Hierarchical Resource profile of XACML

Committee Draft 01, 30 September 2004

Document identifier:

access_control-xacml-2.0-hier_profile-spec-cd-01

Location:

http://docs.oasis-open.org/xacml/access_control-xacml-2.0-hier_profile-spec-cd-01.pdf

Editor:

Anne Anderson, Sun Microsystems (anne.anderson@sun.com)

Abstract:

This document provides a profile for the use XACML with resources that are structured as hierarchies. The profile addresses resources represented as nodes in XML documents or represented in some non-XML way. The profile covers identifying nodes in a hierarchy, requesting access to nodes in a hierarchy, and specifying policies that apply to nodes in a hierarchy.

Status:

This version of the specification is an approved Committee Draft within the OASIS Access Control TC.

Access Control TC members should send comments on this specification to the xacml@lists.oasis-open.org list. Others may use the following link and complete the comment form: http://oasis-open.org/committees/comments/form.php?wg_abbrev=xacml.

For information on whether any patents have been disclosed that may be essential to implementing this specification, and any offers of patent licensing terms, please refer to the Intellectual Property Rights section of the Access Control TC web page (http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=xacml).

For any errata page for this specification, please refer to the Access Control TC web page (http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=xacml).

Table of Contents

1 Introduction

It is often the case that a ***resource*** is organized as a hierarchy. Examples include file systems, XML documents, and organizations. This Profile specifies how XACML can provide ***access control*** for a ***resource*** that is organized as a hierarchy.

Why are ***resources*** organized as hierarchies special? First of all, policies over hierarchies frequently apply the same ***access controls*** to entire sub-trees of the hierarchy. Being able to express a single policy constraint that will apply to an entire sub-tree of ***nodes*** in the hierarchy, rather than having to specify a separate constraint for each ***node***, increases both ease of use and the likelihood that the policy will correctly reflect the desired ***access controls***. Another special characteristic of ***hierarchical resources*** is that access to one ***node*** may depend on the value of another ***node.*** For example, a medical patient might be granted access to the "diagnosis" ***node*** in a XML document medical record only if the patient's name matches the value in the "patient name" ***node.*** Where this is the case, the requested ***node*** can not be processed in isolation from the rest of the ***nodes*** in the hierarchy, and the PDP must have access to the values of other ***nodes***. Finally, the identity of ***nodes*** in a hierarchy often depends on the position of the ***node*** in the hierarchy; there also may be multiple ways to describe the identity of a single ***node.*** In order for policies to apply to ***nodes*** as intended, attention must be paid to consistent representations for the identity of the ***nodes.*** Otherwise, a requester may bypass ***access controls*** by requesting a ***node*** using an identity that differs from the one used by the policy.

In this Profile, a ***resource*** organized as a hierarchy may be a "tree" (a hierarchy with a single root) or a "forest" (a hierarchy with multiple roots), but the hierarchy may not have cycles. Another term for these two types of hierarchy is "Directed Acyclic Graph" or "DAG". All such ***resources*** are called ***hierarchical resources*** in this Profile. An XML document is always structured as a "tree". Other types of ***hierarchical resources,*** such as files in a file system that supports links, may be structured as "forests".

In this Profile, the ***nodes*** in a ***hierarchical resource*** are treated as individual ***resources.*** An ***authorization decision*** that permits ***access*** to an interior ***node*** does not imply that ***access*** to its descendant ***nodes*** is permitted***.*** An ***authorization decision*** that denies ***access*** to an interior ***node*** does not imply that ***access*** to its descendant ***nodes*** is denied***.***

There are three types of facilities specified in this Profile for dealing with ***hierarchical resources:***

- Representing the identity of a ***node.***
- Requesting access to a ***node***.
- Stating policies that apply to one or more ***nodes***.

Support for each of these facilities is optional.

This Profile addresses two ways of representing a hierarchical resource. In the first way, the hierarchy of which the node is a part is represented as an XML document that is included in the the Request, and the requested resource is represented as a node in that document. In the second way, the requested resource is not represented as a node in an XML document, and there is no representation of the hierarchy of which it is a part included in the Request. Note that the actual target resource in the first case need not be part of an XML document - it is merely represented that way in the Request. Likewise, the target resource in the second case might actually be part of an XML document, but is being represented in some other way in the Request. Thus there is no assumed correlation between the structure of the resource as represented in the Request and the actual structure of the physical resource being accessed.

