayName="Telephone"  
147     Name="telephone" />  
148 [a10]  <Attribute Cardinality="single" DataType="string" DisplayName="Last 4  
149     Digits SSN" Name="Last4SSN" />  
150 [a11]  </Attributes>  
151 [a12] - <Predicates>  
152 [a13]  <Predicate Description="For the jurisdiction of the user, a determination that  
153     the subject can travel alone." DisplayName="Adult" Name="IsAdult" />  
154 [a14]  <Predicate Description="A resident of the EU" DisplayName="EU Resident"  
155     Name="IsEUResident" />  
156 [a15]  </Predicates>  
157 [a16] - <Roles>  
158 [a17]  <Role Description="Able to book business class tickets"  
159     DisplayName="Business Class FLYer" Name="BusinessClassFlyer" />  
160 [a18]  <Role Description="The passenger's account is active." DisplayName="Account  
161     active" Name="IsActive" />  
162 [a19]  <Role Description="Person is an employee" Name="Employee" />  
163 [a20]  <Role Description="Person is a contractor" Name="Contractor" />  
164 [a21]  </Roles>  
165 [a22]  <Policies>  
166 [a23]  <wsp:Policy Name="http://tempuri.org/" />  
167 [a24] </Policies>
```

```
168 [a25] </DataDefs>
169 [a26]
```

170 The <DataDefs> element (lines [a03]-,[a26]) defines the attributes, roles, predicates, and privacy policies of
171 interest in the <ClientAttrReq>. Attributes, roles and predicates are the foundational components out of
172 which interactions are built. This document does provide details of privacy policies, these are described in
173 [CARMLProf].

174 Lines [a27] – [a74] defines a number of different <XXXXXXInteraction> elements, each of which references
175 some of the previously defined attribute, role and predicate elements. Multiple interaction elements of each type
176 may be included within a single <ClientAttrReq> element.

```
177 [a27] <ReadInteraction Description="" Name="ReadProfile">
178 [a28]   <wsp:Policy Name="http://tempuri.org" />
179 [a29]   <AttributeRef Ref="#mail" />
180 [a30]   <AttributeRef Ref="#sn" />
181 [a31]   <AttributeRef Ref="#givenname" />
182 [a32]   <AttributeRef Ref="#telephone" Optional="true" />
183 [a33]   <PredicateRef Ref="#IsAdult" Optional="true" />
184 [a34]   <PredicateRef Ref="#IsEUResident" />
185 [a35]   <RoleRef Ref="#BusinessClassFlyer" />
186 [a36] </ReadInteraction>
187 [a37]
188 [a38] - <FindInteraction Description="Locate user for authentication purposes."
189 [a39]   Name="LocateUser">
190 [a39]   <wsp:Policy Name="http://tempuri.org" />
191 [a40] - <Filter Match="all">
192 [a41]   <AttrRefFilter Ref="#mail" PrimaryKey="true" />
193 [a42] - <Filter Match="any">
194 [a43]   <RoleRefFilter Ref="#Employee" />
195 [a44]   <RoleRefFilter Ref="#Contractor" />
196 [a45] </Filter>
197 [a46] </Filter>
198 [a47] </FindInteraction>
199 [a48]
200 [a49] - <SearchInteraction Name="SearchLastName" Description="Returns potential
201 [a50]   matches for a given surname">
202 [a50]   <AttributeRef Ref="#mail" />
203 [a51]   <AttributeRef Ref="#sn" />
204 [a52] - <Filter Match="all">
205 [a53]   <AttrRefFilter Ref="#sn" />
206 [a54]   <RoleRefFilter Ref="#IsActive" />
207 [a55] </Filter>
208 [a56] </SearchInteraction>
209 [a57]
```

