



Client Attribute Requirements Markup Language (“CARML”) Specification

Editors:

Phil Hunt, Oracle Corporation
Prateek Mishra, Oracle Corporation

Contributors:

Shin Adachi, NTT
Conor Cahill, Intel
Makoto Hatakeyama, NEC Corporation
Paul Madsen, NTT
Colin Wallis, New Zealand
Peter Davis, Neustar
Eric Tiffany, Liberty Alliance
Sampo Kellomaki, Symlabs
Hubert Le Van Gong, SUN Microsystems
George Fletcher, AOL LLC

Abstract:

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

Filename: liberty-igf-carml-v1.0.pdf

31 **NOTICE:**

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

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

49 Copyright © 2007-2009

50 ActivIdentity, Trent Adams, Adetti, Adobe Systems, AOL, BEA Systems, Berne, University
51 of Applied Sciences, Gerald Beuchelt, BIPAC, John Bradley, British Telecommunications
52 plc, Hellmuth Broda, Bronnoysund Register Centre, BUPA, CA, Canada Post Corporation,
53 Center for Democracy and Technology, Chief, Information Office Austria, China Internet
54 Network Information Center (CNNIC), ChoicePoint, Citi, City University, Clareity
55 Security, Dan Combs, Computer & Communications Industry Association, Courion
56 Corporation, Danish Biometrics Research Proj. Consortium, Danish National IT and
57 Telecom Agency, Deny All, Deutsche Telekom AG, DGME, Brian Dilley, Diversinet Corp.,
58 Drummond Group Inc., East of England Telematics Development Trust Ltd, EIfEL,
59 Electronics and Telecommunications Research Institute (ETRI), Engineering Partnership in
60 Lancashire, Enterprise Java Victoria Inc., Entr'ouvert, Ericsson, Evidian, Fidelity
61 Investments, Financial Services Technology Consortium (FSTC), Finland National Board of
62 Taxes, Fischer International, France Telecom, Fraunhofer-Gesellschaft, Fraunhofer Institute
63 for Integrated Circuits IIS, Fraunhofer Institute for Secure Information Technology (SIT),
64 Fraunhofer Institut for Experimentelles Software Engineering, Fugen Solutions, Fujitsu
65 Services Oy, Fun Communications GmbH, Gemalto, Giesecke & Devrient GMBH, Global
66 Platform, GSA Office of Governmentwide Policy, Healthcare Financial Management
67 Association (HFMA), Health Information and Management Systems Society (HIMSS),
68 Helsinki Institute of Physics, Jeff Hodges, Hongkong Post, Guy Huntington, Imprivata,
69 Information Card Foundation, Institute of Bioorganic Chemistry Poland, Institute of
70 Information Management of the University, Institut Experimentelles Software Engineering
71 (IESE), Intel Corporation, International Institute of Telecommunications, International
72 Security, Trust and Privacy Alliance, Internet2, Interoperability Clearinghouse (ICH),
73 ISOC, Java Wireless Competency Centre (JWCC), Kantega AS, Kuppinger Cole & Partner,
74 Kuratorium OFFIS e.V., Colin Mallett, Rob Marano, McMaster University,

75 MEDNETWorld.com, Methics Oy, Mortgage Bankers Association (MBA), Mydex,
76 National Institute for Urban Search & Rescue Inc NEC Corporation, Network Applications
77 Consortium (NAC), Neustar, Newspaper Association of America, New Zealand
78 Government State Services Commission, NHK (Japan Broadcasting Corporation) Science &
79 Technical Research Laboratories, Nippon Telegraph and Telephone Company, Nokia
80 Corporation, Nortel, NorthID Oy, Norwegian Agency for Public Management and
81 eGovernment, Norwegian Public Roads Administration, Novell, NRI Pacific, Office of the
82 Information Privacy Commissioner of Ontario, Omnibranch, OpenIAM, Oracle USA, Inc.,
83 Organisation Internationale pour la Sécurité des Transactions Électroniques (OISTE), Oslo
84 University, Our New Evolution, PAM Forum, Parity Communications, Inc., PayPal, Phase2
85 Technology, Ping Identity Corporation, Bob Pinheiro, Platinum Solutions, Postsecondary
86 Electronic Standards Council (PESC), Purdue University, RSA Security, Mary Ruddy,
87 SAFE Bio Pharma, SanDisk Corporation, Shidler Center for Law, Andrew Shikiar, Signicat
88 AS, Singapore Institute of Manufacturing Technology, Software & Information Industry
89 Association, Software Innovation ASA, Sprint Nextel Corporation, Studio Notarile
90 Genghini-SNG, Sunderland City Council, SUNET, Sun Microsystems, SwissSign AG,
91 Technische Universitat Berlin, Telefonica S.A., TeleTrusT, TeliaSonera Mobile Networks
92 AB, TERENA, Thales e-Security, The Boeing Company, The Financial Services
93 Roundtable/BITS, The Open Group, The University of Chicago as Operator of Argonne
94 National Laboratory, TRUSTe, tScheme Limited, UNINETT AS, Universidad Politecnica
95 de Madrid, University of Birmingham, University of Kent, University of North Carolina at
96 Charlotte, University of Ottawa (TTBE), U.S. Department of Defense, VeriSign, Vodafone
97 Group Plc, Web Services Competence Center (WSCC), Zenn New Media

