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

NOTICE:

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

Deleted: 11
Deleted: 11
Deleted: 11
Deleted: 12
Deleted: 12
Deleted: 12
Deleted: 13
Deleted: 14
Deleted: 15
Deleted: 16
Deleted: 17
Deleted: 17
Deleted: 17
Deleted: 18
Deleted: 18
Deleted: 18
Deleted: 18
Deleted: 19
Deleted: 20
Deleted: 20
Deleted: 21
Deleted: 22
Deleted: 23
Deleted: 24
Deleted: 26
Deleted: 26
Deleted: 26

1 Introduction

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. By identity-related data we mean information associated with a *digital subject*. The requirements we have in mind primarily concern identity data required by the entity, but support is also provided for expressing the update of identity data and for search of digital subjects meeting certain criteria.

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

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

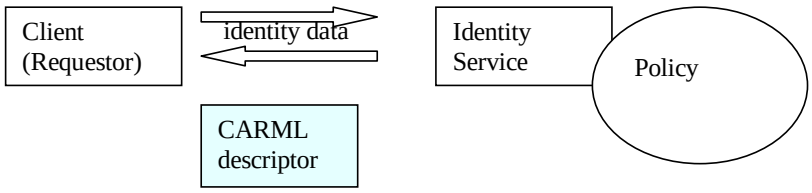

Figure 1

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

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

1. Given a digital subject, retrieve or read attributes, roles or predicate values associated with the subject

2. Given a digital subject, determine if certain predicates, roles, or attribute values are associated with it.

3. Given attribute values or roles, retrieve digital subjects that possess those values or roles

4. Given a set of attribute values or roles, request the creation of a digital subject associated with these values

5. Given a digital subject, request the update of attributes or roles associated with the digital subject

6. Given a digital subject, request that the digital subject be deleted.

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

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

1.1 Example

```
[a01] <carml:ClientAttrReq AppName="CARML Example" Description="Demonstrates features of
      CARML Schema" xmlns:carml="urn:igf:client:0.9:carml"
      xmlns:wsp="http://www.w3.org/ns/ws-policy"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:schemaLocation="urn:igf:client:0.9:carml igf-carml-09.xsd">
[a02]
[a03] <DataDefs>
[a04]
[a05]    <Attributes>
[a06]       <Attribute Cardinality="single" DataType="string"      DisplayName="Surname"
      Name="sn" />
[a07]       <Attribute Cardinality="single" DataType="string" Description="One or more
      names that are considered given names. The first name should be the preferred
      name." DisplayName="Given names" Name="givenname" />
[a08]       <Attribute Cardinality="single" DataType="urn:oasis:names:tc:xacml:1.0:data-
      type:rfc822Name" DisplayName="E-Mail" Name="mail" />
[a09]       <Attribute Cardinality="single" DataType="string" DisplayName="Telephone"
      Name="telephone" />
[a10]       <Attribute Cardinality="single" DataType="string" DisplayName="Last 4 Digits
      SSN" Name="Last4SSN" />
[a11]    </Attributes>
[a12]    <Predicates>
[a13]       <Predicate Description="For the jurisdiction of the user, a determination
      that the subject can travel alone." DisplayName="Adult" Name="IsAdult" />
[a14]       <Predicate Description="A resident of the EU" DisplayName="EU Resident"
      Name="IsEUResident" />
[a15]    </Predicates>
[a16] -  <Roles>
[a17]       <Role Description="Able to book business class tickets"
      DisplayName="Business Class FLyer" Name="BusinessClassFlyer" />
[a18]       <Role Description="The passenger's account is active." DisplayName="Account
      active" Name="IsActive" />
[a19]       <Role Description="Person is an employee" Name="Employee" />
[a20]       <Role Description="Person is a contractor" Name="Contractor" />
[a21]    </Roles>
[a22]    <Policies>
[a23]       <wsp:Policy Name="http://tempuri.org/" />
[a24]    </Policies>
[a25] </DataDefs>
[a26]
```