```
210 [a58] - <CompareInteraction Name="VerifyIdentity" Description="Used to verify
211         information provided by user">
212 [a59] - <Filter Match="all">
213 [a60]   <AttrRefFilter Ref="#Last4SSN" Operator="endswith" />
214 [a61]   <AttrRefFilter Ref="#mail" Operator="equals" />
215 [a62]   </Filter>
216 [a63]   </CompareInteraction>
217 [a64]
218 [a65] - <ModifyInteraction Name="UpdateTelephoneNumber">
219 [a66]   <AttributeRef Ref="#telephone" />
220 [a67]   </ModifyInteraction>
221 [a68]
222 [a69] - <AddInteraction Name="AddNewUser">
223 [a70]   <AttributeRef Ref="#mail" />
224 [a71]   <AttributeRef Ref="#sn" />
225 [a72]   <AttributeRef Ref="#givenname" />
226 [a73]   <AttributeRef Ref="#telephone" Optional="true" />
227 [a74]   <RoleRef Ref="#Employee" Optional="true" />
228 [a75]   <RoleRef Ref="#Contractor" Optional="true" />
229 [a76]   </AddInteraction>
230 [a77]
231 [a78]   <DeleteInteraction Name="UnRegisterUser" Description="User cannot use this
232         service goingforward" />
233 [a79] </carml:ClientAttrReq>
234 [a80]
```

235 The contents of the <ReadInteraction> element ([a27]-[a36]) indicate that the service requires certain
236 attribute, predicate and role values, with some declared optional.

237 The <FindInteraction> element ([a38]-[a47]) indicates that the service plans to search for a digital subject
238 based upon their e-mail address with the additional constraint that the subject possess one of employee or contractor
239 roles.

240 The <SearchInteraction> element ([a49]-[a56]) indicates that the service plans to search for digital subjects
241 based upon social security number and the IsActive role; in addition to retrieving the digital subject, it also requires
242 the social security number and e-mail address to be reported.

243 The <CompareInteraction> element ([a58]-[a63]) indicates that the service plans to check the social security
244 number (last four digits) and e-mail address of certain digital subjects.

245 The <ModifyInteraction> element ([a65]-[a66]) indicates that the service plans to provide the telephone
246 number of certain digital subjects.

247 The <AddInteraction> element ([a69]-[a76]) indicates that the service may register or create new digital
248 subjects with certain attributes and roles; some of this information is marked as optional and may not be provided in
249 the request.

250 The <DeleteInteraction> element ([a78]-[a80]) indicates that the service may request deletion or suspension
251 of certain digital subjects.

252 1.2. Terminology

253 Conventional XML namespace prefixes are used throughout the listings in this specification to stand for their
254 respective namespaces, whether or not a namespace declaration is present in the example:

Prefix	XML Namespace	Comments
carml:	urn:lap:names:1.0:igf:carml	Namespace defined in this specification
pri:	urn:lap:names:1.0:igf:pri	Privacy assertions namespace
wsp:	http://www.w3.org/ns/ws-policy	Web Services Policy namespace
xs:	http://www.w3.org/2001/XMLSchema	This namespace is defined in the W3C XML Schema specification [Schema1-2]. In schema listings, this is the default namespace and no prefix is shown. For clarity, the prefix is generally shown in specification text when XML Schema-related constructs are mentioned.
xsi:	http://www.w3.org/2001/XMLSchema-instance	This namespace is defined in the W3C XML Schema specification [Schema1-2] for schema-related markup that appears in XML instances.

255 1.3. References

256 1.3.1. Normative References

- 257 **[RFC2119]** S. Bradner, "Key words for use in RFCs to Indicate Requirement Levels," RFC 2119, The
258 Internet Engineering Task Force (March 1997). <http://www.ietf.org/rfc/rfc2119.txt>
- 259 **[WS-Policy]** Web Services Polict 1.5 – Framework, October 2007. <http://www.w3.org/TR/2004/REC-xmlschema-1-20041028/>
- 261 **[PrivConstr]** Madsen, Paul, "Liberty IGF Privacy Constraints Specification," Draft Version 1.0-04, Liberty
262 Alliance Project (21 June 2008). <http://www.projectliberty.org/specs>
- 263 **[CARMLProf]** Hunt, Phil "CARML Profile of the Liberty IGF Privacy Constraints Specification," Draft
264 Version 1.0-02, Liberty Alliance Project (21 June 2008). <http://www.projectliberty.org/specs>
- 265 **[Schema1-2]** Thompson, Henry S., Beech, David, Maloney, Murray, Mendelsohn, Noah, eds. (28 October
266 2004). "XML Schema Part 1: Structures Second Edition," Recommendation, World Wide
267 Web Consortium, <http://www.w3.org/TR/xmlschema-1/>

268 1.3.2. Non-Normative References

269 None.

270 1.4. Notation

271 This specification contains schema conforming to W3C XML Schema and normative text to describe the syntax and
272 semantics of XML-encoded policy statements.

273 The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD
274 NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this specification are to be interpreted as described in
275 IETF RFC 2119 **[RFC2119]**

276 *"they MUST only be used where it is actually required for interoperation or to limit behavior which has
277 potential for causing harm (e.g., limiting retransmissions)"*

278 These keywords are thus capitalized when used to unambiguously specify requirements over protocol and
279 application features and behavior that affect the interoperability and security of implementations. When these words
280 are not capitalized, they are meant in their natural-language sense.

281 2. Foundations

282 An identity service may associate name-value pairs with a digital subject; we refer to these as *attribute* names and
283 values. Given an attribute name, there maybe zero or more values associated with it.

284 An identity service may associate named predicates or judgments with a digital subject; we will refer to these as
285 *predicates* and these always evaluate to a boolean value. A special type of predicate is a *group* or *role* associated
286 with a subject. In certain interactions, it is possible to enumerate the roles associated with a digital subject, query
287 for all the digital subjects associated with a role or update roles associated with a digital subject. It is important to
288 note that no particular implementation model is mandated for roles.

289 An identity service may provide means of searching or finding sets of subjects based on attribute values, predicates
290 or roles; we will refer to these constructs as *search filters*.

291 2.1. AttributeOrPredicateSuperType