Facilities for dealing with ***resources*** represented as ***nodes*** in XML documents can make use of the fact that the XML document itself is included in the ***decision request.*** [XPath] expressions can be used to reference ***nodes*** in this document in a standard way, and can provide unique representations for a given ***node*** in the document***.*** These facilities are not available for ***hierarchical resources*** that are not represented as XML documents. Other means must be provided in the case of such non-XML

resources for determining the location of the requested ***node*** in the hierarchy. In some cases this can be done by including the ***node's*** position in the hierarchy as part of the ***node's*** identity. In other cases, a ***node*** may have more than one normative identity, such as when the pathname of a file in a file system can include hard links. In such cases, the XACML ***PDP's*** Context Handler may need to supply the identities of all the ***node's*** ancestors. For all these reasons, the facilities for dealing with ***nodes*** in XML documents differ from the facilities for dealing with ***nodes*** in other ***hierarchical resources.***

In dealing with a ***hierarchical resource***, it may be useful to request ***authorization decisions*** for multiple ***nodes*** in the ***resource*** in a single ***decision request***. Ways to make such requests are specified in another Profile – the *Multiple Resource profile of XACML* [MULTIPLE]. That Profile also provides a way to return a single ***authorization decision*** when access to multiple ***nodes*** in a hierarchy is requested. Readers of this Profile are encouraged to become familiar with the *Multiple Resource profile of XACML.* This Profile may be considered to be layered on top of the Multiple Resource Profile, which in turn is layered on top of the behavior specified in the core XACML specification [XACML]. The functionality in this Profile MAY, however, be layered directly on the functionality in the core XACML specification.

This Profile for ***hierarchical resources*** assumes that all requests for ***access*** to multiple ***nodes*** in a ***hierarchical resource*** [MULTIPLE] have been resolved to individual requests for ***access*** to a single ***node***.

1.1 Terminology

Access - Performing an ***action***.

Access control - Controlling ***access*** in accordance with a ***policy.***

Action – An operation on a ***resource***.

Applicable policy - The set of ***policies*** and ***policy sets*** that governs ***access*** for a specific ***decision request.***

Attribute - Characteristic of a ***subject***, ***resource, action*** or ***environment*** that may be referenced in a ***predicate*** or ***target*** (see also – ***named attribute***) or provided in a ***context***. May also refer to an XML syntactic attribute, in which case the term will be qualified as “XML attribute.”

Authorization decision - The result of evaluating ***applicable policy,*** returned by the ***PDP*** to the ***PEP.*** A function that evaluates to "Permit”, “Deny”, “Indeterminate” or “NotApplicable", and (optionally) a set of ***obligations.***

Bag – An unordered collection of values, in which there may be duplicate values.

Context - The canonical representation of a ***decision request*** and an ***authorization decision.***

Decision – The result of evaluating a ***rule, policy*** or ***policy set.***

Decision request - The request by a ***PEP*** to a ***PDP*** to render an ***authorization decision.***

Hierarchical resource – A ***resource*** that is organized as a tree or forest (Directed Acyclic Graph) of individual ***resources*** called ***nodes.***

Node – An individual ***resource*** that is part of a ***hierarchical resource.***

Obligation - An operation specified in a ***policy*** or ***policy set*** that should be performed by the ***PEP*** in conjunction with the enforcement of an ***authorization decision.***

Policy - A set of ***rules,*** an identifier for the ***rule-combining algorithm*** and (optionally) a set of ***obligations.*** May be a component of a ***policy set.***

Policy administration point (PAP) - The system entity that creates a ***policy*** or ***policy set.***

Policy decision point (PDP) - The system entity that evaluates ***applicable policy*** and renders an ***authorization decision***. This term is defined in a joint effort by the IETF Policy Framework Working

Group and the Distributed Management Task Force (DMTF)/Common Information Model (CIM) in [RFC3198]. This term corresponds to "*Access Decision Function*" (ADF) in [ISO10181-3].

Policy enforcement point (PEP) - The system entity that performs ***access control***, by making ***decision requests*** and enforcing ***authorization decisions***. This term is defined in a joint effort by the IETF Policy Framework Working Group and the Distributed Management Task Force (DMTF)/Common Information Model (CIM) in [RFC3198]. This term corresponds to "*Access Enforcement Function*" (AEF) in [ISO10181-3].