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

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

```
[a27]   <ReadInteraction Description="" Name="ReadProfile">
[a28]     <wsp:Policy Name="http://tempuri.org" />
[a29]     <AttributeRef Ref="#mail" />
[a30]     <AttributeRef Ref="#sn" />
[a31]     <AttributeRef Ref="#givenname" />
[a32]     <AttributeRef Ref="#telephone" Optional="true" />
[a33]     <PredicateRef Ref="#IsAdult" Optional="true" />
[a34]     <PredicateRef Ref="#IsEUResident" />
[a35]     <RoleRef Ref="#BusinessClassFlyer" />
[a36]   </ReadInteraction>
[a37]
[a38]   <FindInteraction Description="Locate user for authentication purposes."
      Name="LocateUser">
[a39]     <wsp:Policy Name="http://tempuri.org" />
[a40]     <Filter Match="all">
[a41]       <AttrRefFilter Ref="#mail" PrimaryKey="true" />
[a42]       <Filter Match="any">
[a43]         <RoleRefFilter Ref="#Employee" />
[a44]         <RoleRefFilter Ref="#Contractor" />
[a45]       </Filter>
[a46]     </Filter>
[a47]   </FindInteraction>
[a48]
[a49]   <SearchInteraction Name="SearchLastName" Description="Returns potential matches
      for a given surname">
[a50]     <AttributeRef Ref="#mail" />
[a51]     <AttributeRef Ref="#sn" />
[a52]     <Filter Match="all">
[a53]       <AttrRefFilter Ref="#sn" />
[a54]       <RoleRefFilter Ref="#IsActive" />
[a55]     </Filter>
[a56]   </SearchInteraction>
[a57]
[a58]   <CompareInteraction Name="VerifyIdentity" Description="Used to verify information
      provided by user">
[a59]     <Filter Match="all">
[a60]       <AttrRefFilter Ref="#Last4SSN" Operator="endswith" />
[a61]       <AttrRefFilter Ref="#mail" Operator="equals" />
[a62]     </Filter>
[a63]   </CompareInteraction>
[a64]
[a65]   <ModifyInteraction Name="UpdateTelephoneNumber">
[a66]     <AttributeRef Ref="#telephone" />
[a67]   </ModifyInteraction>
[a68]
[a69]   <AddInteraction Name="AddNewUser">
[a70]    <AttributeRef Ref="#mail" />
[a71]     <AttributeRef Ref="#sn" />
[a72]     <AttributeRef Ref="#givenname" />
[a73]     <AttributeRef Ref="#telephone" Optional="true" />
[a74]     <RoleRef Ref="#Employee" Optional="true" />
[a75]     <RoleRef Ref="#Contractor" Optional="true" />
[a76]   </AddInteraction>
[a77]
[a78]   <DeleteInteraction Name="UnRegisterUser" Description="User cannot use this
      service goingforward" />
[a79] </carml:ClientAttrReq>
```

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

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

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

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

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

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

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

1.2 Terminology

Conventional XML namespace prefixes are used throughout the listings in this specification to stand for their 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

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.

1.3 References

1.3.1 Normative References

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

[WS-Policy] **Web Services Policy 1.5 – Framework, October 2007.**
http://www.w3.org/TR/2004/REC-xmlschema-1-20041028/

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

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

1.3.2 Non-Normative References

None

1.4 Notation

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

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

2 Foundations

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

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

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

2.1 AttributeOrPredicateSuperType

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

Name

The name of the attribute, predicate or filter

DisplayName

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

Description

String description or definition of the attribute, predicate or filter

2.2 CardinalityType

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

```
                <enumeration value="zero"/>
                <enumeration value="single"/>
                <enumeration value="multiple"/>
            </restriction>
    </simpleType>
```

2.3 AttributeType

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

```
<complexType name="AttributeType">
    <complexContent>
        <extension base="carml:AttributeOrPredicateSuperType">
            <attribute name="Cardinality"
type="carml:CardinalityType" use="optional"/>
            <attribute name="DataType" type="anyURI"
use="optional" default="string"/>
        </extension>
    </complexContent>
</complexType>
```

Cardinality

Whether the attribute is zero, single or multi-valued

DataType

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

2.4 PredicateType

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

```
<complexType name="PredicateType">
    <complexContent>
        <extension
base="carml:AttributeOrPredicateSuperType"/>
    </complexContent>
</complexType>
```

2.5 RefType

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

```
<complexType name="RefType">
          <attribute name="Ref" type="anyURI" use="required"/>
          <attribute name="PolicyRef" type="anyURI"
use="optional"/>
          <attribute name="Optional" type="boolean" use="optional"
default="false"/>
          <attribute name="Description" type="string"
use="optional"/>
</complexType>
```