```
292 <complexType name="AttributeOrPredicateSuperType" abstract="true">  
293   <attribute name="Name" type="ID" use="required"/>  
294   <attribute name="DisplayName" type="string" use="optional"/>  
295   <attribute name="Description" type="string" use="optional"/>  
296   <anyAttribute namespace="##other" processContents="lax"/>  
297 </complexType>
```

298 Name

299 The name of the attribute, predicate or filter

300 DisplayName

301 Human-friendly name which might be displayed on a form or on-screen

302 Description

303 String description or definition of the attribute, predicate or filter

304 2.2. CardinalityType

```
305 <simpleType name="CardinalityType">  
306   <restriction base="string">  
307     <enumeration value="zero"/>  
308     <enumeration value="single"/>  
309     <enumeration value="multiple"/>  
310   </restriction>  
311 </simpleType>
```

312 2.3. AttributeType

313 *AttributeType* defines a single named attribute which may have zero or more associated values. All of the
314 values must be of a single type. A client may request the value of an attribute from an identity service or provide it
315 to an identity service.

```
316 <complexType name="AttributeType">
317   <complexContent>
318     <extension base="carml:AttributeOrPredicateSuperType">
319       <attribute name="Cardinality" type="carml:CardinalityType" use="optional"/>
320       <attribute name="DataType" type="anyURI" use="optional" default="string"/>
321     </extension>
322   </complexContent>
323 </complexType>
```

324 **Cardinality**

325 Whether the attribute is zero, single or multi-valued

326 **DataType**

327 The data type of the value(s) associated with the attribute. Appendix A.1 lists datatypes that MUST be
328 supported by a conformant identity service.

329 **2.4. PredicateType**

330 PredicateType describes a single named predicate, a boolean valued decision or judgment, provided by an
331 identity service to a client.

```
332 <complexType name="PredicateType">
333   <complexContent>
334     <extension base="carml:AttributeOrPredicateSuperType"/>
335   </complexContent>
336 </complexType>
```

337 **2.5. RefType**

338 RefType defines a utility type that combines reference to a privacy policy with reference to a <carml:Attribute>,
339 <carml:Role> or <carml:Predicate>.

```
340 <complexType name="RefType">
341   <attribute name="Ref" type="anyURI" use="required"/>
342   <attribute name="PolicyRef" type="anyURI" use="optional"/>
343   <attribute name="Optional" type="boolean" use="optional" default="false"/>
344   <attribute name="Description" type="string" use="optional"/>
345 </complexType>
```

346 **Ref**

347 URI of local or external <Attribute>, <Predicate> or <Role> element

348 **PolicyRef**

349 URI of local or external privacy policy

350 **Optional**

351 Whether the referenced entity MUST be provided by the requestor or the responder

352 **2.6. FilterRefType**

353 FilterRefType extends RefType with additional attributes useful in defining a filter.

```
354 <complexType name="FilterRefType">
355   <complexContent>
356     <extension base="carml:RefType">
357       <attribute name="Cardinality" type="carml:CardinalityType" use="optional"
358         default="single"/>
359       <attribute name="PrimaryKey" type="boolean" default="false"/>
360       <attribute name="Operator" default="equals">
361         <simpleType>
362           <restriction base="string">
363             <enumeration value="contains"/>
364             <enumeration value="doesnotcontain"/>
365             <enumeration value="beginswith"/>
366             <enumeration value="endswith"/>
367             <enumeration value="equals"/>
368             <enumeration value="notequals"/>
369             <enumeration value="gt"/>
370             <enumeration value="lt"/>
371             <enumeration value="geq"/>
372             <enumeration value="leq"/>
373           </restriction>
374         </simpleType>
375       </attribute>
376       <attribute name="Name" type="ID" use="optional"/>
377     </extension>
378   </complexContent>
379 </complexType>
```

380 **Cardinality**

381 Whether the requestor provides single or multiple values

382 **DataType**

383 The data type of the value(s) provided by the requestor

384 **PrimaryKey**

385 Whether the client or requestor views the attribute as a key or index

386 **Operator**

387 Allows the requestor to describe the operation to be applied by the identity service to the values provided by the
388 requestor. Details of the operation are given in Appendix A.2

389 **2.7. FilterType**

390 FilterType defines the means by which a requestor proposes to identify digital subjects. Digital subjects may be
391 identified using attributes, predicates or roles.