Policy set* –** A set of ***policies, other ***policy sets,*** a policy-combining algorithm and {optionally} a set of ***obligations*.** May be a component of another ***policy set.***

Resource - Data, service or system component. The object for which ***access*** is requested in a ***decision request***.

1.2 Notation

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 behavior 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.

The phrase ***{Normative, but optional}*** means that the described functionality is optional for compliant XACML implementations, but, if the functionality is claimed as being supported according to this Profile, then it SHALL be supported in the way described.

```
Example code listings appear like this.
```

In descriptions of syntax, elements in angle brackets (“<”, “>”) are to be replaced by appropriate values, square brackets (“[“, “]”) enclose optional elements, elements in quotes are literal components, and “*” indicates that the preceding element may occur zero or more times.

2 Representing the identity of a node

{Normative}

In order for XACML ***policies*** to apply consistently to ***nodes*** in a ***hierarchical resource***, it is necessary for the ***nodes*** in that ***resource*** to be represented in a consistent way. If a ***policy*** refers to a ***node*** using one representation, but a ***request*** refers to the ***node*** using a different representation, then the ***policy*** will not apply, and security may be compromised.

The following sections describe RECOMMENDED representations for ***nodes*** in ***hierarchical resources.*** Alternative representations of ***nodes*** in a given ***resource*** are permitted so long as all ***Policy Administration Points*** and all ***Policy Enforcement Points*** that deal with that ***resource*** have contracted to use the alternative representation.

2.1 Nodes in XML documents

{Normative, but optional}

The following URI SHALL be used as the identifier for the functionality specified in this Section of this Profile: `urn:oasis:names:tc:xacml:2.0:profile:hierarchical:xml-node-id.`

The identity of a ***node*** in a ***resource*** that is represented as an XML document instance SHALL be an XPath expression that evaluates to exactly that one ***node*** in the copy of the ***resource*** that is contained in the `<ResourceContent>` element of the `<Resource>` element of the `<Request>`.

2.2 Nodes in resources that are not XML documents

{Normative, but optional}

The following URI SHALL be used as the identifier for the functionality specified in this Section of this Profile: `urn:oasis:names:tc:xacml:2.0:profile:hierarchical:non-xml-node-id.`

The identity of a ***node*** in a ***hierarchical resource*** that is not represented as an XML document instance SHALL be represented as a URI that conforms to [RFC2396]. Such URIs are of the following form.

```
<scheme> “:” <authority> “/” <pathname>
```

File system ***resources*** SHALL use the “`file:`” scheme. If no standard `<scheme>` for the ***resource*** type is specified in [RFC2396] or in a related standard for a registered URI scheme, then the URI SHALL use the “`file:`” scheme.

The `<pathname>` portion of the URI SHALL be of the form

```
<root name> [ “/” <node name> ]*
```

The sequence of `<root name>` and `<node name>` values SHALL correspond to the individual hierarchical component names of ancestors of the represented ***node*** along the path from a `<root>` ***node*** to the represented ***node***.

The following canonicalization SHALL be used.

- The encoding of the URI SHALL be UTF8.
- Case-insensitive portions of the URI SHALL be lower case.
- Escaping of characters SHALL conform to [RFC2396].
- The `<authority>` portion of the URI SHALL be specified and SHALL be the standard authority representation for the given ***resource*** type. Where the `<authority>` could be specified using either a Domain Name Service (DNS) [RFC1034] name or a numeric IPv4 or IPv6 address, the DNS name SHALL be used.

- The components of the `<pathname>` portion of the URI SHALL be specified using the canonical form for such path components at the `<authority>`.
- In accordance with [RFC2396], the separator character between hierarchical components of the `<pathname>` portion of the URI SHALL be the character "`/`". Sequences of the "`/`" character SHALL be resolved to a single "`/`". ***Node*** identities SHALL NOT terminate with the "`/`" character.
- The `<pathname>` SHALL contain no soft links.
- All `<pathname>` values SHALL be absolute.
- If there is more than one fully resolved, absolute path from a `<root> at` the `<authority>` to the represented ***node***, then a separate ***resource attribute*** with `AttributeId` "`urn:oasis:names:tc:xacml:1.0:resource:resource-id`" and DataType `http://urn:oasis:names:tc:xacml:1.0:data-type:anyURI` SHALL be present in the Request Context for each such path.