98 All rights reserved
99

100				
101	1	Introduction	5	
102	1.1	Example	7	
103	1.2	Terminology.....	9	
104	1.3	References	10	
105	1.3.1	Normative References.....	10	
106	1.3.2	Non-Normative References	10	
107	1.4	Notation.....	10	
108	2	Foundations.....	12	Deleted: 11
109	2.1	AttributeOrPredicateSuperType.....	12	Deleted: 11
110	2.2	CardinalityType	12	Deleted: 11
111	2.3	AttributeType.....	13	Deleted: 12
112	2.4	PredicateType	13	Deleted: 12
113	2.5	RefType.....	14	Deleted: 12
114	2.6	FilterRefType.....	14	Deleted: 12
115	2.7	FilterType.....	15	Deleted: 13
116	3	Client Attribute Requirements	18	Deleted: 14
117	3.1	DataDefs	20	Deleted: 15
118	3.1.1	ExternalDefsRef.....	21	Deleted: 16
119	3.1.2	Attributes.....	21	Deleted: 17
120	3.1.3	Predicates	21	Deleted: 17
121	3.1.4	Roles.....	21	Deleted: 17
122	3.1.5	Policies	21	Deleted: 18
123	3.2	Interaction	21	Deleted: 18
124	3.2.1	BaseInteractionType.....	22	Deleted: 18
125	3.2.2	AddInteraction	22	Deleted: 18
126	3.2.3	DeleteInteraction.....	23	Deleted: 18
127	3.2.4	ModifyInteraction	24	Deleted: 19
128	3.2.5	ReadInteraction.....	25	Deleted: 20
129	3.2.6	CompareInteraction	26	Deleted: 20
130	3.2.7	FindInteraction.....	27	Deleted: 21
131	3.2.8	SearchInteraction	28	Deleted: 22
132	4. Appendix A	30	Deleted: 23	
133	4.1.	Data Type URIs.....	30	Deleted: 24
134	4.2.	Comparison Operators.....	31	Deleted: 26
135				Deleted: 26

1 Introduction

163

164

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

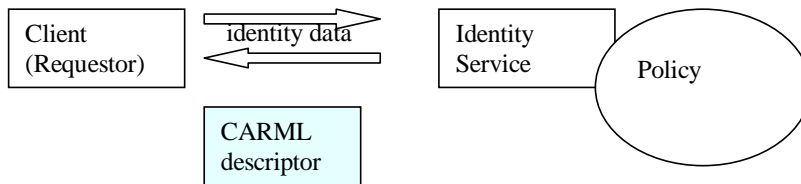
171

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

176

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

182



183 **Figure 1**

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

190 We do assume that the identity service supports some of the following operations, each of
191 which is expressed by one or more CARML interaction elements:
192

- 193 1. Given a digital subject, retrieve or read attributes, roles or predicate values
194 associated with the subject
195
- 196 2. Given a digital subject, determine if certain predicates, roles, or attribute values are
197 associated with it.
198
- 199 3. Given attribute values or roles, retrieve digital subjects that possess those values or
200 roles
201
- 202 4. Given a set of attribute values or roles, request the creation of a digital subject
203 associated with these values
204
- 205 5. Given a digital subject, request the update of attributes or roles associated with the
206 digital subject
207
- 208 6. Given a digital subject, request that the digital subject be deleted.
209
210

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

218 For example, in a distributed multi-application environment, a single application's
219 "AddInteraction", a request to add a new record, should be considered solely as a request for
220 a certain type of service. The identity service may respond to the request in many different
221 ways – adding a new record in persistent store, or just modifying an existing identity record
222 to add information specific to an application to that record. Likewise, for a DeleteInteraction,
223 it will be policy and context information within the identity service and other infra-structure
224 that determine the actions carried out when the deletion of a digital subject is requested (e.g.
225 delete from persistent store, log and archive request, set flag indicating delete requested).
226 The means by which a CARML descriptor is defined or created is outside the scope of this
227 specification. Depending upon business-context, such a descriptor may be created via
228 automatic or manual negotiation or provided unilaterally by the client or the identity service.
229

