

1

2

3

4

OASIS EXTENSIBLE ACCESS CONTROL MARKUP LANGUAGE (XACML)

5

6

TECHNICAL COMMITTEE

7

8

ISSUES LIST

9

10

VERSION 03

11

FEBRUARY 18, 2002

12

Ken Yagen, Editor

13

draft-xacml-issues-03.doc

14	PURPOSE.....	4
15	INTRODUCTION	4
16	USE CASE ISSUES.....	5
17	Group 1: Group Name	5
18	DESIGN ISSUES	5
19	Group 1: Group Name	5
20	POLICY MODEL ISSUES	5
21	Group 1: Rules.....	5
22	ISSUE:[PM-1-01: Negative Authorizations]	5
23	ISSUE:[PM-1-01-A: Implementing global deny and Meta-Policies]	6
24	ISSUE:[PM-1-02: Post-Conditions]	11
25	ISSUE:[PM-1-03: Post-Conditions as a term]	14
26	ISSUE:[PM-1-04: References to attributes in XACML predicates]	15
27	ISSUE:[PM-1-05: how NOT-APPLICABLE impacts a combinator expression]	15
28	ISSUE:[PM-1-06: result of <N-OF n=0> combinator expression]	18
29	ISSUE:[PM-1-07: How can the set of combinators be extended?]	18
30	ISSUE:[PM-1-08: syntax for <applicablePolicyReference>]	19
31	Group 2: Applicable Policy	19
32	ISSUE:[PM-2-01: Referencing Multiple Policies]	19
33	ISSUE:[PM-2-02: Target Specification]	20
34	ISSUE:[PM-2-03: Meaningful Actions]	20
35	ISSUE:[PM-2-04: Indexing Policy]	20
36	ISSUE:[PM-2-05: Ensuring Completeness]	21
37	ISSUE:[PM-2-06: Encapsulation of XACML policy (was Policy Security)]	22
38	ISSUE:[PM-2-07: valueRef type]	22
39	Group 3: Policy Composition	22
40	ISSUE:[PM-3-01: Combining Policy Elements]	23
41	ISSUE:[PM-3-02: Specifying Policy Outcome]	23
42	ISSUE:[PM-3-03: multiple Base Policies]	24
43	ISSUE:[PM-3-03: default PDP result]	24
44	Group 4: Syntax	25
45	ISSUE:[PM-4-01: Triplet Syntax (was Syntactic Sugar)]	25
46	ISSUE:[PM-4-02: Policy names as URIs]	25
47	ISSUE:[PM-4-03: Required type in policy]	26
48	ISSUE:[PM-4-04: syntax extension]	26
49	ISSUE:[PM-4-05: Policy Name a URI]	26
50	ISSUE:[PM-4-06: Comment element]	26
51	ISSUE:[PM-4-07: policy element in a rule]	27
52	ISSUE:[PM-4-08: XML elements include xsi:type]	27
53	ISSUE:[PM-4-09: complex types]	27
54	ISSUE:[PM-4-10: preserve PAP identity]	27
55	Group 5: SAML Related.....	28
56	ISSUE:[PM-5-01: Non-SAML Input]	28
57	ISSUE:[PM-5-02: Wildcards on Resource Hierarchies]	28
58	ISSUE:[PM-5-03: Roles and Group Hierarchies]	29
59	ISSUE:[PM-5-04: SAML Assertions URI]	29
60	ISSUE:[PM-5-05: XPath]	30
61	ISSUE:[PM-5-06: Multiple actions in single request]	30
62	ISSUE:[PM-5-07: Delegation]	31
63	ISSUE:[PM-5-08: saml:Action is a "string"]	32
64	ISSUE:[PM-5-09: saml:AuthorizationQuery requires actions]	33
65	ISSUE:[PM-5-10: single subject in AuthorizationQuery]	33