```
392 <complexType name="FilterType">
393   <choice maxOccurs="unbounded">
394     <element name="AttrRefFilter" type="carml:FilterRefType" minOccurs="0"
395     maxOccurs="unbounded"/>
396     <element name="RoleRefFilter" type="carml:RefType" minOccurs="0"
397     maxOccurs="unbounded"/>
398     <element name="PredRefFilter" type="carml:RefType" minOccurs="0"
399     maxOccurs="unbounded"/>
400     <element name="Filter" type="carml:FilterType" minOccurs="0" maxOccurs="unbounded"/>
401   </choice>
402   <attribute name="Match" default="all">
403     <simpleType>
404       restriction base="string">
405         <enumeration value="any"/>
406         <enumeration value="all"/>
407       </restriction>
408     </simpleType>
409   </attribute>
410   <attribute name="Description" use="optional"/>
411 </complexType>
```

412 **AttRefFilter**

413 The Ref attribute MUST reference a <carml:Attribute> element using a URI.

414 **RoleRefFilter**

415 The Ref attribute MUST reference a <carml:Role> element using a URI.

416 **PredRefFilter**

417 The Ref attribute MUST reference a <carml:Predicate> element using a URI.

418 **Filter**

419 Allows for additional nested filter elements to be included within a single element of type <FilterType>

420 **Match**

421 Describes whether the elements found within an element of type <FilterType> should be evaluated as a
422 conjunction (“all”) or disjunction (“any”).

423 3. Client Attribute Requirements

```
424 <element name="ClientAttrReq">
425   <!-- root element for a CARML declaration -->
426   <complexType>
427     <sequence>
428       <element name="DataDefs">
429         ...
430       </element>
431       <choice minOccurs="0" maxOccurs="unbounded">
432         <element name="AddInteraction" maxOccurs="unbounded">
433           ...
434         </element>
435         <element name="DeleteInteraction" type="carml:BaseInteractionType"
436 maxOccurs="unbounded"/>
437         ...
438       </element>
439       <element name="ReadInteraction" maxOccurs="unbounded">
440         ...
441       </element>
442         <element name="ModifyInteraction" maxOccurs="unbounded">
443           ...
444         </element>
445         <element name="CompareInteraction" minOccurs="0" maxOccurs="unbounded">
446           ...
447         </element>
448         <element name="FindInteraction" maxOccurs="unbounded">
449           ...
450         </element>
451         <element name="SearchInteraction" maxOccurs="unbounded">
452           ...
453         </element>
454       </choice>
455   <!-- Application policy -->
456   <choice minOccurs="0" maxOccurs="unbounded">
457     <element ref="wsp:Policy"/>
458     <element ref="wsp:PolicyReference"/>
459   </choice>
460 </sequence>
461   <attribute name="AppName" type="string" use="required"/>
462   <attribute name="Description" type="string" use="optional"/>
463   <attribute name="CarmlURI" type="anyURI" use="optional"/>
464 </complexType>
465 </element>
```

466 <ClientAttrReq> is the root element that captures the client attribute requirements of a specific entity. The
467 requirements are captured by a set of zero or more interaction elements. Interaction elements include
468 <AddInteraction>, <ReadInteraction>, <ModifyInteraction>, <UpdateInteraction>,
469 <CompareInteraction>, <FindInteraction> and <SearchInteraction> elements. Each of these
470 elements references attributes, predicates, roles and policies declared in the <DataDefs> element.

471 In some cases, only the <DataDefs> element may be present; this corresponds to a client or applications group
472 publishing a list of standard or preferred attributes, predicates, roles and policies. Such a declaration might be used
473 to publish a standard set of names and types for reference by other <ClientAttrReq> elements.

474 [PrivConstr] defines privacy policy assertions that express privacy constraints over the use of identity data. [WS-
475 Policy] provides a general framework for expressing composite policies built out of atomic assertions.

476 The <wsp:Policy> or <wsp:PolicyReference> element carries policy assertions based on WS-Policy with
477 atomic assertions drawn only from [PrivConstr]. These policies apply to all of the interactions defined within the
478 <ClientAttrReq> element.

479 **AppName**

480 String name associated with <ClientAttrReq> element

481 **CarmlURI**

482 URI associated with the <ClientAttrReq> element

483 **3.1. DataDefs**

484 The <DataDefs> element defines all the different entities that might be used via reference by one or more
485 <Interaction> elements found within the <ClientAttrReq> element.