3 Requesting access to a node

{Normative}

In order for XACML ***policies*** to apply consistently to ***nodes*** in a ***hierarchical resource***, it is necessary for each request ***context*** that represents a request for ***access*** to a ***node*** in that ***resource*** to use a consistent description of that ***node access***. If a ***policy*** refers to certain expected ***attributes*** of a ***node***, but the request ***context*** does not contain those ***attributes,*** or if the ***attributes*** are not expressed in the expected way, then the ***policy*** may not apply, and security may be compromised.

The following sections describe RECOMMENDED request ***context*** descriptions of ***access*** to ***nodes*** in ***hierarchical resources.*** Alternative representations of such requests are permitted so long as all ***Policy Administration Points*** and all ***Policy Enforcement Points*** that deal with that ***resource*** have contracted to use the alternative representation***.***

3.1 Nodes in an XML document

{Normative, but optional}

The following URI SHALL be used as the identifier for the functionality specified in this Section of this Profile: `urn:oasis:names:tc:xacml:2.0:profile:hierarchical:xml-node-req.` The ***attributes*** with `AttributeIds` of "`urn:oasis::names:tc:xacml:2.0:resource:resource-parent`", "`urn:oasis::names:tc:xacml:2.0:resource:resource-ancestor`" and "`urn:oasis::names:tc:xacml:2.0:resource:resource-ancestor-or-self`" are optional to implement. If supported for use in resources represented as XML documents, the following URIs SHALL be used as identifiers for the functionality they represent: "`urn:oasis:names:tc:xacml:2.0:profile:hierarchical:xml-node-req:resource-parent`", "`urn:oasis:names:tc:xacml:2.0:profile:hierarchical:xml-node-req:resource-ancestor`", and "`urn:oasis:names:tc:xacml:2.0:profile:hierarchical:xml-node-req:resource-ancestor-or-self`".

In order to request ***access*** to a ***resource*** represented as a ***node*** in an XML document, the request ***context*** `<Resource>` element SHALL contain the following elements and XML attributes.

- A `<ResourceContent>` element that contains the entire XML document instance of which the requested ***node*** is a part.
- An `<Attribute>` element with an `AttributeId` of "`urn:oasis::names:tc:xacml:1.0:resource:resource-id`" and a `DataType` of "`urn:oasis:names:tc:xacml:2.0:data-type:xpath-expression`". The `<AttributeValue>` of this `<Attribute>` SHALL be an XPath expression whose context node SHALL be the one and only child of the `<ResourceContent>` element. This XPath expression SHALL evaluate to a nodeset containing the single ***node*** in the `<ResourceContent>` element that is the ***node*** to which ***access*** is requested. This `<Attribute>` MAY specify an `Issuer`.
- An `<Attribute>` element with an `AttributeId` of "`urn:oasis::names:tc:xacml:2.0:resource:resource-parent`" and a `DataType` of "`urn:oasis:names:tc:xacml:2.0:data-type:xpath-expression`". The `<AttributeValue>` of this `<Attribute>` SHALL be an XPath expression; the context node for this XPath expression SHALL be the one and only child of the `<ResourceContent>` element. This XPath expression SHALL evaluate to a nodeset containing the single ***node*** in the `<ResourceContent>` element that is the immediate parent of the ***node*** represented in the "`resource-id`" ***attribute.*** This `<Attribute>` MAY specify an `Issuer`.
- For each ***node*** in the XML document instance that is an ancestor of the ***node*** represented by the "`resource-id`" ***attribute***, an `<Attribute>` element with an `AttributeId` of "`urn:oasis::names:tc:xacml:2.0:resource:resource-ancestor`" and a `DataType` of

"urn:oasis:names:tc:xacml:2.0:data-type:xpath-expression". The <AttributeValue> of this <Attribute> SHALL be an XPath expression; the context node for this XPath expression SHALL be the one and only child of the <ResourceContent> element. This XPath expression SHALL evaluate to a nodeset containing the single ***node*** in the <ResourceContent> element that is the respective ancestor of the ***node*** represented in the "resource-id" ***attribute.*** For each "resource-parent" ***attribute,*** there SHALL be a corresponding "resource-ancestor" ***attribute.*** This <Attribute> MAY specify an Issuer.