draft-xacml-issues-03.doc

66	ISSUE:[PM-5-11:XACML container in SAML].....	34
67	ISSUE:[PM-5-12:derive attribute from saml:AttributeValueType].....	34
68	ISSUE:[PM-5-13: Base Policy supplied as part of AuthorizationDecisionQuery]	34
69	Group 6: Predicate Cononicalization	35
70	ISSUE:[PM-6-01: SAML Assertions URI]	35
71	Group 7: Extensibility	35
72	ISSUE:[PM-7-01: XACML extensions]	35
73	MISCELLANEOUS ISSUES	36
74	Group 1: Glossary	36
75	ISSUE:[MI-1-01: Consistency]	36
76	Group 2: Conformance	36
77	ISSUE:[MI-2-01: Successfully Using]	36
78	Group 3: Patents, IP	37
79	ISSUE:[MI-3-01: XrML]	37
80	Group 4: Other Standards	38
81	ISSUE:[MI-4-01: RuleML]	38
82	ISSUE:[MI-4-02: RAD]	38
83	ISSUE:[MI-4-03: DSML].....	39
84	ISSUE:[MI-4-04: Java Security Model]	40
85	DOCUMENT HISTORY	40
86		

87 Purpose

88 This document catalogs issues for the eXtensible Access Control Markup Language (XACML)
89 developed the Oasis eXtensible Access Control Markup Language Technical Committee.

90 Introduction

91 The issues list presented here documents issues brought up in response to draft documents as
92 well as other issues mentioned on the xacml mailing list, in conference calls, and in other venues.
93 The structure of this document was taken from the Security Assertion Markup Language
94 (SAML) Issues List document maintained at the Security Services Technical Committee
95 document repository. Each issue is formatted as follows:

96 ISSUE:[Document/Section Abbreviation-Issue Number: Short name] Issue long description.
97 Possible resolutions, with optional editor resolution Decision

98 The issues are informally grouped according to general areas of concern. For this document, the
99 "Issue Number" is given as "#-##", where the first number is the number of the issue group.

100 To make reading this document easier, the following convention has been adopted for shading
101 sections in various colors.

102 Gray is used to indicate issues that were previously closed.

103 Blue is used to indicate issues that have been flagged as ready to close in the most recent
104 revision. These require review and voting by the committee and they can be closed.

105 Yellow is used to indicated issues which have recently been created or modified or are actively
106 being debated.

107 Other open issues are not marked, i.e. left white.

108 Issues with lengthy write-ups, that have been closed "for some time" will be removed from this
109 document, in order to reduce its overall size. The headings, a short description and resolution
110 will be retained. All vote summaries from closed issues will also be removed.

Use Case Issues

Group 1: Group Name

Design Issues

Group 1: Group Name

Policy Model Issues

Group 1: Rules

ISSUE:[PM-1-01: Negative Authorizations]

Authorizations can be either positive (permit) or negative (deny). Should we allow both?

See also PM-1-01-A which was split off from this issue.

Potential Resolutions:

[Michiharu] There seems to be agreement on the fact that the core schema should support positive authorizations only. Negative ones are supported as an extension.

[Tim] XACML shall address the requirement for "negative rules" by means of an "and-not-or" construct. [PM-1-01]

[Tim] We use a construct of the following form ...

```
<and>
  <rule1/><rule2/><rule3/>
  <not>
    <or>
      <rule4/><rule5/>
    </or></not></and>
```

Rule4 and rule5 specify circumstances under which, if either were to hold, access is to be denied. While rule1, rule 2 and rule3 specify circumstances, all of which must hold if access is to be granted.

Champion: Michiharu

Status: Open

ISSUE:[PM-1-01-A: Implementing global deny and Meta-Policies]

Implementing global "deny" semantics using schema 0.8 and meta-policies

[Anne] USE CASE: policy is to deny access to Principal "Anne Anderson" under all conditions. The policy is distributed across many sub-policies, which are all combined to produce the global policy that is to be applied.

Michiharu's concern was with needing to put something like

```
<not><equal>
  <valueRef entity="principal">saml:Subject/NameIdentifier/Name</valueRef>
  <value>"Anne Anderson"</value>
</equal></not>
```

Into every sub-policy if there was no global "deny" syntax.