```
486 <element name="DataDefs">
487   <complexType>
488     <sequence>
489       <element name="ExternalDataDefsRef" minOccurs="0" maxOccurs="unbounded">
490         <complexType>
491           <attribute name="CarmlURI" type="anyURI" use="required"/>
492           <attribute name="AppName" type="string" use="optional"/>
493           <attribute name="ProcessNestedDefinitions" type="boolean" default="true"/>
494           <anyAttribute namespace="##any" processContents="lax"/>
495         </complexType>
496       </element>
497       <element name="Attributes">
498         <complexType>
499           <sequence>
500             <element name="Attribute" type="carml:AttributeType" minOccurs="0"
501 maxOccurs="unbounded"/>
502           </sequence>
503         </complexType>
504       </element>
505       <element name="Predicates">
506         <complexType>
507           <sequence>
508             <element name="Predicate" type="carml:PredicateType" minOccurs="0"
509 maxOccurs="unbounded"/>
510           </sequence>
511         </complexType>
512       </element>
513       <element name="Roles">
514         <complexType>
515           <sequence>
516             <element name="Role" type="carml:PredicateType" minOccurs="0"
517 maxOccurs="unbounded"/>
518           </sequence>
519         </complexType>
520       </element>
521       <element name="Policies">
522         <complexType>
523           <sequence>
524             <element ref="wsp:Policy" minOccurs="0" maxOccurs="unbounded"/>
525           </sequence>
526         </complexType>
527       </element>
528     </sequence>
529   </complexType>
530 </element>
```

531 3.1.1. ExternalDefsRef

532 The <ExternalDefsRef> element supports reference to attributes, roles, predicates and policies that may be
533 defined in other <ClientAttrReq> elements.

534 **CarmlURI**

535 URI of referenced <ClientAttrReq> element

536 **AppName**

537 Optional name of the referenced <ClientAttrReq> element

538 **ProcessNestedDefinitions**

539 Whether the <ExternalDefsRef> element of the referenced <ClientAttrReq> element is to be
540 recursively included in scope.

541

542 **3.1.2. Attributes**

543 The <Attributes> element defines all of the the <Attribute> elements available to be referenced by
544 <Interaction> elements.

545 **3.1.3. Predicates**

546 The <Predicates> element all of the <Predicate> elements available to be referenced by <Interaction>
547 elements.

548 **3.1.4. Roles**

549 The <Roles> element defines all of the <Role> elements available to be referenced by <Interaction>
550 elements.

551 **3.1.5. Policies**

552 [CARMLProf] defines privacy policy assertions that express privacy constraints for identity data. The <Policies>
553 element carries policy assertions based on WS-Policy [WS-Policy] with atomic assertions drawn only from
554 [PrivConstr]. These assertions may be referenced by <Interaction> elements.

555 **3.2. Interaction**

556 An interaction represents a single exchange between a client and an identity service. Some interactions assume that
557 the client or requestor will provide information about a digital subject (the target identity) whereas other interactions
558 require the identity service to find or create a digital subject.

559 <ReadInteraction>, <ModifyInteraction>, <DeleteInteraction>, <CompareInteraction>
560 require the requestor to provide information about the target identity.

561 <AddInteraction> has the requestor providing information about a new digital subject; the identity service
562 then returns a digital subject descriptor to the requestor.

563 <SearchInteraction> and <FindInteraction> have the requestor describing digital subjects using
564 roles, predicates and attributes; the identity service returns digital subject handles for matching subjects.

565 There are three components in the overall structure of an interaction element:

- 566 1. Information about the client's intent , whether identity information is being read or updated or whether
567 digital subjects are to be retrieved based on certain criteria.
- 568 2. The attributes, roles, predicates and policies relevant to the interaction.
- 569 3. Additional privacy policies that constrain the exchange, specific to the interaction.

570 3.2.1. BaseInteractionType

```
571 <complexType name="BaseInteractionType" abstract="true">
572   <sequence>
573     <!-- Holds interaction policies -->
574     <choice minOccurs="0" maxOccurs="unbounded">
575       <element ref="wsp:Policy"/>
576       <element ref="wsp:PolicyReference"/>
577     </choice>
578   </sequence>
579   <attribute name="Name" type="ID" use="required"/>
580   <attribute name="Description" use="optional"/>
581 </complexType>
```

582 3.2.2. AddInteraction