230 1.1 Example

```
231 [a01] <carml:ClientAttrReq AppName="CARML Example" Description="Demonstrates features of  
232 CARML Schema" xmlns:carml="urn:igf:client:0.9:carml"  
233 xmlns:wsp="http://www.w3.org/ns/ws-policy"  
234 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
235 xsi:schemaLocation="urn:igf:client:0.9:carml igf-carml-09.xsd">  
236  
237 [a02] <DataDefs>  
238 [a03] <DataDefs>  
239 [a04] <Attributes>  
240 [a05] <Attribute Cardinality="single" DataType="string" DisplayName="Surname"  
241 Name="sn" />  
242 [a06] <Attribute Cardinality="single" DataType="string" Description="One or more  
243 names that are considered given names. The first name should be the preferred  
244 name." DisplayName="Given names" Name="givenname" />  
245 [a07] <Attribute Cardinality="single" DataType="urn:oasis:names:tc:xacml:1.0:data-  
246 type:rfc822Name" DisplayName="E-Mail" Name="mail" />  
247 [a08] <Attribute Cardinality="single" DataType="string" DisplayName="Telephone"  
248 Name="telephone" />  
249 [a09] <Attribute Cardinality="single" DataType="string" DisplayName="Last 4 Digits  
250 SSN" Name="Last4SSN" />  
251 [a10] </Attributes>  
252 [a11] <Predicates>  
253 [a12] <Predicate Description="For the jurisdiction of the user, a determination  
254 that the subject can travel alone." DisplayName="Adult" Name="IsAdult" />  
255 [a13] <Predicate Description="A resident of the EU" DisplayName="EU Resident"  
256 Name="IsEUResident" />  
257 [a14] </Predicates>  
258 [a15] <Roles>  
259 [a16] <Role Description="Able to book business class tickets"  
260 DisplayName="Business Class FLYer" Name="BusinessClassFlyer" />  
261 [a17] <Role Description="The passenger's account is active." DisplayName="Account  
262 active" Name="IsActive" />  
263 [a18] <Role Description="Person is an employee" Name="Employee" />  
264 [a19] <Role Description="Person is a contractor" Name="Contractor" />  
265 [a20] </Roles>  
266 [a21] <Policies>  
267 [a22] <wsp:Policy Name="http://tempuri.org/" />  
268 [a23] </Policies>  
269 [a24] </DataDefs>  
270 [a25] </DataDefs>  
271 [a26]
```

274 The <DataDefs> element (lines [a03]-,[a26]) defines the attributes, roles, predicates, and
275 privacy policies of interest in the <ClientAttrReq>. Attributes, roles and predicates are
276 the foundational components out of which interactions are built. This document does provide
277 details of privacy policies, these are described in [CARML-Profile-Privacy-Constraints].

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

```
282 [a27] <ReadInteraction Description="" Name="ReadProfile">
283 [a28]   <wsp:Policy Name="http://tempuri.org" />
284 [a29]   <AttributeRef Ref="#mail" />
285 [a30]   <AttributeRef Ref="#sn" />
286 [a31]   <AttributeRef Ref="#givenname" />
287 [a32]   <AttributeRef Ref="#telephone" Optional="true" />
288 [a33]   <PredicateRef Ref="#IsAdult" Optional="true" />
289 [a34]   <PredicateRef Ref="#IsEUResident" />
290 [a35]   <RoleRef Ref="#BusinessClassFlyer" />
291 [a36] </ReadInteraction>
292 [a37]
293 [a38] <FindInteraction Description="Locate user for authentication purposes."
294 Name="LocateUser">
295 [a39]   <wsp:Policy Name="http://tempuri.org" />
296 [a40]   <Filter Match="all">
297 [a41]     <AttrRefFilter Ref="#mail" PrimaryKey="true" />
298 [a42]     <Filter Match="any">
299 [a43]       <RoleRefFilter Ref="#Employee" />
300 [a44]       <RoleRefFilter Ref="#Contractor" />
301 [a45]     </Filter>
302 [a46]   </Filter>
303 [a47] </FindInteraction>
304 [a48]
305 [a49] <SearchInteraction Name="SearchLastName" Description="Returns potential matches
306 for a given surname">
307 [a50]   <AttributeRef Ref="#mail" />
308 [a51]   <AttributeRef Ref="#sn" />
309 [a52]   <Filter Match="all">
310 [a53]     <AttrRefFilter Ref="#sn" />
311 [a54]     <RoleRefFilter Ref="#IsActive" />
312 [a55]   </Filter>
313 [a56] </SearchInteraction>
314 [a57]
315 [a58] <CompareInteraction Name="VerifyIdentity" Description="Used to verify information
316 provided by user">
317 [a59]   <Filter Match="all">
318 [a60]     <AttrRefFilter Ref="#Last4SSN" Operator="endswith" />
319 [a61]     <AttrRefFilter Ref="#mail" Operator="equals" />
320 [a62]   </Filter>
321 [a63] </CompareInteraction>
322 [a64]
323 [a65] <ModifyInteraction Name="UpdateTelephoneNumber">
324 [a66]   <AttributeRef Ref="#telephone" />
325 [a67] </ModifyInteraction>
326 [a68]
327 [a69] <AddInteraction Name="AddNewUser">
328 [a70]   <AttributeRef Ref="#mail" />
329 [a71]   <AttributeRef Ref="#sn" />
330 [a72]   <AttributeRef Ref="#givenname" />
331 [a73]   <AttributeRef Ref="#telephone" Optional="true" />
332 [a74]   <RoleRef Ref="#Employee" Optional="true" />
333 [a75]   <RoleRef Ref="#Contractor" Optional="true" />
334 [a76] </AddInteraction>
335 [a77]
336 [a78] <DeleteInteraction Name="UnRegisterUser" Description="User cannot use this
337 service goingforward" />
338 [a79] </carml:ClientAttrReq>
339
```