My proposed solution depends on the idea of having meta-policies. I think meta-policies solve multiple problems:

1. "Where do I get policies",
 2. Knowing when you have obtained all the relevant policies,
 3. Knowing how to combine policies
 4. being able to implement global "deny" and meta-policies does not introduce any new syntax.
- It is just very explicit in specifying what "applicable policy" means.

Potential Resolutions:

[Anne] Each PDP (or PRP) needs to be configured with a single policy that serves as that PDP's "meta-policy". The syntax of this single policy is exactly that in 0.8.

This "meta-policy" determines where and under what conditions various sub-policies are retrieved. I may not be using <externalFunction> correctly, or the subpolicies may need more enclosing namespace information, but I hope these examples will give the idea. The final example shows how global "deny" semantics are implemented.

EXAMPLE SIMPLE META-POLICY FOR DISTRIBUTED POLICIES:

```
<?xml version="1.0" encoding="UTF-8"?>
<applicablePolicy xmlns=... issuer="<identity that ultimately controls policy for this PDP>"
policyName="...">
  <!-- target omitted, since this policy applies to all targets -->
  <policy>
    <and>
```

```

169     <externalFunction>http://www.site1/policy1.xml</externalFunction>
170     <externalFunction>http://www.site2/policy2.xml</externalFunction>
171     ...
172 </and>
173 </policy>
174 </applicablePolicy>

```

175 What is found at each of the <externalFunction> locations is another <applicablePolicy>, which
 176 may be more specific as to which resources it applies to (that applicablePolicy in turn may refer
 177 to still other policies). If one of these <applicablePolicy> elements does not apply to the current
 178 request, then the result is "does not apply" and does not affect the result of the <and> evaluation.

179 META-POLICY THAT USES SUB-POLICIES BASED ON RESOURCE

```

180 <?xml version="1.0" encoding="UTF-8"?>
181 <applicablePolicy xmlns=... issuer="<identity that ultimately controls policy for this PDP>"
182   policyName="...">
183   <!-- target omitted, since this policy applies to all targets -->
184   <policy>
185     <or>
186       <and>
187         <equal>
188           <valueRef>saml:Resource</valueRef>
189           <value>"file:/host1/*"</value>
190         </equal>
191         <externalFunction>http://www.site1/policy1.xml</externalFunction>
192       </and>
193       <and>
194         <equal>
195           <valueRef>saml:Resource</valueRef>
196           <value>"file:/host2/*"</value>
197         </equal>
198         <externalFunction>http://www.site2/policy2.xml</externalFunction>
199       </and>
200     ...
201   </or>
202 </policy>
203 </applicablePolicy>

```

204 META-POLICY THAT IMPLEMENTS GLOBAL DENY SEMANTICS

```

205 <?xml version="1.0" encoding="UTF-8"?>
206 <applicablePolicy xmlns=... issuer="<identity that ultimately controls policy for this PDP>"
207   policyName="...">

```

```

208 <!-- target omitted, since this policy applies to all targets -->
209 <policy>
210   <and>
211     <not>
212       <equal>
213         <valueRef entity="principal">saml:Subject/NameIdentifier/Name</valueRef>
214         <value>"Anne Anderson"</value>
215       </equal>
216     </not>
217   <or>
218     <and>
219       <equal>
220         <valueRef>saml:Resource</valueRef>
221         <value>"file:/host1/*"</value>
222       </equal>
223       <externalFunction>http://www.site1/policy1.xml</externalFunction>
224     </and>
225     <and>
226       <equal>
227         <valueRef>saml:Resource</valueRef>
228         <value>"file:/host2/*"</value>
229       </equal>
230       <externalFunction>http://www.site2/policy2.xml</externalFunction>
231     </and>
232     ...
233   </or>
234 </and>
235 </policy>
236 </applicablePolicy>

```

237 For administrative ease in a more realistic situation, the set of globally denied attribute/value
 238 combinations would be placed in one <externalFunction> policy.