```
583 <element name="AddInteraction" maxOccurs="unbounded">
584   <complexType>
585     <complexContent>
586       <extension base="carml:BaseInteractionType">
587         <sequence>
588           <element name="AttributeRef" type="carml:RefType" minOccurs="0"
589             maxOccurs="unbounded"/>
590           <element name="RoleRef" type="carml:RefType" minOccurs="0" maxOccurs="unbounded"/>
591         </sequence>
592       </extension>
593     </complexContent>
594   </complexType>
595 </element>
```

596 The <AttributeRef> element has two attributes: the Ref attribute MUST reference an <Attribute> element
597 using a URI; the PolicyRef MUST reference a policy element using a URI.

598 The <RoleRef> element has two attributes: the Ref attribute MUST reference an <Role> element using a URI;
599 the PolicyRef MUST reference a policy element using a URI.

600 The identity service MUST return an identifier representing a digital subject distinct from any previously provided
601 to the requestor or an error message indicating that the identity service is unable to process the request.

602 The identity service MUST receive values for all <Attributes> or <Roles> that have the Optional attribute set
603 to false; otherwise, it MUST return an error indication to the client.

604 If the identity service cannot process the request due to the subject being known prior to the request, it MUST return
605 an error indication to the client.

606 If the identity service cannot process the request due to use policy incompatibility, it MUST return an error
607 indication to the client.

608 If the identity service cannot provide requested information due to lack of user consent, it MUST return an error
609 indication to the client.

610 If the identity service cannot process the information provided for other reasons, it MUST return an error indication
611 to the client.

612 3.2.3. DeleteInteraction

```
613 <element name="DeleteInteraction" type="carml:BaseInteractionType" maxOccurs="unbounded"/>
```

614 The identity service MUST return an indication of whether the service successfully received the request to delete the
615 digital subject, or, whether the operation failed to complete. There is no implication that the digital subject has been
616 expunged from persistent store; only that future retrieval or update requests for the specified digital subject
617 SHOULD fail.

618 If the identity service cannot process the request due to the subject not being known prior to the request, it MUST
619 return an error indication to the client.

620 If the identity service cannot process the request due to use policy incompatibility, it MUST return an error
621 indication to the client.

622 If the identity service cannot process the information provided for other reasons, it MUST return an error indication
623 to the client.

624 3.2.4. ModifyInteraction

```
625 <element name="ModifyInteraction" maxOccurs="unbounded">  
626   <complexType>  
627     <complexContent>  
628       <extension base="carml:BaseInteractionType">  
629         <sequence>  
630           <element name="AttributeRef" type="carml:RefType" minOccurs="0"  
631             maxOccurs="unbounded"/>  
632           <element name="RoleRef" type="carml:RefType" minOccurs="0" maxOccurs="unbounded"/>  
633         </sequence>  
634       </extension>  
635     </complexContent>  
636   </complexType>  
637 </element>
```

638 The <AttributeRef> element has two attributes: the Ref attribute MUST reference an <Attribute> element
639 using a URI; the PolicyRef MUST reference a policy element using a URI.

640 The <RoleRef> element has two attributes: the Ref attribute MUST reference a <Role> element using a URI;
641 the PolicyRef MUST reference a policy element using a URI.

642 The identity service MUST return an indication of whether the service successfully received the request to update
643 the digital subjects' attributes or roles, or, whether the operation failed to complete.

644 The identity service MUST receive values for all <Attributes> or <Roles> that have Optional attribute set
645 to false; otherwise, it MUST return an error indication to the client.

646 If the identity service cannot process the request due to the subject not being known prior to the request, it MUST
647 return an error indication to the client.

648 If the identity service cannot provide requested information due to lack of user consent, it MUST return an error
649 indication to the client.

650 If the identity service cannot process the request due to use policy incompatibility, it MUST return an error
651 indication to the client.

652 If the identity service cannot process the information provided for other reasons, it MUST return an error indication
653 to the client.

654 3.2.5. ReadInteraction