340
341 The contents of the <ReadInteraction> element ([a27]-[a36]) indicate that the service
342 requires certain attribute, predicate and role values, with some declared optional.

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

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

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

353 The <ModifyInteraction> element ([a65]-[a67]) indicates that the service plans to
354 provide the telephone number of certain digital subjects.

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

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

360 **1.2 Terminology**

361
362 Conventional XML namespace prefixes are used throughout the listings in this specification
363 to stand for their respective namespaces, whether or not a namespace declaration is present in
364 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

Prefix	XML Namespace	Comments
		specification [XML-Schema1]. 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 [XML-Schema1] for schema-related markup that appears in XML instances.

365

366 1.3 References

367 1.3.1 Normative References

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

371 [WS-Policy] **Web Services Policy 1.5 – Framework, October 2007.**
372 <http://www.w3.org/TR/2004/REC-xmldata-model-20041028/>

373 [PrivAssert] **Liberty Alliance Privacy Constraints Specification**

374 [CARML-Profile-Privacy-Constraints] **CARML Profile of Privacy Policy Constraints**

375

376 1.3.2 Non-Normative References

377 None

378 1.4 Notation

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

381 The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT",
382 "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this
383 specification are to be interpreted as described in IETF RFC 2119 [[RFC2119](#)]
384 "they MUST only be used where it is actually required for interoperation or to limit
385 behaviour which has potential for causing harm (e.g., limiting retransmissions)"
386 These keywords are thus capitalized when used to unambiguously specify requirements over
387 protocol and application features and behavior that affect the interoperability and security of
388 implementations. When these words are not capitalized, they are meant in their natural-
389 language sense.
390

391 2 Foundations

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

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

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

403 2.1 AttributeOrPredicateSuperType

404

```
405 <complexType name="AttributeOrPredicateSuperType" abstract="true">  
406   <attribute name="Name" type="ID" use="required"/>  
407   <attribute name="DisplayName" type="string"  
408 use="optional"/>  
409   <attribute name="Description" type="string"  
410 use="optional"/>  
411   <anyAttribute namespace="##other" processContents="lax"/>  
412 </complexType>
```

413

414 Name

415 The name of the attribute, predicate or filter

416 DisplayName

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

418 Description

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

420 2.2 CardinalityType

421

```
422 <simpleType name="CardinalityType">  
423   <restriction base="string">
```

```
424         <enumeration value="zero"/>
425         <enumeration value="single"/>
426         <enumeration value="multiple"/>
427     </restriction>
428 </simpleType>
429
```

430 2.3 AttributeType

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

```
435 <complexType name="AttributeType">
436     <complexContent>
437         <extension base="carml:AttributeOrPredicateSuperType">
438             <attribute name="Cardinality"
439 type="carml:CardinalityType" use="optional"/>
440             <attribute name="DataType" type="anyURI"
441 use="optional" default="string"/>
442         </extension>
443     </complexContent>
444 </complexType>
445
```

445

446 Cardinality

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

448 DataType

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

451

452 2.4 PredicateType

453

454 PredicateType describes a single named predicate, a boolean valued decision or judgement,
455 provided by an identity service to a client.

456

457

```
458 <complexType name="PredicateType">
459     <complexContent>
460         <extension
461 base="carml:AttributeOrPredicateSuperType"/>
462     </complexContent>
463 </complexType>
464
```

464

465 2.5 RefType

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

```
468  
469 <complexType name="RefType">  
470   <attribute name="Ref" type="anyURI" use="required"/>  
471   <attribute name="PolicyRef" type="anyURI"  
472 use="optional"/>  
473   <attribute name="Optional" type="boolean" use="optional"  
474 default="false"/>  
475   <attribute name="Description" type="string"  
476 use="optional"/>  
477 </complexType>
```