- For each ***node*** in the XML document instance that is an ancestor of the ***node*** represented by the "resource-id" ***attribute,*** and for the "resource-id" ***node*** itself, an <Attribute> element with an AttributeId of "urn:oasis::names:tc:xacml:2.0:resource:resource-ancestor-or-self" and a DataType of "urn:oasis:names:tc:xacml:2.0:data-type:xpath-expression". The <AttributeValue> of this <Attribute> SHALL be an XPath expression; the context node for this XPath expression SHALL be the one and only child of the <ResourceContent> element. This XPath expression SHALL evaluate to a nodeset containing the single ***node*** in the <ResourceContent> element that is the respective ancestor of the ***node*** represented in the "resource-id" ***attribute,*** or that is the "resource-id" ***node*** itself***.*** For each "resource-parent" and "resource-id" ***attribute,*** there SHALL be a corresponding "resource-ancestor-or-self" ***attribute.*** This <Attribute> MAY specify an Issuer.

Additional ***attributes*** MAY be included in the <Resource> element. In particular, the following ***attribute*** MAY be included.

- An <Attribute> element with an AttributeId of "urn:oasis::names:tc:xacml:2.0:resource:document-id" and a DataType of "urn:oasis:names:tc:xacml:1.0:data-type:anyURI". The <AttributeValue> of this <Attribute> SHALL be a URI that identifies the XML document of which the requested ***resource*** is a part, and of which a copy is present in the <ResourceContent> element. This <Attribute> MAY specify an Issuer.

3.2 Nodes in a resource that is not an XML document

{Normative, but optional}

The following URI SHALL be used as the identifier for the functionality specified in this Section of this Profile: urn:oasis:names:tc:xacml:2.0:profile:hierarchical:non-xml-node-req. The ***attributes*** with AttributeIds of "urn:oasis::names:tc:xacml:2.0:resource:resource-parent", "urn:oasis::names:tc:xacml:2.0:resource:resource-ancestor", and "urn:oasis::names:tc:xacml:2.0:resource:resource-ancestor-or-self" are optional to implement. If supported for use in resources that are not represented as XML documents, the following URIs SHALL be used as identifiers for the functionality they represent: "urn:oasis:names:tc:xacml:2.0:profile:hierarchical:non-xml-node-req:resource-parent", "urn:oasis:names:tc:xacml:2.0:profile:hierarchical:non-xml-node-req:resource-ancestor", and "urn:oasis:names:tc:xacml:2.0:profile:hierarchical:non-xml-node-req:resource-ancestor-or-self".

In order to request ***access*** to a ***node*** in a ***hierarchical resource*** that is not represented as an XML document, the request ***context*** <Resource> element SHALL NOT contain a <ResourceContent> element. The request ***context*** <Resource> element SHALL contain the following elements and XML attributes. Note that a ***node*** in a ***hierarchical resource*** that is not represented as an XML document MAY have multiple parents. For example, in a file system that supports hard links, there may be multiple normative paths to a single file. Each such path MAY contain different sets of parents and ancestors.

- For each normative representation of the requested ***node,*** an <Attribute> element with AttributeId of "urn:oasis::names:tc:xacml:1.0:resource:resource-id". The <AttributeValue> of this <Attribute> SHALL be a unique, normative identity of the ***node*** to which ***access*** is requested. The DataType of this <Attribute> SHALL depend on the representation chosen for the identity of ***nodes*** in this particular ***resource***. This <Attribute> MAY specify an Issuer.

- For each immediate parent of the ***node*** specified in the "resource-id" ***attribute*** or ***attributes,*** and for each normative representation of that parent ***node,*** an <Attribute> element with AttributeId "urn:oasis::names:tc:xacml:2.0:resource:resource-parent". The <AttributeValue> of this <Attribute> SHALL be the normative identity of the parent ***node***. The DataType of this <Attribute> SHALL depend on the representation chosen for the identity of ***nodes*** in this particular ***resource***. This <Attribute> MAY specify an Issuer. If the requested ***node*** is part of a forest rather than part of a single tree, or if the parent ***node*** has more than one normative representation, there SHALL be at least one instance of this ***attribute*** for each parent along each path to the multiple roots of which the requested ***node*** is a descendant, and for each normative representation of each such parent.

