

Hierarchical Resources: Non-XML Resource Use Case

Working Draft 01, 17 June 2004

Document identifier:

xacml-profile-hierarchical-resources-nonXML-1.0-draft01

Location:

http://www.oasis-open.org/committees/documents.php?wg_abbrev=xacml

Editor:

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

Contributors:

Frank Siebenlist, Argonne National Laboratory

Daniel Engovatov, BEA

Hal Lockhart, BEA

Tim Moses, Entrust

Simon Godik, GlueCode Software

Bill Parducci, GlueCode Software

Michiharu Kudo, IBM

Michael McIntosh, IBM

Anthony Nadalin, IBM

Steve Anderson, Open Network

Seth Proctor, Sun Microsystems

Polar Humenn, Syracuse University

Abstract:

This document provides a use case using the XACML Profile for Hierarchical Resources with a hierarchical resource that is not an XML document.

Status:

This version of the document is a working draft of the Committee. As such, it is expected to change before becoming part of any standard. Committee members should send comments on this specification to the xacml@lists.oasis-open.org list. Others should use the comment form at http://www.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 XXXX TC web page (<http://www.oasis-open.org/committees/xacml/ipr.php>).

36 **Table of Contents**

37 1 Introduction.....3

38 1.1 Terminology.....3

39 2 Use Case: A Portal-Managed Directory.....4

40 3 Problem Statement.....5

41 4 XACML Solutions..... 6

42 4.1 Solution 1: Policy predicates for each directory object.....6

43 4.2 Solution 2: resource-ancestor and resource-parent attributes.....7

44 4.3 Solution 3: anyURI-match function.....8

45 5 References..... 10

46

47 1 Introduction

48 This document provides a use case and solutions for using XACML to protect a hierarchical resource
49 that is not an XML document. The solutions make use of functionality described in the *XACML Profile*
50 *for Hierarchical Resources* [HIER].

51 **This document is non-normative.** It provides guidance and suggestions, but does not set limits on
52 valid uses of XACML.

53 1.1 Terminology

54 The policy examples in this document assume the following XML Internal Entities have been defined.

```
55 ^lt;!ENTITY action-id "urn:oasis:names:tc:xacml:1.0:action:action-id">  
56 ^lt;!ENTITY and "urn:oasis:names:tc:xacml:1.0:function:and">  
57 ^lt;!ENTITY anyURI-match  
58     "urn:oasis:names:tc:xacml:2.0:function:anyURI-match">  
59 ^lt;!ENTITY not "urn:oasis:names:tc:xacml:1.0:function:not">  
60 ^lt;!ENTITY or "urn:oasis:names:tc:xacml:1.0:function:or">  
61 ^lt;!ENTITY permit-overrides "urn:oasis:names:tc:xacml:1.0:rule-  
62     combining-algorithm:permit-overrides">  
63 ^lt;!ENTITY resource-ancestor  
64     "urn:oasis:names:tc:xacml:2.0:resource:resource-ancestor">  
65 ^lt;!ENTITY resource-id  
66     "urn:oasis:names:tc:xacml:1.0:resource:resource-id">  
67 ^lt;!ENTITY resource-parent  
68     "urn:oasis:names:tc:xacml:1.0:resource:resource-parent">  
69 ^lt;!ENTITY string "http://www.w3.org/2001/XMLSchema#string">  
70 ^lt;!ENTITY string-is-in  
71     "urn:oasis:names:tc:xacml:1.0:function:string-is-in">  
72 ^lt;!ENTITY string-one-and-only  
73     "urn:oasis:names:tc:xacml:1.0:function:string-one-and-only">  
74 ^lt;!ENTITY subject-id  
75     "urn:oasis:names:tc:xacml:1.0:subject::subject-id">
```