239 [Ernesto] I support this proposal. I believe it could deal smoothly with the distributed scenario
 240 Anne described many times during the last conference call. It goes in the same direction of a
 241 previous suggestion of mine (deal with composition and distributed deployment at the
 242 ApplicablePolicy level), but does it far better. However, I would suggest some minor
 243 observations/amendments (otherwise there is no fun :-))

244 1. Maybe this is trivial, but any change to the current schema should keep policies fully
 245 embeddable in the Applicable policy element, besides being able to point to them using external
 246 functions. In simple environments there will be only one local policy, stated in a single
 247 document.

2. I happen not to like very much using the word "meta-policy" to describe this proposal, for several reasons some of which would be too long to explain in this message. Basically, I regard Anne's technique mainly as a way to define how a global policy can be deployed in distributed, independently maintained retrieval units. In passing, it also solves the problem of stating which criterion should be applied to compose the outcome of such units (this is essential when "deny" is a possible outcome, as the criterion may have an impact on what actually needs to be retrieved), but I cannot convince myself this requirement is equally important. I believe (but would like to hear the opinion of the industrial researchers on this one) that there will be a default policy composition technique that will be used 99.9% of the times. Therefore, in the schema I would prefer to concentrate the deployment description functionality in a new element, perhaps called "ApplicablePolicies", possibly defined as an extension of the base (Applicable)Policy type. This element could optionally (via an attribute) specify the composition criterion as well. Tim, what are your views?

[Hal] I am not sure if I agree with Anne's approach. I certainly like it better than the alternative proposed. I actually thought we had previously agreed that there had to be some rules (policy) for determining how independently created policies should be combined to achieve an authorization decision.

Instead of meta-policy, which I think Ernesto fears will be take to mean "more abstract policy" or "policy about policy", perhaps something like Policy Federation Rules would be better.

It seems to me the key issues are:

1. Where and how are PFR specified? Anne's approach is a distinct XML document, which must be consistent throughout the policy federation. This seems reasonable to me.

2. What are the possible PFR's? I think "AND" is impractical, and "OR" is most likely, however some kind of best-match-to-target is conceivable although perhaps too expensive to implement in practice.

3. Do all legal PFR's have to support all decision strategies? I have been thinking about this and I think the right approach is to explicitly call out the possible decision strategies and for each legal PFR state which can or cannot be used.

Here's what I have so far on decision strategies.

Strategy I - Basic

1. Collect all applicable policies
2. Obtain all required inputs
3. Evaluate all policies
4. Apply PFR to resolve conflicting results

Strategy II - Optimized

1. Collect all applicable policies
2. Use PFR to create equivalent combined policy
3. Evaluate policies incrementally, gathering inputs as needed, defer evaluations based on inputs requirements (this for example allows "lazy authentication" where authentication is not done if the result can be determined without it)
4. Once the result is known, stop evaluation

Strategy III- Incremental collection

1. Collect "some" policies
2. Obtain required inputs
3. Evaluate current policy set
4. Use PFR to combine latest results with previous results (if any)
5. If result is known, stop evaluation
6. If not all policies have been collected, repeat previous steps

These are all the possibilities I can think of. Can anyone think of others? I think anything proposed to date works equally for I and II, but not all work for III. However, we may find future possibilities that only work for one of them.

To answer Ernesto's question, our product uses "OR" for authorization decisions and "AND" for audit decisions and there have been no complaints. However we do not have post conditions, which may change things.

As far as the global deny, I would like to understand the requirements better. It seems the problem Anne is trying to solve is "master policy admin can globally deny regardless of what the policy combining rules are."

Is this the right problem to solve? If an "OR" combining rule is used (which I happen to think is the most common case) then any admin can implement a global deny without any special machinery. I think the example given is a red herring to some extent, because the right way to cut off an individual user is to change their attributes at the Attribute Authority or revoke their credentials.

The problem I see is that most evaluation engines will want to use a relatively fixed decision strategy in order to optimize it according to the criteria that apply in that environment. Finding it out in the middle of policy evaluation will interfere with this goal.

[Michiharu] I also support Anne's proposal. I think this technique deal with the distributed scenario nicely. I said the similar idea that uses an external function to call sub applicable policies in the policy model con-call on Dec. 17 but Anne's description is much more concrete and easy to understand. For the global deny policy, I agree that this technique is useful to specify the global deny semantics. If this technique is agreed, we may need more intuitive name for the externalFunction.