- For each ancestor of the ***node*** specified in the "resource-id" ***attribute*** or ***attributes,*** and for each normative representation of that ancestor ***node,*** an <Attribute> element with AttributeId "urn:oasis::names:tc:xacml:2.0:resource:resource-ancestor". The <AttributeValue> of this <Attribute> SHALL be the normative identity of the ancestor ***node***. The DataType of this <Attribute> SHALL depend on the representation chosen for the identity of ***nodes*** in this particular ***resource***. This <Attribute> MAY specify an Issuer. For each "resource-parent" ***attribute***, there SHALL be a corresponding "resource-ancestor" ***attribute***. If the requested ***node*** is part of a forest rather than part of a single tree, or if the ancestor ***node*** has more than one normative representation, there SHALL be at least one instance of this ***attribute*** for each ancestor along each path to the multiple roots of which the requested ***node*** is a descendant, and for each normative representation of each such ancestor. The order of the values for this ***attribute*** do not necessarily reflect the position of each ancestor ***node*** in the hierarchy.

- For each ancestor of the ***node*** specified in the "resource-id" ***attribute*** or ***attributes,*** and for each normative representation of that ancestor ***node,*** and for each normative representation of the "resource-id" ***node*** itself, an <Attribute> element with AttributeId "urn:oasis::names:tc:xacml:2.0:resource:resource-ancestor-or-self". The <AttributeValue> of this <Attribute> SHALL be the respective normative identity of the ancestor ***node*** or of the "resource-id" ***node*** itself. The DataType of this <Attribute> SHALL depend on the representation chosen for the identity of ***nodes*** in this particular ***resource***. This <Attribute> MAY specify an Issuer. For each "resource-ancestor" and "resource-id" ***attribute***, there SHALL be a corresponding "resource-ancestor-or-self" ***attribute***. If the requested ***node*** is part of a forest rather than part of a single tree, or if the ancestor ***node*** has more than one normative representation, there SHALL be at least one instance of this ***attribute*** for each ancestor along each path to the multiple roots of which the requested ***node*** is a descendant, and for each normative representation of each such ancestor. The order of the values for this ***attribute*** do not necessarily reflect the position of each ancestor ***node*** in the hierarchy.

Additional ***attributes*** MAY be included in the <Resource> element.

4 Stating policies that apply to nodes

{Non-normative}

This Section describes various ways to specify a ***policy*** predicate that can apply to multiple ***nodes*** in a ***hierarchical resource***. This is not intended to be an exhaustive list.

4.1 Policies applying to nodes in any hierarchical resource

{Non-normative}

Resource attributes with the following `AttributeId` values, described in Section 6: *New attribute identifiers for hierarchical resources* of this Profile, MAY be used to state ***policies*** that apply to one or more ***nodes*** in any ***hierarchical resource.***

```
        urn:oasis:names:tc:xacml:2.0:resource:resource-parent
        urn:oasis:names:tc:xacml:2.0:resource:resource-ancestor
        urn:oasis:names:tc:xacml:2.0:resource:resource-ancestor-or-self
```

Note that a `<ResourceAttributeDesignator>` that refers to the “`resource-parent`”, “`resource-ancestor`”, or “`resource-ancestor-or-self`” ***attribute*** will return a bag of values representing all normative identities of all parents, ancestors, or ancestors plus the ***resource*** itself, respectively, of the ***resource*** to which ***access*** is being requested. The representations of the identities of these parents, ancestors, or self will not necessarily indicate the path from the root of the hierarchy to the respective parent, ancestor, or self unless the representation recommended in Section 3.2: *Nodes in a resource that is not an XML document* is used.

The standard XACML [XACML] bag and higher-order bag functions MAY be used to state ***policies*** that apply to one or more ***nodes*** in any ***hierarchical resource.*** The ***nodes*** used as arguments to these functions MAY be specified using a `<ResourceAttributeDesignator>` with the “`resource-parent`”, “`resource-ancestor`”, or “`resource-ancestor-or-self`” `AttributeId` value.