478

479 Ref

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

481 PolicyRef

482 URI of local or external privacy policy

483 Optional

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

485 2.6 FilterRefType

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

```
487  
488 <complexType name="FilterRefType">  
489   <complexContent>  
490     <extension base="carml:RefType">  
491       <attribute name="Cardinality"  
492 type="carml:CardinalityType" use="optional" default="single"/>  
493       <attribute name="PrimaryKey" type="boolean"  
494 default="false"/>  
495       <attribute name="Operator" default="equals">  
496         <simpleType>  
497           <restriction base="string">  
498             <enumeration value="contains"/>  
499             <enumeration value="doesnotcontain"/>  
500             <enumeration value="dynamic"/>  
501             <enumeration value="beginswith"/>
```

```
502         <enumeration value="endswith"/>
503         <enumeration value="equals"/>
504         <enumeration value="notequals"/>
505         <enumeration value="gt"/>
506         <enumeration value="lt"/>
507         <enumeration value="geq"/>
508         <enumeration value="leq"/>
509     </restriction>
510 </simpleType>
511 </attribute>
512 <attribute name="Name" type="ID" use="optional"/>
513 </extension>
514 </complexContent>
515 </complexType>
```

516
517

518 **Cardinality**

519 Whether the requestor provides single or multiple values

520 **Data Type**

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

522 **Primary Key**

523 Whether the client or requestor views the attribute as
524 a key or index

525 **Operator**

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

529

530 **2.7 FilterType**

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

533

```
534 <complexType name="FilterType">
535     <choice maxOccurs="unbounded">
536         <element name="AttrRefFilter" type="carml:FilterRefType"
537 minOccurs="0" maxOccurs="unbounded"/>
538         <element name="RoleRefFilter" type="carml:RefType"
539 minOccurs="0" maxOccurs="unbounded"/>
540         <element name="PredRefFilter" type="carml:RefType"
541 minOccurs="0" maxOccurs="unbounded"/>
542         <element name="Filter" type="carml:FilterType"
543 minOccurs="0" maxOccurs="unbounded"/>

```

```
544     </choice>
545     <attribute name="Match" default="all">
546         <simpleType>
547             restriction base="string">
548                 <enumeration value="any"/>
549                 <enumeration value="all"/>
550             </restriction>
551         </simpleType>
552     </attribute>
553     <attribute name="Description" use="optional"/>
554 </complexType>
555
556
```


557 **AttRefFilter**

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

559 **RoleRefFilter**

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

561 **PredRefFilter**

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

563 **Filter**

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

566 **Match**

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

569

570 3 Client Attribute Requirements

```
571
572 <element name="ClientAttrReq">
573   <!-- root element for a CARML declaration -->
574   <complexType>
575     <sequence>
576       <element name="DataDefs">
577         ...
578       </element>
579       <choice minOccurs="0" maxOccurs="unbounded">
580         <element name="AddInteraction"
581 maxOccurs="unbounded">
582         ...
583       </element>
584         <element name="DeleteInteraction"
585 type="carml:BaseInteractionType" maxOccurs="unbounded"/>
586         ...
587       </element>
588         <element name="ReadInteraction"
589 maxOccurs="unbounded">
590         ...
591       </element>
592         <element name="ModifyInteraction"
593 maxOccurs="unbounded">
594         ...
595       </element>
596         <element name="CompareInteraction" minOccurs="0"
597 maxOccurs="unbounded">
598         ...
599       </element>
600         <element name="FindInteraction"
601 maxOccurs="unbounded">
602         ...
603       </element>
604         <element name="SearchInteraction"
605 maxOccurs="unbounded">
606         ...
607       </element>
608     </choice>
609   <!-- Application policy -->
610   <choice minOccurs="0" maxOccurs="unbounded">
611     <element ref="wsp:Policy"/>
612     <element ref="wsp:PolicyReference"/>
613   </choice>
614 </sequence>
615   <attribute name="AppName" type="string" use="required"/>
616   <attribute name="Description" type="string" use="optional"/>
617   <attribute name="CarmlURI" type="anyURI" use="optional"/>
618 </complexType>
```

619 `</element>`

620
621 `<ClientAttrReq>` is the root element that captures the client attribute requirements of a
622 specific entity. The requirements are captured by a set of zero or more interaction elements.
623 Interaction elements include `<AddInteraction>`, `<ReadInteraction>`, `<ModifyInteraction>`,
624 `<UpdateInteraction>`, `<CompareInteraction>`, `<FindInteraction>` and `<SearchInteraction>`
625 elements. Each of these elements references attributes, predicates, roles and policies declared
626 in the `<DataDefs>` element.

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

631
632 [PrivAssert] defines privacy policy assertions that express privacy constraints over the use of
633 identity data. [WS-Policy] provides a general framework for expressing composite policies
634 built out of atomic assertions.

635 The `<wsp:Policy>` or `<wsp:PolicyReference>` element carries policy assertions based on
636 WS-Policy with atomic assertions drawn only from [PrivAssert]. These policies apply to all
637 of the interactions defined within the `<ClientAttrReq>` element.