Ref

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

PolicyRef

URI of local or external privacy policy

Optional

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

2.6 FilterRefType

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

```
<complexType name="FilterRefType">
    <complexContent>
          <extension base="carml:RefType">
          <attribute name="Cardinality"
type="carml:CardinalityType" use="optional" default="single"/>
          <attribute name="PrimaryKey" type="boolean"
default="false"/>
          <attribute name="Operator" default="equals">
                <simpleType>
                      <restriction base="string">
                            <enumeration value="contains"/>
                            <enumeration value="doesnotcontain"/>
                            <enumeration value="dynamic"/>
                            <enumeration value="beginswith"/>
```

```
                            <enumeration value="endswith"/>
                            <enumeration value="equals"/>
                            <enumeration value="notequals"/>
                            <enumeration value="gt"/>
                            <enumeration value="lt"/>
                            <enumeration value="geq"/>
                            <enumeration value="leq"/>
                        </restriction>
                    </simpleType>
            </attribute>
            <attribute name="Name" type="ID" use="optional"/>
            </extension>
    </complexContent>
</complexType>
```

Cardinality

Whether the requestor provides single or multiple values

DataType

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

PrimaryKey

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

Operator

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

2.7 FilterType

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

```
<complexType name="FilterType">
    <choice maxOccurs="unbounded">
            <element name="AttrRefFilter" type="carml:FilterRefType"
minOccurs="0" maxOccurs="unbounded"/>
            <element name="RoleRefFilter" type="carml:RefType"
minOccurs="0" maxOccurs="unbounded"/>
            <element name="PredRefFilter" type="carml:RefType"
minOccurs="0" maxOccurs="unbounded"/>
            <element name="Filter" type="carml:FilterType"
minOccurs="0" maxOccurs="unbounded"/>
```

```
    </choice>
    <attribute name="Match" default="all">
          <simpleType>
                restriction base="string">
                      <enumeration value="any"/>
                      <enumeration value="all"/>
                </restriction>
          </simpleType>
    </attribute>
    <attribute name="Description" use="optional"/>
</complexType>
```

AttRefFilter

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

RoleRefFilter

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

PredRefFilter

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

Filter

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

Match

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

3 Client Attribute Requirements

```
<element name="ClientAttrReq">
   <!-- root element for a CARML declaration -->
   <complexType>
         <sequence>
               <element name="DataDefs">
               ...
               </element>
         <choice minOccurs="0" maxOccurs="unbounded">
               <element name="AddInteraction"
maxOccurs="unbounded">
               ...
               </element>
               <element name="DeleteInteraction"
type="carml:BaseInteractionType" maxOccurs="unbounded"/>
               ...
               </element>
               <element name="ReadInteraction"
maxOccurs="unbounded">
               ...
               </element>
                   <element name="ModifyInteraction"
maxOccurs="unbounded">
               ...
               </element>
               <element name="CompareInteraction" minOccurs="0"
maxOccurs="unbounded">
               ...
               </element>
               <element name="FindInteraction"
maxOccurs="unbounded">
               ...
               </element>
               <element name="SearchInteraction"
maxOccurs="unbounded">
               ...
               </element>
         </choice>
<!-- Application policy -->
         <choice minOccurs="0" maxOccurs="unbounded">
               <element ref="wsp:Policy"/>
               <element ref="wsp:PolicyReference"/>
         </choice>
</sequence>
   <attribute name="AppName" type="string" use="required"/>
   <attribute name="Description" type="string" use="optional"/>
   <attribute name="CarmlURI" type="anyURI" use="optional"/>
</complexType>
```

```
</element>
```

<ClientAttrReq> is the root element that captures the client attribute requirements of a specific entity. The requirements are captured by a set of zero or more interaction elements. Interaction elements include <AddInteraction>, <ReadInteraction>, <ModifyInteraction>, <UpdateInteraction>, <CompareInteraction>, <FindInteraction> and <SearchInteraction> elements. Each of these elements references attributes, predicates, roles and policies declared in the <DataDefs> element.
In some cases, only the <DataDefs> element may be present; this corresponds to a client or applications group publishing a list of standard or preferred attributes, predicates, roles and policies. Such a declaration might be used to publish a standard set of names and types for reference by other <ClientAttrReq> elements.