77 For example, “&string” is equivalent to “http://www.w3.org/2001/XMLSchema#string”.

78

2 Use Case: A Portal-Managed Directory

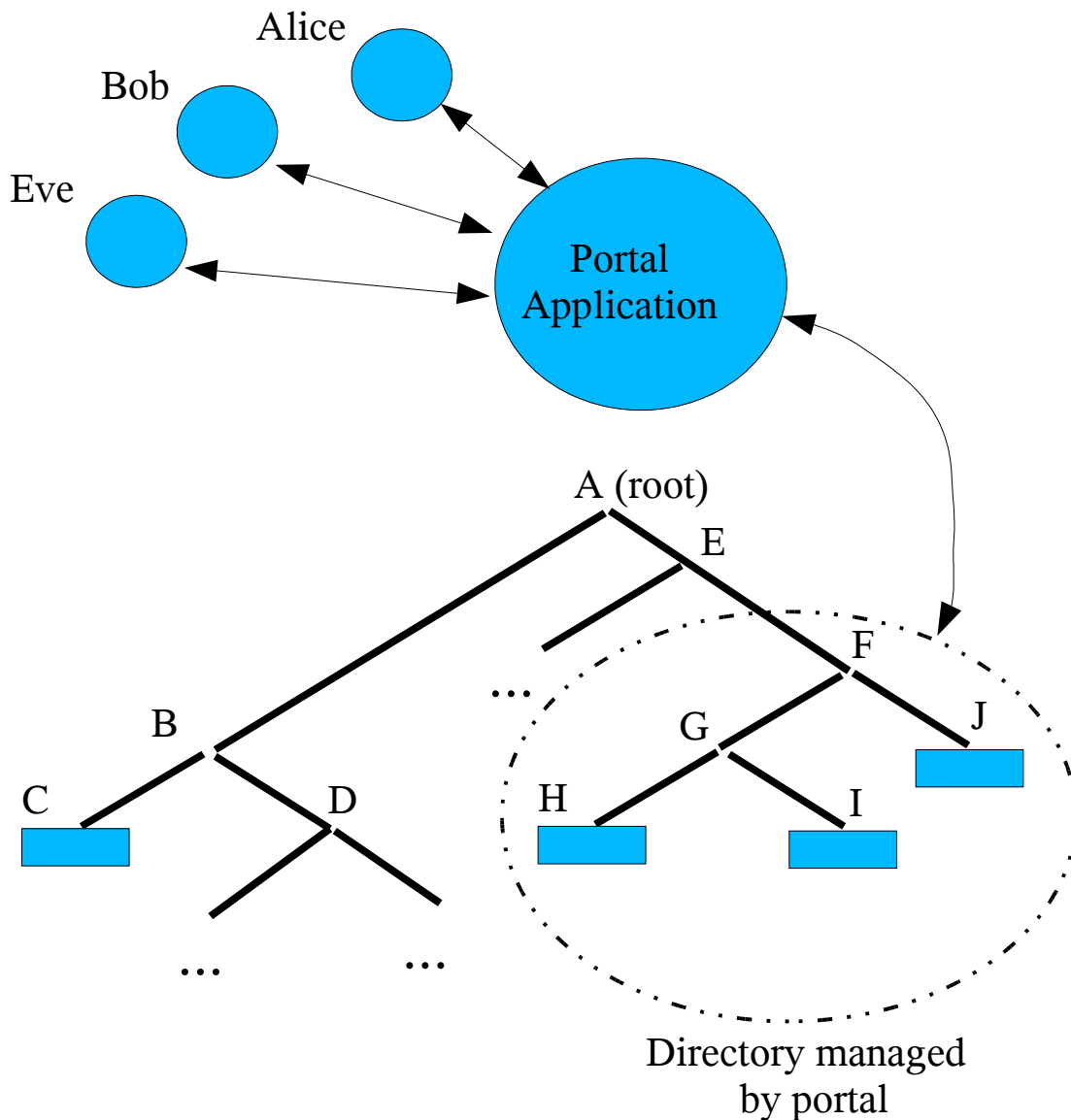
79

In this use case, a portal application serves as a front-end for client access to a directory in a Unix File System (UFS). The portal application is responsible for accessing objects in its directory on behalf of clients, but only if the access conforms to the enterprise policy. The portal application itself has full operating system permissions on objects in its directory, and has “execute” rights on the path from the root directory to its own directory. No other identities have rights in its directory, and the portal application has no rights in directories other than its own.

85

An illustration of the use case is shown in Drawing 1 below.

86



Drawing 1 Portal-Managed Directory

3 Problem Statement

87

88 We assume that the portal application has a Policy Enforcement Point (PEP) that traps each client
89 attempt to access an object in the managed directory. The PEP sends a description of the access
90 attempt to a Policy Decision Point (PDP), which compares the access attempt to its policies and returns
91 an access decision of "Permit" or "Deny". The PEP enforces the access decision, and will perform the
92 requested access for the client only if the access decision was "Permit".