4.2 Policies applying only to nodes in XML documents

{Non-normative}

For ***hierarchical resources*** that are represented as XML document instances, the following function, described in the XACML 2.0 Specification [XACML] MAY be used to state ***policy*** predicates that apply to one or more ***nodes*** in that ***resource.***

```
        urn:oasis:names:tc:xacml:2.0:function:xpath-node-match
```

The standard XACML `<AttributeSelector>` element MAY be used in ***policies*** to refer to all or portions of a ***resource*** represented as an XML document and contained in the `<ResourceContent>` element of a request ***context.***

The standard XACML [XACML] bag and higher-order bag functions MAY be used to state ***policies*** that apply to one or more ***nodes*** in a resource represented as an XML document***.*** The ***nodes*** used as arguments to these functions MAY be specified using an `<AttributeSelector>` that selects a portion of the `<ResourceContent>` element of the `<Resource>` element`.`

4.3 Policies applying only to nodes in non-XML resources

{Non-normative}

For ***hierarchical resources*** that are not represented as XML document instances, and where the URI representation of ***nodes*** specified in Section 2 of this Profile is used, the following functions described in the XACML 2.0 Specification [XACML] MAY be used to state ***policies*** that apply to one or more ***nodes*** in that ***resource.***

```
      urn:oasis:names:tc:xacml:1.0:function:anyURI-equal
      urn:oasis:names:tc:xacml:1.0:function:regexp-uri-match
```

5 New DataType

{Normative, but optional}

The following value for the XML DataType attribute value MAY be supported for use with ***hierarchical resources*** represented as XML documents. Support for this DataType is required in order to support Section 3.1 in this Profile.

5.1 xpath-expression

The DataType represented by the following URI represents an XPath expression. ***Attribute*** values having this DataType SHALL be strings that are to be interpreted as XPath expressions. The result of evaluating such an ***attribute*** SHALL be the nodeset that results from evaluating the XPath expression. If the string is not a valid XPath expression, the result of evaluating the ***attribute*** SHALL be Indeterminate.

```
Urn:oasis:names:tc:xacml:2.0:data-type:xpath-expression.
```

6 New attribute identifiers

{Normative, but optional}

6.1 document-id

The following identifier indicates the identity of the XML document that represents the hierarchy of which the requested ***resource*** is a part, and of which a copy is present in the `<ResourceContent>` element. Whenever ***access*** to a ***node*** in a ***resource*** represented as an XML document is requested, one or more instances of an ***attribute*** with this `AttributeId` MAY be provided in the `<Resource>` element of the request ***context.*** The `DataType` of these ***attributes*** SHALL be "`urn:oasis:names:tc:xacml:1.0:data-type:anyURI`".

```
urn:oasis:names:tc:xacml:2.0:resource:document-id
```

6.2 resource-parent

The following identifier indicates one normative identity of one parent ***node*** in the tree or forest of which the requested ***node*** is a part. Whenever ***access*** to a ***node*** in a ***hierarchical resource*** is requested, one instance of an ***attribute*** with this `AttributeId` SHALL be provided in the `<Resource>` element of the request ***context*** for each normative representation of each ***node*** that is a parent of the requested ***node.***

```
urn:oasis:names:tc:xacml:2.0:resource:resource-parent
```

6.3 resource-ancestor

The following identifier indicates one normative identity of one ancestor ***node*** in the tree or forest of which the requested ***node*** is a part. Whenever ***access*** to a ***node*** in a ***hierarchical resource*** is requested, one instance of an ***attribute*** with this `AttributeId` SHALL be provided in the `<Resource>` element of the request ***context*** for each normative representation of each ***node*** that is an ancestor of the requested ***node***.

```
urn:oasis:names:tc:xacml:2.0:resource:resource-ancestor
```

6.4 resource-ancestor-or-self

The following identifier indicates one normative identity of one ancestor ***node*** in the tree or forest of which the requested ***node*** is a part, or one normative identity of the requested ***node*** itself. Whenever ***access*** to a ***node*** in a ***hierarchical resource*** is requested, one instance of an ***attribute*** with this `AttributeId` SHALL be provided in the `<Resource>` element of the request ***context*** for each normative representation of each ***node*** that is an ancestor of the requested ***node,*** and for each normative representation of the requested ***node*** itself.

```
urn:oasis:names:tc:xacml:2.0:resource:resource-ancestor-or-self
```