[PrivAssert] defines privacy policy assertions that express privacy constraints over the use of identity data. [WS-Policy] provides a general framework for expressing composite policies built out of atomic assertions.
The <wsp:Policy> or <wsp:PolicyReference> element carries policy assertions based on WS-Policy with atomic assertions drawn only from [PrivAssert]. These policies apply to all of the interactions defined within the <ClientAttrReq> element.

AppName

String name associated with <ClientAttrReq> element

CarmlURI

URI associated with the <ClientAttrReq> element

3.1 DataDefs

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

```
<element name="DataDefs">
    <complexType>
          <sequence>
          <element name="ExternalDataDefsRef" minOccurs="0"
maxOccurs="unbounded">
          <complexType>
                <attribute name="CarmlURI" type="anyURI"
use="required"/>
                <attribute name="AppName" type="string"
use="optional"/>
                <attribute name="ProcessNestedDefinitions"
type="boolean" default="true"/>
                <anyAttribute namespace="##any"
processContents="lax"/>
          </complexType>
          </element>
          <element name="Attributes">
          <complexType>
                <sequence>
                <element name="Attribute"
type="carml:AttributeType" minOccurs="0" maxOccurs="unbounded"/>
                </sequence>
          </complexType>
          </element>
          <element name="Predicates">
          <complexType>
          <sequence>
                <element name="Predicate"
type="carml:PredicateType" minOccurs="0" maxOccurs="unbounded"/>
          </sequence>
          </complexType>
          </element>
          <element name="Roles">
          <complexType>
          <sequence>
                <element name="Role" type="carml:PredicateType"
minOccurs="0" maxOccurs="unbounded"/>
          </sequence>
          </complexType>
          </element>
          <element name="Policies">
          <complexType>
          <sequence>
                <element ref="wsp:Policy" minOccurs="0"
maxOccurs="unbounded"/>
          </sequence>
          </complexType>
```

```
            </element>
            </sequence>
        </complexType>
</element>
```

3.1.1 ExternalDefsRef

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

CarmlURI

URI of referenced <ClientAttrReq> element

AppName

Optional name of the referenced <ClientAttrReq> element

ProcessNestedDefinitions

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

3.1.2 Attributes

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

3.1.3 Predicates

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

3.1.4 Roles

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

3.1.5 Policies

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

3.2 Interaction

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

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

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

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

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

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

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

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

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

3.2.1 BaseInteractionType

```
<complexType name="BaseInteractionType" abstract="true">
         <sequence>
               <!-- Holds interaction policies -->
               <choice minOccurs="0" maxOccurs="unbounded">
                     <element ref="wsp:Policy"/>
                     <element ref="wsp:PolicyReference"/>
               </choice>
         </sequence>
         <attribute name="Name" type="ID" use="required"/>
         <attribute name="Description" use="optional"/>
                     <attribute name="EntityName" type="NCName"
use="optional"/>
    </complexType>
```

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

3.2.2 AddInteraction

```
<element name="AddInteraction" maxOccurs="unbounded">
    <complexType>
```

```
          <complexContent>
               <extension base="carml:BaseInteractionType">
               <sequence>
               <element name="AttributeRef" type="carml:RefType"
minOccurs="0" maxOccurs="unbounded"/>
               <element name="RoleRef" type="carml:RefType"
minOccurs="0" maxOccurs="unbounded"/>
               </sequence>
               </extension>
          </complexContent>
    </complexType>
</element>
```

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

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

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

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

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

If the identity service cannot process the request due to use policy incompatibility, it MUST return an error indication to the client.
If the identity service cannot provide requested information due to lack of user consent, it MUST return an error indication to the client.

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

3.2.3 DeleteInteraction

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

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

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

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

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

3.2.4 ModifyInteraction

```
<element name="ModifyInteraction" maxOccurs="unbounded">
    <complexType>
          <complexContent>
                <extension base="carml:BaseInteractionType">
                <sequence>
                <element name="AttributeRef" type="carml:RefType"
minOccurs="0" maxOccurs="unbounded"/>
                <element name="RoleRef" type="carml:RefType"
minOccurs="0" maxOccurs="unbounded"/>
                </sequence>
                <attribute name = "Mode" type="carml:ModeType"
default="replace">
                </extension>
          </complexContent>
    </complexType>
</element>
```