93 The enterprise Policy Administration Point (PAP) will create the policies that will be used by the portal
94 application's PDP in making its policy decisions. The problem is how to express these policies.

95 As a very simple example, assume the PAP wants to enforce the following rules:

- 96 • Anyone can "read" directory F. That is, anyone can view the identities of the immediate child objects
97 in directory F. In addition,
- 98 • Alice can "read" any file or sub-directory under directory F except for file I
- 99 • Bob can "read" only files H and I.

100 This policy could be expressed as a decision table as follows:

Object	Read Permission
Directory F	anyone
Directory G	Alice
File H	Alice, Bob
File I	Bob
File J	Alice

101 The next sections will describe various ways of expressing this policy in XACML.

4 XACML Solutions

102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161

XACML provides several ways to solve the problem of expressing policies for the portal-managed directory. Some of these solutions are described below.

Note that in this use case, access control by XACML policies is limited by the privileges enforced by the operating system. For example, if the operating system does not allow the portal application to read a given file, then the portal application will be unable to read the file on behalf of a given client even if there is an XACML policy that would return “Permit” for that access. As another example, if the operating system grants a client access to read a file, then the portal application will be unable to prevent the client from reading the file directly (that is, not through the portal application).

4.1 Solution 1: Policy predicates for each directory object

The simplest solution is to include predicates in the XACML policy for each object in the managed directory. For each of the objects in the managed directory – directories F, G, and files H, I, and J – the policy will explicitly describe the conditions under which access is allowed.

An example of a policy implementing the decision table shown in Section 3 follows.

```
<Policy PolicyId="PortalPolicy" RuleCombiningAlgId="permit-overrides;">
  <Target>
    <Actions>
      <Action>
        <ActionMatch MatchId="string-is-in;">
          <AttributeValue DataType="string;">read</AttributeValue>
          <ActionAttributeDesignator AttributeId="action-id;"
            DataType="string;"/>
        </ActionMatch>
      </Action>
    </Actions>
  </Target>
  <Rule RuleId="Directory:F" Effect="Permit">
    <Condition FunctionId="string-is-in;">
      <AttributeValue DataType="string;">F</AttributeValue>
      <ResourceAttributeDesignator AttributeId="resource-id;"
        DataType="string;"/>
    </Condition>
  </Rule>
  <Rule RuleId="Directory:G" Effect="Permit">
    <Condition FunctionId="and;">
      <Apply FunctionId="string-is-in;">
        <AttributeValue DataType="string;">Alice</AttributeValue>
        <SubjectAttributeDesignator AttributeId="subject-id;"
          DataType="string;"/>
      </Apply>
      <Apply FunctionId="string-is-in;">
        <AttributeValue DataType="string;">G</AttributeValue>
        <ResourceAttributeDesignator AttributeId="resource-id;"
          DataType="string;"/>
      </Apply>
    </Condition>
  </Rule>
  <Rule RuleId="File:H" Effect="Permit">
    <Condition FunctionId="and;">
      <Apply FunctionId="or;">
        <Apply FunctionId="string-is-in;">
          <AttributeValue DataType="string;">Alice</AttributeValue>
          <SubjectAttributeDesignator AttributeId="subject-id;"
            DataType="string;"/>
        </Apply>
        <Apply FunctionId="string-is-in;">
          <AttributeValue DataType="string;">Bob</AttributeValue>
          <SubjectAttributeDesignator AttributeId="subject-id;"
            DataType="string;"/>
        </Apply>
      </Apply>
    </Condition>
  </Rule>
</Policy>
```

```

162     </Apply>
163     <Apply FunctionId="&string-is-in;">
164         <AttributeValue DataType="&string;">H</AttributeValue>
165         <ResourceAttributeDesignator AttributeId="&resource-id;"
166     DataType="&string;">
167     </Apply>
168     </Condition>
169 </Rule>
170 <Rule RuleId="File:I" Effect="Permit">
171     <Condition FunctionId="&and;">
172         <Apply FunctionId="&string-is-in;">
173             <AttributeValue DataType="&string;">Bob</AttributeValue>
174             <SubjectAttributeDesignator AttributeId="&subject-id;"
175     DataType="&string;">
176     </Apply>
177         <Apply FunctionId="&string-is-in;">
178             <AttributeValue DataType="&string;">I</AttributeValue>
179             <ResourceAttributeDesignator AttributeId="&resource-id;"
180     DataType="&string;">
181     </Apply>
182     </Condition>
183 </Rule>
184 <Rule RuleId="File:J" Effect="Permit">
185     <Condition FunctionId="&and;">
186         <Apply FunctionId="&string-is-in;">
187             <AttributeValue DataType="&string;">Alice</AttributeValue>
188             <SubjectAttributeDesignator AttributeId="&subject-id;"
189     DataType="&string;">
190     </Apply>
191         <Apply FunctionId="&string-is-in;">
192             <AttributeValue DataType="&string;">G</AttributeValue>
193             <ResourceAttributeDesignator AttributeId="&resource-id;"
194     DataType="&string;">
195     </Apply>
196     </Condition>
197 </Rule>
198 </Policy>