638

639 **AppName**

640 String name associated with `<ClientAttrReq>` element

641 **CarmlURI**

642 URI associated with the `<ClientAttrReq>` element

643

644

645 3.1 DataDefs

646 The <DataDefs> element defines all the different entities that might be used via reference by
647 one or more <Interaction> elements found within the <ClientAttrReq> element.
648

```
649 <element name="DataDefs">
650   <complexType>
651     <sequence>
652       <element name="ExternalDataDefsRef" minOccurs="0"
653 maxOccurs="unbounded">
654         <complexType>
655           <attribute name="CarmlURI" type="anyURI"
656 use="required"/>
657           <attribute name="AppName" type="string"
658 use="optional"/>
659           <attribute name="ProcessNestedDefinitions"
660 type="boolean" default="true"/>
661           <anyAttribute namespace="##any"
662 processContents="lax"/>
663         </complexType>
664       </element>
665       <element name="Attributes">
666         <complexType>
667           <sequence>
668             <element name="Attribute"
669 type="carml:AttributeType" minOccurs="0" maxOccurs="unbounded"/>
670           </sequence>
671         </complexType>
672       </element>
673       <element name="Predicates">
674         <complexType>
675           <sequence>
676             <element name="Predicate"
677 type="carml:PredicateType" minOccurs="0" maxOccurs="unbounded"/>
678           </sequence>
679         </complexType>
680       </element>
681       <element name="Roles">
682         <complexType>
683           <sequence>
684             <element name="Role" type="carml:PredicateType"
685 minOccurs="0" maxOccurs="unbounded"/>
686           </sequence>
687         </complexType>
688       </element>
689       <element name="Policies">
690         <complexType>
691           <sequence>
692             <element ref="wsp:Policy" minOccurs="0"
693 maxOccurs="unbounded"/>
694           </sequence>
695         </complexType>

```

```
696         </element>  
697     </sequence>  
698 </complexType>  
699 </element>
```

700

701 3.1.1 ExternalDefsRef

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

704

705 CarmlURI

706 URI of referenced <ClientAttrReq> element

707 AppName

708 Optional name of the referenced <ClientAttrReq> element

709 ProcessNestedDefinitions

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

712

713 3.1.2 Attributes

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

716 3.1.3 Predicates

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

719 3.1.4 Roles

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

722 3.1.5 Policies

723 [CARML-Profile-Privacy-Constraints] defines privacy policy assertions that express privacy
724 constraints for identity data. The <Policies> element carries policy assertions based on WS-
725 Policy [WS-Policy] with atomic assertions drawn only from [PrivAssert]. These assertions
726 may be referenced by <Interaction> elements.

727 3.2 Interaction

728 An interaction represents a single exchange between a client and an identity service. Some
729 interactions assume that the client or requestor will provide information about a digital

730 subject (the target identity) whereas other interactions require the identity service to find or
731 create a digital subject.

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

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

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

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

740 (1) Information about the client’s intent , whether identity information is being read or
741 updated or whether digital subjects are to be retrieved based on certain criteria.

742 (2) The attributes, roles, predicates and policies relevant to the interaction.

743 (3) Additional privacy policies that constrain the exchange, specific to the interaction.
744

3.2.1 BaseInteractionType

```
747 <complexType name="BaseInteractionType" abstract="true">  
748   <sequence>  
749     <!-- Holds interaction policies -->  
750     <choice minOccurs="0" maxOccurs="unbounded">  
751       <element ref="wsp:Policy"/>  
752       <element ref="wsp:PolicyReference"/>  
753     </choice>  
754   </sequence>  
755   <attribute name="Name" type="ID" use="required"/>  
756   <attribute name="Description" use="optional"/>  
757     <attribute name="EntityName" type="NCName"  
758 use="optional"/>  
759 </complexType>
```

760
761 EntityName - this attribute allows a set of related interactions to share a common identifier
762

3.2.2 AddInteraction

```
764  
765 <element name="AddInteraction" maxOccurs="unbounded">  
766   <complexType>
```

```
767     <complexContent>
768         <extension base="carml:BaseInteractionType">
769             <sequence>
770                 <element name="AttributeRef" type="carml:RefType"
771 minOccurs="0" maxOccurs="unbounded"/>
772                 <element name="RoleRef" type="carml:RefType"
773 minOccurs="0" maxOccurs="unbounded"/>
774             </sequence>
775         </extension>
776     </complexContent>
777 </complexType>
778 </element>
```