```
655 <element name="ReadInteraction" maxOccurs="unbounded">
656   <complexType>
657     <complexContent>
658       <extension base="carml:BaseInteractionType">
659         <sequence>
660           <element name="AttributeRef" type="carml:RefType" minOccurs="0"
661             maxOccurs="unbounded"/>
662           <element name="PredicateRef" type="carml:RefType" minOccurs="0"
663             maxOccurs="unbounded"/>
664           <element name="RoleRef" type="carml:RefType" minOccurs="0" maxOccurs="unbounded"/>
665         </sequence>
666       </extension>
667     </complexContent>
668   </complexType>
669 </element>
```

670 The <AttributeRef> element has two attributes: the Ref attribute MUST reference an <Attribute> element
671 using a URI; the PolicyRef MUST reference a policy element using a URI.

672 The <PredicateRef> element has two attributes: the Ref attribute MUST reference an <Predicate> element
673 using a URI; the PolicyRef MUST reference a policy element using a URI.

674 The <RoleRef> element has two attributes: the Ref attribute MUST reference an <Role> element using a URI;
675 the PolicyRef MUST reference a policy element using a URI.

676 The identity service MUST return values of the prescribed type and cardinality for each element referenced withing
677 <AttributeRefs>, <PredicateRefs> and <RoleRefs>, with the exception of those elements that have
678 attribute optional set to true. If unable to do so, it MUST return an appropriate error message to the client.

679 The identity service MUST return only those attributes, predicates and roles whose release is consistent with the
680 <wsp:Policy> element found within the <Interaction> element and individual <Attribute>,
681 <Predicate> or <Role> elements.

682 If the identity service cannot provide requested information due to use policy incompatibility, it MUST return an
683 error indication to the client.

684 If the identity service cannot provide requested information due to lack of user consent, it MUST return an error
685 indication to the client.

686 If the identity service cannot provide the requested information for other reasons, it MUST return an error indication
687 to the client.

688 3.2.6. CompareInteraction

```
689 <element name="CompareInteraction" minOccurs="0" maxOccurs="unbounded">
690   <complexType>
691     <complexContent>
692       <extension base="carml:BaseInteractionType">
693         <sequence>
694           <element name="Filter" type="carml:FilterType"/>
695           <!-- Must have one or more filters -->
696         </sequence>
697       </extension>
698     </complexContent>
699   </complexType>
700 </element>
```

701 The client MUST provide values of the prescribed type and cardinality for each <AttrRefFilter>, <RoleRefFilter>, <PredRefFilter> element, with attribute Optional set to false, found within the <Filter> element. Otherwise, the identity service MUST return an appropriate error indication.

704 The identity service MUST return a failure indication if it cannot match against the values described by the <Filter> element, with attribute Optional set to false, based on the relationship defined by the attribute Operator. Else, it MUST return an indication of success.

707 If the identity service cannot provide requested information due to use policy incompatibility, it MUST return an error indication to the client.

709 If the identity service cannot provide requested information due to lack of user consent, it MUST return an error indication to the client.

711 If the identity service cannot provide the requested information for other reasons, it MUST return an error indication to the client.

713 3.2.7. FindInteraction

```
714 <element name="FindInteraction" maxOccurs="unbounded">
715   <complexType>
716     <complexContent>
717       <extension base="carml:BaseInteractionType">
718         <sequence>
719           <element name="AttributeRef" type="carml:RefType" minOccurs="0"
720 maxOccurs="unbounded"/>
721           <element name="PredicateRef" type="carml:RefType" minOccurs="0"
722 maxOccurs="unbounded"/>
723           <element name="RoleRef" type="carml:RefType" minOccurs="0" maxOccurs="unbounded"/>
724           <element name="Filter" type="carml:FilterType"/>
725           <!-- Must have one or more filters -->
726         </sequence>
727       </extension>
728     </complexContent>
729   </complexType>
730 </element>
```

731 The client MUST provide values of the prescribed type and cardinality for each <AttrRefFilter>, <RoleRefFilter>, <PredRefFilter> element, with attribute Optional set to false, found within the <Filter> element. Otherwise, the identity service MUST return an appropriate error indication.

734 One of the <AttributeRef> elements MAY have a PrimaryKey attribute set to True.

735 The identity service MUST return only those digital subjects such that each returned subject appropriately matches
736 the elements referenced within the <Filter> element which have Optional attribute set to False. The
737 identity service SHOULD use any PrimaryKey information available to optimize or design its search technique.

738 In addition, for each returned digital subject, the the identity service MUST return values of the prescribed type and
739 cardinality for each element referenced withing <AttributeRefs>, <PredicateRefs> and <RoleRefs>,
740 with the exception of those elements that have attribute optional set to true. If unable to do so, it MUST return an
741 appropriate error message to the client.

742 The identity service MUST return only those digital subjects whose use policies are consistent with the
743 <wsp:Policy> elements found in the <Interaction> element and individual filters.

744 The identity service MUST return a single digital subject. If more than one matching digital subject is found, it
745 MUST return an appropriate error indication to the client. If no matching digital subject is found, it MUST return an
746 appropriate error indication to the client.

747 3.2.8. SearchInteraction