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

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

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

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

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

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

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

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

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

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

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

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

3.2.5 ReadInteraction

```
<element name="ReadInteraction" maxOccurs="unbounded">
    <complexType>
          <complexContent>
          <extension base="carml:BaseInteractionType">
                <sequence>
                <element name="AttributeRef" type="carml:RefType"
minOccurs="0" maxOccurs="unbounded"/>
                <element name="PredicateRef" type="carml:RefType"
minOccurs="0" maxOccurs="unbounded"/>
                <element name="RoleRef" type="carml:RefType"
minOccurs="0" maxOccurs="unbounded"/>
</sequence>
          </extension>
          </complexContent>
    </complexType>
</element>
```

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

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

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

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

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

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

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

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

3.2.6 CompareInteraction

```
<element name="CompareInteraction" minOccurs="0"
maxOccurs="unbounded">
    <complexType>
          <complexContent>
                <extension base="carml:BaseInteractionType">
                <sequence>
                <element name="Filter" type="carml:FilterType"/>
                <!-- Must have one or more filters -->
                </sequence>
                </extension>
          </complexContent>
    </complexType>
</element>
```

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.

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.

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

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

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

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

3.2.7 FindInteraction

```
<element name="FindInteraction" maxOccurs="unbounded">
<complexType>
          <complexContent>
                <extension base="carml:BaseInteractionType">
                <sequence>
                <element name="AttributeRef" type="carml:RefType"
minOccurs="0" maxOccurs="unbounded"/>
                <element name="PredicateRef" type="carml:RefType"
minOccurs="0" maxOccurs="unbounded"/>
                <element name="RoleRef" type="carml:RefType"
minOccurs="0" maxOccurs="unbounded"/>
                <element name="Filter" type="carml:FilterType"/>
                <!-- Must have one or more filters -->
                </sequence>
                </extension>
          </complexContent>
    </complexType>
</element>
```

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

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

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

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

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

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

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

3.2.8 SearchInteraction

```
<element name="SearchInteraction" maxOccurs="unbounded">
    <complexType>
          <complexContent>
          <extension base="carml:BaseInteractionType">
          <sequence>
          <element name="AttributeRef" type="carml:RefType"
minOccurs="0" maxOccurs="unbounded"/>
          <element name="PredicateRef" type="carml:RefType"
minOccurs="0" maxOccurs="unbounded"/>
          <element name="RoleRef" type="carml:RefType"
minOccurs="0" maxOccurs="unbounded"/>
          <element name="Filter" type="carml:FilterType"/>
          <!-- Must have one or more filters -->
          </sequence>
          <attribute name="MaxSubjects" type="integer"
use="optional" default="100"/>
```

```
            <attribute name="PageSize" type="integer" use="optional"
default="1"/>
            </extension>
            </complexContent>
        </complexType>
</element>
```

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

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

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

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

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

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

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

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

4. Appendix A

4.1. DataType URIs

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

1. http://www.w3.org/2001/XMLSchema#string
2. http://www.w3.org/2001/XMLSchema#boolean
3. http://www.w3.org/2001/XMLSchema#integer
4. http://www.w3.org/2001/XMLSchema#double
5. http://www.w3.org/2001/XMLSchema#time
6. http://www.w3.org/2001/XMLSchema#date
7. http://www.w3.org/2001/XMLSchema#dateTime
8. http://www.w3.org/2001/XMLSchema#anyURI
9. http://www.w3.org/2001/XMLSchema#hexBinary
10. http://www.w3.org/2001/XMLSchema#base64Binary
11. http://www.w3.org/TR/2002/WD-xquery-operators-20020816#dayTimeDuration
12. http://www.w3.org/TR/2002/WD-xquery-operators-20020816#yearMonthDuration
13. urn:oasis:names:tc:xacml:1.0:data-type:x500Name
14. urn:oasis:names:tc:xacml:1.0:data-type:rfc822Name
15. urn:oasis:names:tc:xacml:2.0:data-type:ipAddress
16. urn:oasis:names:tc:xacml:2.0:data-type:dnsName

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

XACML defines three data-types; these are:
"urn:oasis:names:tc:xacml:1.0:data-type:x500Name",
"urn:oasis:names:tc:xacml:1.0:data-type:rfc822Name"
"urn:oasis:names:tc:xacml:2.0:data-type:ipAddress"
"urn:oasis:names:tc:xacml:2.0:data-type:dnsName" and
These types represent identifiers for subjects or resources and appear in several standard applications, such as TLS/SSL and electronic mail.

4.2. 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 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