779

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

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

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

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

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

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

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

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

798 3.2.3 DeleteInteraction

799

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

802

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

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

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

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

813 3.2.4 ModifyInteraction

814

```
815 <element name="ModifyInteraction" maxOccurs="unbounded">  
816   <complexType>  
817     <complexContent>  
818       <extension base="carml:BaseInteractionType">  
819         <sequence>  
820           <element name="AttributeRef" type="carml:RefType"  
821 minOccurs="0" maxOccurs="unbounded"/>  
822           <element name="RoleRef" type="carml:RefType"  
823 minOccurs="0" maxOccurs="unbounded"/>  
824         </sequence>  
825         <attribute name="Mode" type="carml:ModeType"  
826 default="replace"/>  
827       </extension>  
828     </complexContent>  
829   </complexType>  
830 </element>
```

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

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

837 The <Mode> attribute indicates the form of modification desired by the client.

838 1) replace - the client indicates that the current bindings for the referenced roles and
839 attributes be replaced with values provided by the client

840 2) add - the client indicates that the current bindings for the referenced roles and attributes be
841 augmented with values provided by the client

842 3) remove - the client indicates that the values provided by the client, be removed from the
843 current bindings of the referenced roles and attributes

844 The identity service MUST return an indication of whether the service successfully received
845 the request to update the digital subjects’ attributes or roles, or, whether the operation failed
846 to complete.

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

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

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

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

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

857 3.2.5 ReadInteraction

858

```
859 <element name="ReadInteraction" maxOccurs="unbounded">  
860   <complexType>  
861     <complexContent>  
862       <extension base="carml:BaseInteractionType">  
863         <sequence>  
864           <element name="AttributeRef" type="carml:RefType"  
865 minOccurs="0" maxOccurs="unbounded"/>  
866           <element name="PredicateRef" type="carml:RefType"  
867 minOccurs="0" maxOccurs="unbounded"/>  
868           <element name="RoleRef" type="carml:RefType"  
869 minOccurs="0" maxOccurs="unbounded"/>  
870         </sequence>  
871       </extension>  
872     </complexContent>  
873   </complexType>  
874 </element>
```

875

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

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

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

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

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

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

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

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

897

898 3.2.6 CompareInteraction

899

```
900 <element name="CompareInteraction" minOccurs="0"  
901 maxOccurs="unbounded">  
902   <complexType>  
903     <complexContent>  
904       <extension base="carml:BaseInteractionType">  
905         <sequence>  
906           <element name="Filter" type="carml:FilterType"/>  
907           <!-- Must have one or more filters -->  
908         </sequence>  
909       </extension>  
910     </complexContent>  
911   </complexType>  
912 </element>
```

913

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

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

921 Clients MAY omit <AttrRefFilter>, <RoleRefFilter>,<PredRefFilter> elements found
922 within the <Filter> element which have attribute Optional set to true. In such a case, the
923 identity service SHOULD treat the
924 corresponding conditions as satisfied, that is they always evaluate to true.

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

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

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

931 3.2.7 FindInteraction

932

```
933 <element name="FindInteraction" maxOccurs="unbounded">  
934 <complexType>  
935   <complexContent>  
936     <extension base="carml:BaseInteractionType">  
937       <sequence>  
938         <element name="AttributeRef" type="carml:RefType"  
939 minOccurs="0" maxOccurs="unbounded"/>  
940         <element name="PredicateRef" type="carml:RefType"  
941 minOccurs="0" maxOccurs="unbounded"/>  
942         <element name="RoleRef" type="carml:RefType"  
943 minOccurs="0" maxOccurs="unbounded"/>  
944         <element name="Filter" type="carml:FilterType"/>  
945         <!-- Must have one or more filters -->  
946       </sequence>  
947     </extension>  
948   </complexContent>  
949 </complexType>  
950 </element>
```

951

952 The client MUST provide values of the prescribed type and cardinality for each
953 <AttrRefFilter>, <RoleRefFilter>,<PredRefFilter> element, with attribute

954 Optional set to false, found within the <Filter> element. Otherwise, the identity service
955 MUST return an appropriate error indication.
956 One of the <AttributeRef> elements MAY have a PrimaryKey attribute set to True.

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

Comment [jb1]: Is the formatting correct for the “PrimaryKey” here?

961 Clients MAY omit <AttrRefFilter>, <RoleRefFilter>, <PredRefFilter> elements found
962 within the <Filter> element which have attribute Optional set to true. In such a case, the
963 identity service SHOULD treat the corresponding conditions as satisfied, that is they always
964 evaluate to true.