7 New profile identifiers

{normative}

The following URI values SHALL be used as identifiers for the functionality specified in various Sections of this Profile:

Section 2.1: *Nodes in XML documents*

```
urn:oasis:names:tc:xacml:2.0:profile:hierarchical:xml-node-id
```

Section 2.2: *Nodes in resources that are not XML documents*

```
urn:oasis:names:tc:xacml:2.0:profile:hierarchical:non-xml-node-id
```

Section 3.1: *Nodes in an XML document*

```
urn:oasis:names:tc:xacml:2.0:profile:hierarchical:xml-node-req
```

Support for the “`resource-parent`”, “`resource-ancestor`”, and “`resource-ancestor-or-self`” ***attributes*** is optional within this Section, so these have separate identifiers:

```
urn:oasis:names:tc:xacml:2.0:profile:hierarchical:xml-node-req:resource-parent
```

```
urn:oasis:names:tc:xacml:2.0:profile:hierarchical:xml-node-req:resource-ancestor
```

```
urn:oasis:names:tc:xacml:2.0:profile:hierarchical:xml-node-req:resource-ancestor-or-self
```

Section 3.2: *Nodes in a resource that is not an XML document*

```
urn:oasis:names:tc:xacml:2.0:profile:hierarchical:non-xml-node-req
```

Support for the “`resource-parent`”, “`resource-ancestor`”, and “`resource-ancestor-or-self`” ***attributes*** is optional within this Section, so these have separate identifiers:

```
urn:oasis:names:tc:xacml:2.0:profile:hierarchical:non-xml-node-req:resource-parent
```

```
urn:oasis:names:tc:xacml:2.0:profile:hierarchical:non-xml-node-req:resource-ancestor
```

```
urn:oasis:names:tc:xacml:2.0:profile:hierarchical:non-xml-node-req:resource-ancestor-or-self
```

8 References

[ISO10181-3] ISO/IEC JTC 1, *Information technology -- Open Systems Interconnection -- Security frameworks for open systems: Access control framework,* ISO/IEC 10181-3:1996, 1996.

[RFC1034] P. Mockapetris, *DOMAIN NAMES – CONCEPTS AND FACILITIES,* IETF RFC 1034, November 1987, ftp://ftp.isi.edu/in-notes/rfc1034.txt

[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.

[RFC2396] T. Berners-Lee, et al., Uniform Resource Identifiers (URI): Generic Syntax, http://www.ietf.org/rfc/rfc2396.txt, IETF RFC 2396, August 1998.

[RFC3198] A. Westerinen, et al., *Terminology for Policy-Based Management, http://www.ietf.org/rfc/rfc3198.txt*, IETF RFC 3198, November 2001.

[MULTIPLE] A. Anderson, ed., *Multiple Resource profile of XACML,* Committee Draft 01, 30 September 2004, *http://docs.oasis-open.org/xacml/access_control-xacml-2.0-mult_profile-spec-cd-01.pdf*

[XACML] S. Godik, T. Moses, eds., *OASIS eXtensible Access Control Markup Language (XACML) Version 2.0,* Committee Draft 01, 16 September 2004, *http://docs.oasis-open.org/xacml/access_control-xacml-2.0-core-spec-cd-01.pdf*.

[XPath] *XML Path Language (XPath),* Version 1.0, W3C Recommendation 16, November 1999. Available at http://www.w3.org/TR/xpath

A. Acknowledgments

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

- Frank Siebenlist, Argonne National Laboratory
- Daniel Engovatov, BEA Systems, Inc.
- Hal Lockhart, BEA Systems, Inc.
- Rebekah Metz, Booz Allen Hamilton
- Ronald Jacobson, Computer Associates
- Tim Moses, Entrust
- Simon Godik, GlueCode Software
- Bill Parducci, GlueCode Software
- Michiharu Kudo, IBM
- Michael McIntosh, IBM
- Anthony Nadalin, IBM
- Steve Anderson, OpenNetwork
- Anne Anderson, Sun Microsystems
- Seth Proctor, Sun Microsystems
- Polar Humenn, Syracuse University
- Edward Coyne, Veterans Health Administration

B. Revision History

Rev	Date	By Whom	What
CD-01	30 Sept 2004	Anne Anderson	Committee Draft

C. Notices