[Pierangela] I agree with the fact that the current proposal is able to implement the global deny scenario. No doubt about that: if you restrictions (i.e., the deny you want to enforce) ANDED with the other possible policies nobody will be able to overrule your restrictions.

The reason why I am not too excited with the current proposal is that it seems perfectly fine for communicating policies, but it seems complex to manage.

First of all you have to make sure that the applicable policy is in a single place (sure possibly using URL of other policies) but you cannot allow overlapping targets (which seemed to be the case till now, I believe).

Second the priority of your rules is explicitly managed with the policy definition, which may make administration heavy. Who is in charge of specifying the applicable policy? This will be the only one able to specify global deny: if understand Tim/Anne's proposals correctly possible negative authorizations in other policies have the effect only within that policy (this is fine with me, it seems conceptually clean).

Now for instance, suppose you want to enforce a situation in which any of us can grant authorizations and, possibly denials, for some access and a denial-take-precedence policy should be enforced (meaning it sufficient that one of us says "deny (because of a negative authorization), and the access should be rejected. How do you enforce this? You cannot have the different administrators operate on the applicable policy (meaning actually have writing privilege on that document).

Champion: Anne

Status: Open

ISSUE:[PM-1-02: Post-Conditions]

The current schema [Tim, Jan.3] mentions post-conditions, distinguishing between external and internal, depending on whether their execution requires dialoging with external entities. The current schema suggests (via a comment) that post-conditions can be expressed as invocations of SOAP services. Post-conditions are still to be discussed in details: what is their semantics; how are they executed? A complication of post-conditions associated with a rule involves the distributed scenario (see POLICY COMPOSITION issue). In fact, if I say that a post-condition should be applied whenever a rule fires then I have to evaluate *all* rules. A possible way to overcome this problem is to consider that post-conditions associated with the authorizations that

were evaluated to get to an access decision should be executed [Tim]. Note: a possible drawback of this approach is that deterministic behavior may be lost. For instance, there may be N rules applying to an access. If the evaluation of 1 of them brings to a ``permit" decision (so there is no need to evaluate the others). Then, you would ignore the post conditions possibly associated with the other N-1. Different execution of the same request on the same state could then have a different behavior (because a different rule is considered as authorizing the request).

[Tim] The alternative view is that post-conditions must be executed if and only if the associated rule contributes to the permit decision.

[Polar] What is the purpose for actions (i.e. these post conditions) after checking a policy? What types of actions are allowed? Do they change the state of the policy?

[Pierangela] examples that were brought up for post-conditions were things like "logging the request", essentially they are actions that the system executes in response to granting an access, or simply having evaluated the authorizations (discussion on the specific behavior is still open).

Do they change the state of the policy? If you mean the set of rules I guess the answer is no (they should not change the rules). But again, post-conditions are one of the issues which have not discussed fully.

[Polar] Well, I had originally thought that a "post-condition" would be something that would be true if the policy evaluated to true according to its input. That is, a "post-condition" should be a logical consequence, but maybe not fully derivable by all available information. This post-condition would merely be some advice to the evaluator.

Such as Policy stating that:

Subject is in Role of MissileLauncher to the Resource of Missile on Action Launch.

Post-condition Subject is dangerous.

I really don't like the fact that these post conditions mandate that some generic operation be performed, i.e. it could be used to alter state, especially the state of the policy.

[Simon] Post-condition is executed after the rule fires and does not affect grant/deny

Outcome of the rule. With this definition we can not predict which post condition(s) will be executed for a given authorization request. This is not desirable. One way to make post-conditions predictable is to associate post condition not with a rule but with the outcome of grant or deny, e.g.:

on_grant do_something
on_deny do_something

That means every time any subject is granted (or denied) action on any resource all post-conditions listed in on_grant (or on_deny) will be predictably executed. On_grant and on_deny

Colors: Gray Blue Yellow

383 post-conditions could be associated with specific action, subject, and resource triplet, meaning
384 that given post-condition will be executed every time subject is granted or denied permission to
385 access resource.