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

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

973 The identity service MUST return a single digital subject. If more than one matching digital
974 subject is found, it MUST return an appropriate error indication to the client. If no matching
975 digital subject is found, it MUST return an appropriate error indication to the client.
976

3.2.8 SearchInteraction

```
978  
979 <element name="SearchInteraction" maxOccurs="unbounded">  
980   <complexType>  
981     <complexContent>  
982       <extension base="carml:BaseInteractionType">  
983         <sequence>  
984           <element name="AttributeRef" type="carml:RefType"  
985 minOccurs="0" maxOccurs="unbounded"/>  
986           <element name="PredicateRef" type="carml:RefType"  
987 minOccurs="0" maxOccurs="unbounded"/>  
988           <element name="RoleRef" type="carml:RefType"  
989 minOccurs="0" maxOccurs="unbounded"/>  
990           <element name="Filter" type="carml:FilterType"/>  
991           <!-- Must have one or more filters -->  
992         </sequence>  
993         <attribute name="MaxSubjects" type="integer"  
994 use="optional" default="100"/>  
995
```

```
995         <attribute name="PageSize" type="integer" use="optional"  
996 default="1"/>  
997     </extension>  
998     </complexContent>  
999 </complexType>  
1000 </element>
```

1001
1002 The client MUST provide values of the prescribed type and cardinality for each
1003 <AttrRefFilter>, <RoleRefFilter>, <PredRefFilter> element, with attribute
1004 Optional set to false, found within the <Filter> element. Otherwise, the identity service
1005 MUST return an appropriate error indication.
1006 One of the <AttributeRef> elements MAY have a `PrimaryKey` attribute set to True.

1007 The identity service MUST return only those digital subjects such that each returned subject
1008 appropriately matches the elements referenced within the <Filter> element which have
1009 Optional attribute set to False. The identity service SHOULD use any `PrimaryKey`
1010 information available to optimize or design its search technique.

1011 Clients MAY omit <AttrRefFilter>, <RoleRefFilter>, <PredRefFilter> elements found within
1012 the <Filter> element which have attribute Optional set to true. In such a case, the identity service
1013 SHOULD treat the corresponding conditions as satisfied, that is they always evaluate to true.

1014 In addition, for each returned digital subject, the the identity service MUST return values of the
1015 prescribed type and cardinality for each element referenced withing <AttributeRefs>,
1016 <PredicateRefs> and <RoleRefs>, with the exception of those elements that have attribute
1017 optional set to true. If unable to do so, it MUST return an appropriate error message to
1018 the client.

1019 The identity service MUST return only those digital subjects whose use policies are
1020 consistent with the <wsp:Policy> elements found in the <Interaction> element and
1021 individual filters.

1022 If the identity service cannot provide requested information due to use policy
1023 incompatibility, it MUST return an error indication to the client.

1024 If the identity service cannot provide requested information due to lack of user consent, it
1025 MUST return an error indication to the client.

1026 If the identity service cannot provide the requested information for other reasons, it MUST
1027 return an error indication to the client.

1028

1029 4. Appendix A

1030 4.1. DataType URIs

1031 Based on Section A.2 of the XACML 2.0 specification.

- 1032
- 1033 1. <http://www.w3.org/2001/XMLSchema#string>
- 1034 2. <http://www.w3.org/2001/XMLSchema#boolean>
- 1035 3. <http://www.w3.org/2001/XMLSchema#integer>
- 1036 4. <http://www.w3.org/2001/XMLSchema#double>
- 1037 5. <http://www.w3.org/2001/XMLSchema#time>
- 1038 6. <http://www.w3.org/2001/XMLSchema#date>
- 1039 7. <http://www.w3.org/2001/XMLSchema#dateTime>
- 1040 8. <http://www.w3.org/2001/XMLSchema#anyURI>
- 1041 9. <http://www.w3.org/2001/XMLSchema#hexBinary>
- 1042 10. <http://www.w3.org/2001/XMLSchema#base64Binary>
- 1043 11. <http://www.w3.org/TR/2002/WD-xquery-operators-20020816#dayTimeDuration>
- 1044 12. <http://www.w3.org/TR/2002/WD-xquery-operators-20020816#yearMonthDuration>
- 1045 13. urn:oasis:names:tc:xacml:1.0:data-type:x500Name
- 1046 14. urn:oasis:names:tc:xacml:1.0:data-type:rfc822Name
- 1047 15. urn:oasis:names:tc:xacml:2.0:data-type:ipAddress
- 1048 16. urn:oasis:names:tc:xacml:2.0:data-type:dnsName
- 1049
- 1050

1051 For the sake of improved interoperability, it is RECOMMENDED that all time references be
1052 in UTC time.

1053

1054 XACML defines three data-types; these are:

- 1055 “urn:oasis:names:tc:xacml:1.0:data-type:x500Name”,
- 1056 “urn:oasis:names:tc:xacml:1.0:data-type:rfc822Name”
- 1057 “urn:oasis:names:tc:xacml:2.0:data-type:ipAddress”
- 1058 “urn:oasis:names:tc:xacml:2.0:data-type:dnsName” and

1059 These types represent identifiers for subjects or resources and appear in several standard
1060 applications, such as TLS/SSL and electronic mail.

1061 **4.2. Comparison Operators**

1062

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 that 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
dynamic	Operator-dependent	operator value specified at run-time