```
748 <element name="SearchInteraction" maxOccurs="unbounded">  
749   <complexType>  
750     <complexContent>  
751       <extension base="carml:BaseInteractionType">  
752         <sequence>  
753           <element name="AttributeRef" type="carml:RefType" minOccurs="0" maxOccurs="unbounded"/>  
754           <element name="PredicateRef" type="carml:RefType" minOccurs="0" maxOccurs="unbounded"/>  
755           <element name="RoleRef" type="carml:RefType" minOccurs="0" maxOccurs="unbounded"/>  
756           <element name="Filter" type="carml:FilterType"/>  
757           <!-- Must have one or more filters -->  
758         </sequence>  
759         <attribute name="MaxSubjects" type="integer" use="optional" default="100"/>  
760         <attribute name="PageSize" type="integer" use="optional" default="1"/>  
761       </extension>  
762     </complexContent>  
763   </complexType>  
764 </element>
```

765 The client MUST provide values of the prescribed type and cardinality for each <AttrRefFilter>,
766 <RoleRefFilter>, <PredRefFilter> element, with attribute Optional set to false, found within
767 the <Filter> element. Otherwise, the identity service MUST return an appropriate error indication.

768 One of the <AttributeRef> elements MAY have a PrimaryKey attribute set to True.

769 The identity service MUST return only those digital subjects such that each returned subject appropriately matches
770 the elements referenced within the <Filter> element which have Optional attribute set to False. The identity
771 service SHOULD use any PrimaryKey information available to optimize or design its search technique.

772 In addition, for each returned digital subject, the the identity service MUST return values of the prescribed type and
773 cardinality for each element referenced withing <AttributeRefs>, <PredicateRefs> and <RoleRefs>,
774 with the exception of those elements that have attribute optional set to true. If unable to do so, it MUST return an
775 appropriate error message to the client.

776 The identity service MUST return only those digital subjects whose use policies are consistent with the
777 <wsp:Policy> elements found in the <Interaction> element and individual filters.

778 If the identity service cannot provide requested information due to use policy incompatibility, it MUST return an
779 error indication to the client.

780 If the identity service cannot provide requested information due to lack of user consent, it MUST return an error
781 indication to the client.

782 If the identity service cannot provide the requested information for other reasons, it MUST return an error indication
783 to the client.

784 **Appendix A.**

785 **A.1. DataType URIs**

786 Based on Section A.2 of the XACML 2.0 specification.

- 787 1. <http://www.w3.org/2001/XMLSchema#string>
- 788 2. <http://www.w3.org/2001/XMLSchema#boolean>
- 789 3. <http://www.w3.org/2001/XMLSchema#integer>
- 790 4. <http://www.w3.org/2001/XMLSchema#double>
- 791 5. <http://www.w3.org/2001/XMLSchema#time>
- 792 6. <http://www.w3.org/2001/XMLSchema#date>
- 793 7. <http://www.w3.org/2001/XMLSchema#dateTime>
- 794 8. <http://www.w3.org/2001/XMLSchema#anyURI>
- 795 9. <http://www.w3.org/2001/XMLSchema#hexBinary>
- 796 10. <http://www.w3.org/2001/XMLSchema#base64Binary>
- 797 11. <http://www.w3.org/TR/2002/WD-xquery-operators-20020816#dayTimeDuration>
- 798 12. <http://www.w3.org/TR/2002/WD-xquery-operators-20020816#yearMonthDuration>
- 799 13. <urn:oasis:names:tc:xacml:1.0:data-type:x500Name>
- 800 14. <urn:oasis:names:tc:xacml:1.0:data-type:rfc822Name>
- 801 15. <urn:oasis:names:tc:xacml:2.0:data-type:ipAddress>
- 802 16. <urn:oasis:names:tc:xacml:2.0:data-type:dnsName>

803 For the sake of improved interoperability, it is RECOMMENDED that all time references be in UTC time.

804 XACML defines three data-types; these are:

- 805 1. <urn:oasis:names:tc:xacml:1.0:data-type:x500Name>
- 806 2. <urn:oasis:names:tc:xacml:1.0:data-type:rfc822Name>
- 807 3. <urn:oasis:names:tc:xacml:2.0:data-type:ipAddress>
- 808 4. <urn:oasis:names:tc:xacml:2.0:data-type:dnsName>

809 These types represent identifiers for subjects or resources and appear in several standard applications, such as
810 TLS/SSL and electronic mail.

811 **A.2. Comparison Operators**

OPERATOR	Type	Description
doesnotcontain	string	Determine if value provided is a substring of the referenced value
beginswith	string	Determine if value provided is a prefix of the referenced value
endswith	string	Determine if value provided is a suffix of the referenced value
equals	All types	
notequals	All types	
gt	Int, double	Determine if value provided is a greater than referenced value
lt	Int, double	Determine if value provided is less than referenced value
geq	Int, double	Determine if value provided is a greater than or equal to the referenced value
leq	Int, double	Determine if value provided is a less than or equal to the referenced value