386 on_grant(action, subject, resource) do_something;
387 on_deny(action, subject, resource) do_something;

388 [John]
389 > Post-condition is executed after the rule fires and does not affect
390 > grant/deny outcome of the rule.

391 I thought this was only true of *external* post-conditions? I thought that an internal post-
392 condition must be executed (by the PDP) BEFORE the response is asserted, and therefore does
393 affect the outcome...

394 The spec says:

395 "...Post-condition - A process specified in a rule that must be completed in conjunction with
396 access. There are two types of post-condition: an internal post-condition must be executed by the
397 PDP prior to the issuance of a "permit" response, and an external post-condition must be
398 executed by the PEP prior to permitting access..."

399 I'm assuming that the "musts" here imply that the required actions are successfully executed. Is
400 this not the case?

401 [Simon] The way I remember post-conditions discussions is that outcome of internal post
402 condition does not affect the outcome of azn decision, i.e., first grant (or deny) is computed and
403 then internal post-condition is executed. If, for example, pdp fails to add a record to the log it
404 still returns computed outcome (grant or deny) to the pep. So the internal post-condition may not
405 be successfully executed by the pdp.

406 [Tim] This can be accomplished with the current syntax.

407 applicablePolicy/policy/rule+post-condition

408 This post-condition is executed if access is permitted.

409 applicablePolicy/policy/not/Rule+post-condition

410 This post-condition is executed if access is denied.

411 [Bill]

412 If given this:

413 > With this definition we can not predict which post condition(s) will be

414 > executed for a given
415 > Authorization request. This is not desirable.
416 'do_something' cannot be guaranteed:
417 > on_grant(action, subject, resource) do_something;
418 > on_deny(action, subject, resource) do_something;
419 Because that would require acknowledgement that it occurred (implying dependence on
420 grant/deny). Sounds like 'post condition' in this sense is more like 'post request'.
421 [Hal] I clearly remember that the sense of the group was that the PDP MUST insure that an
422 internal post condition occurs, but not necessarily before the permit decision is returned. Post
423 conditions were never considered optional. They are just as required for "permit" as pre-
424 conditions are. That was the rationale for the name.
425 Potential Resolutions:
426 [Tim] XACML shall require the PDP/PEP to execute just those post-conditions that accompany
427 the rules that contribute to the "permit" decision. [PM-1-02]
428 Champion: Simon
429 Status: Open
430 [ISSUE:\[PM-1-03: Post-Conditions as a term\]](#)
431 [Bill] I know that it is late to bring this up, but I find the term 'post condition' unintuitive.
432 Typically, this phrase means the *state* of something after an action, not something to be acted
433 upon. It seems that the way we are using the term implies quite a bit about the context of what is
434 being done. (post what? where?) I think this is being demonstrated by the discussions
435 surrounding the scope of said phrase. In my mind, it would seem that something like 'adjunct
436 policy' or 'adjunct policy condition' would be more appropriate?
437 [Pierangela] I share this feeling (incidentally, I brought it up in the last conference call, and also
438 in previous once). I was interpreting them more as "actions" than "conditions".
439 [Pierangela] in today's TC conference call, some people mentioned that "action" is already used
440 with different semantics (=the operation the principal is requesting). That's true, so we should
441 find another term. The point is, however, that the semantics of "post conditions" now seems
442 really to be a reaction of the system, not the evaluation of a state, so terminology should reflect
443 the semantics.
444 Potential Resolutions:

1. adjunct policy
2. adjunct policy condition
3. actions

Champion: Bill

Status: Open

ISSUE:[PM-1-04:References to attributes in XACML predicates]

What information needs to be provided in order to refer to an attribute in an XACML policy predicate?

Potential Resolutions:

Proposed Resolution:

References to attributes associated with the access request in XACML predicates consist of a URI to a document instance that contains the value of the attribute to be evaluated, a URI for the schema for the document, a schema-dependent path for locating a particular attribute instance in the document according to the schema, and an optional name for the Attribute Authority trusted to assign values for this attribute. The AA is located using the PKI with which the PDP is configured.