```

199 There are other ways this policy could be expressed, but the important point is that, with this solution,
200 each file and directory under directory F must be mentioned specifically in the policy. If new directories
201 and files are created in the future under directory F, then new predicates will need to be supplied in order
202 for clients to have access to them.

203 4.2 Solution 2: resource-ancestor and resource-parent attributes

204 This solution uses two new XACML AttributeIds defined in the XACML Profile for Hierarchical
205 Resources:

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

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

208 The “resource-ancestor” AttributeId includes all ancestors of the specific node to which access is
209 requested (i.e. the Attribute with AttributeId of
210 “urn:oasis:names:tc:xacml:2.0:resource:resource-id”). The “resource-parent” AttributeId
211 includes the parents¹ of the “resource-id” node.

212 Using these two new AttributeId values allows the policy to be stated more compactly. One way of
213 stating the policy using these AttributeId values is as follows.

```

214 <Policy PolicyId="PortalPolicy" RuleCombiningAlgId="&permit-overrides;">
215     <Target>
216         <Actions>
217             <Action>
218                 <ActionMatch MatchId="&string-is-in;">

```

1 “resource-parent” will include more than a single “parent” node only if the “resource-id” node is in a hierarchy that is in the form of a “forest” rather than a “tree”.

```

219         <AttributeValue DataType="&string;">read</AttributeValue>
220         <ActionAttributeDesignator AttributeId="&action-id;"
221         DataType="&string;">/>
222         </ActionMatch>
223         </Action>
224         </Actions>
225     </Target>
226     <Rule RuleId="Descendants:of:F" Effect="Permit">
227         <Condition FunctionId="&and;">
228             <Apply FunctionId="&string-is-in;">
229                 <AttributeValue DataType="&string;">Alice</AttributeValue>
230                 <SubjectAttributeDesignator AttributeId="&subject-id;"
231                 DataType="&string;">/>
232             </Apply>
233             <Apply FunctionId="&string-is-in;">
234                 <AttributeValue DataType="&string;">F</AttributeValue>
235                 <ResourceAttributeDesignator AttributeId="&resource-ancestor;"
236                 DataType="&string;">/>
237             </Apply>
238             <Apply FunctionId="&not;">
239                 <Apply FunctionId="&string-is-in;">
240                     <AttributeValue DataType="&string;">I</AttributeValue>
241                     <ResourceAttributeDesignator AttributeId="&resource-id;"
242                     DataType="&string;">/>
243                 </Apply>
244             </Apply>
245         </Condition>
246     </Rule>
247     <Rule RuleId="Children:of:G" Effect="Permit">
248         <Condition FunctionId="&and;">
249             <Apply FunctionId="&string-is-in;">
250                 <AttributeValue DataType="&string;">Bob</AttributeValue>
251                 <SubjectAttributeDesignator AttributeId="&subject-id;"
252                 DataType="&string;">/>
253             </Apply>
254             <Apply FunctionId="&string-is-in;">
255                 <AttributeValue DataType="&string;">G</AttributeValue>
256                 <ResourceAttributeDesignator AttributeId="&resource-parent;"
257                 DataType="&string;">/>
258             </Apply>
259         </Condition>
260     </Rule>
261 </Policy>

```

262 In order to use this method, each file and directory MUST have a unique name. If directory F had a sub-
263 directory also named F, for example, a request for an immediate child of the higher directory "F" would
264 get the same privileges as a request for an immediate child of the lower directory "F". One way to
265 ensure that each node in the hierarchy has a unique name is to use the URI representation for nodes
266 described in the *XACML Profile for Hierarchical Resources* [HIER].

267 Please note that this method may grant privileges to new directories and files that may be created in the
268 future under the existing directories. For example, if a new directory is created under directory F, then
269 this policy will give Alice the right to read anything in that new directory. This may or may not be the
270 intent of the policy writer, so caution is urged.

271 4.3 Solution 3: anyURI-match function

272 This solution uses a function defined in XACML 2.0 with the FunctionId
273 "urn:oasis:names:tc:xacml:2.0:function:anyURI-match". This solution requires that the
274 resources be represented using URIs that reflect the position of each resource in the hierarchy. For
275 example, directory F will be identified as "file://example.com/F" and file H will be identified as
276 "file://example.com/F/G/H". A description of this naming scheme is included in the *XACML Profile for*
277 *Hierarchical Resources* [HIER].

278 This solution solves the problem of assigning unique names to each file and directory, as well as
279 allowing use of the "anyURI-match" function. Nodes named using the URI representation may also use

280 the “resource-ancestor” and “resource-parent” AttributeIds described in Solution 2.

```
281 <Policy PolicyId="PortalPolicy" RuleCombiningAlgId="&permit-overrides;">
282   <Target>
283     <Actions>
284       <Action>
285         <ActionMatch MatchId="&string-is-in;">
286           <AttributeValue DataType="&string;">read</AttributeValue>
287           <ActionAttributeDesignator AttributeId="&action-id;"
288           DataType="&string;" />
289         </ActionMatch>
290       </Action>
291     </Actions>
292   </Target>
293   <Rule RuleId="Descendants:of:F" Effect="Permit">
294     <Condition FunctionId="&and;">
295       <Apply FunctionId="&string-is-in;">
296         <AttributeValue DataType="&string;">Alice</AttributeValue>
297         <SubjectAttributeDesignator AttributeId="&subject-id;"
298         DataType="&string;" />
299       </Apply>
300       <Apply FunctionId="&anyURI-match;">
301         <AttributeValue
302         DataType="&string;">http://www.example.com/F*</AttributeValue>
303         <Apply FunctionId="&string-one-and-only;">
304           <ResourceAttributeDesignator AttributeId="&resource-id;"
305           DataType="&string;" />
306         </Apply>
307       </Apply>
308       <Apply FunctionId="&not;">
309         <Apply FunctionId="&string-is-in;">
310           <AttributeValue DataType="&string;">I</AttributeValue>
311           <ResourceAttributeDesignator AttributeId="&resource-id;"
312           DataType="&string;" />
313         </Apply>
314       </Apply>
315     </Condition>
316   </Rule>
317   <Rule RuleId="Children:of:G" Effect="Permit">
318     <Condition FunctionId="&and;">
319       <Apply FunctionId="&string-is-in;">
320         <AttributeValue DataType="&string;">Bob</AttributeValue>
321         <SubjectAttributeDesignator AttributeId="&subject-id;"
322         DataType="&string;" />
323       </Apply>
324       <Apply FunctionId="&anyURI-match;">
325         <AttributeValue
326         DataType="&string;">http://www.example.com/F/G/*</AttributeValue>
327         <Apply FunctionId="&string-one-and-only;">
328           <ResourceAttributeDesignator AttributeId="&resource-id;"
329           DataType="&string;" />
330         </Apply>
331       </Apply>
332     </Condition>
333   </Rule>
334 </Policy>
```

335 Please note that this method may grant privileges to new directories and files that may be created in the
336 future under the existing directories. For example, if a new directory is created under directory F, then
337 this policy will give Alice the right to read anything in that new directory. This may or may not be the
338 intent of the policy writer, so caution is urged.

339

5 References

340

[HIER]

A. Anderson,ed., *XACML Profile for Hierarchical Resources*, Working Draft 04, 3
June 2004, <http://www.oasis->

341

[open.org/committees/documents.php?wg_abbrev=xacml](http://www.oasis-open.org/committees/documents.php?wg_abbrev=xacml)

342

343 **A. Revision History**

344 *[This appendix is optional, but helpful. It should be removed for specifications that are at OASIS*
345 *Standard level. Set the number format for the Rev and Date fields as you wish (select the desired string*
346 *and choose Format>Number Format...); the examples below are user-defined formats.]*
347

Rev	Date	By Whom	What
01	17 June 2004	Anne Anderson	Initial version.

348

B. Notices

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

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

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

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

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

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