Champion: Anne

Status: Ready to Close

ISSUE:[PM-1-05: how NOT-APPLICABLE impacts a combinator expression]

A "combinator expression" is a combination of predicates, where possible combinators are <AND>, <OR>, <NOT>, <N-OF>, <ORDERED-[AND|OR|N-OF]>. This list of Combinators can be extended.

Example:

<AND>

predicate1,

predicate2,

predicate3

</AND>

The issue occurs when one or more of the predicates in the list returns a result of NOT-APPLICABLE (this can occur if the predicate is a <referencedPolicy>). What should the result of the combinator expression be? What if ALL predicates in the combinator expression return NOT-APPLICABLE?

Potential Resolution:

[Anne]

a) Any predicate evaluating to NOT-APPLICABLE is logically removed from the combinator expression.

Example: if predicate3 in the example above returned a result of NOT-APPLICABLE, then the combinator expression is the result of

<AND>

predicate1,

predicate2

<AND>

b) An empty combinator expression has the following results:

<AND></AND> -> TRUE

<OR></OR> -> FALSE

<NOT></NOT> -> TRUE

<N-OF></N-OF> -> FALSE

<ORDERED-[whatever]> has same result as [whatever] above. Extended combinators must define the result of an empty expression.

Example: If predicates 1, 2, and 3 in the example above all evaluate to NOT-APPLICABLE, then the combinator expression is <AND></AND>, and the result is TRUE.

b)-alternative: An empty combinator expression has a result of NOT-APPLICABLE.

[Polar] It's sort of like Anne's alternative #2 below with a couple of differences.

First, NOT-APPLICABLE (or Inapplicable?) and Error, are values that do not have an XML representation and are merely a artifact of evaluating policy expressions.

I propose the following consistent semantic model.

T = true, F = false, N = NOT-APPLICABLE, E = Error

Colors: Gray Blue Yellow

The basic crux is that getting a NOT-APPLICABLE in the equation is as if its the NOT-APPLICABLE value isn't even there. For instance,

$$(\text{and } x \text{ N } y) = (\text{and } x \text{ } y)$$

$$(\text{or } x \text{ N } y) = (\text{or } x \text{ } y)$$

I think that is the semantics we want. That is to say, if the policy doesn't apply, it doesn't enter into the equation. I also surmise to keep things easily consistent in inductive arguments about ANDs and ORs of sequences. The AND or OR of a zero length sequence of values can be anything constant we want, but the minimum element NOT-APPLICABLE would make the most sense, since $(\text{and } x \text{ N}) = (\text{and } x)$, from our assumption above, and, $(\text{and } x) = x$, which is still another wily assumption, but makes sense,

So therefore $(\text{and } N) = N$, but from above, $(\text{and } N) = (\text{and})$, Therefore, $(\text{and}) = N$

So we would have,

`<and></and> = NOT-APPLICABLE`

`<or></or> = NOT-APPLICABLE`

Also, to satisfy Hals "the customer's want it", I am almost on the side of allowing NOT in the language with the following semantics:

`p NOT p`

T F

F T

N N

E E

That is to say NOT of NOT-APPLICABLE is still NOT-APPLICABLE. Then NOT distributes through the AND and ORs (i.e. DeMorgan's Law) quite nicely.

$$(\text{NOT } (\text{AND } N \text{ } x)) = (\text{OR } (\text{NOT } N) (\text{NOT } x))$$

$$(\text{NOT } x) = (\text{OR } N (\text{NOT } x))$$

$$(\text{NOT } x) = (\text{NOT } x)$$

$$(\text{NOT } (\text{OR } N \text{ } x)) = (\text{AND } (\text{NOT } N) (\text{NOT } x))$$

$$(\text{NOT } x) = (\text{AND } N (\text{NOT } x))$$

$$(\text{NOT } x) = (\text{NOT } x)$$

However, differing from alternative #2 in the proposal below, I believe `<NOT></NOT>` shouldn't exist, and it should have one and only one constituent. And empty NOT is a syntax error, as well as having more than one, i.e. `<NOT> x y </NOT>` shouldn't type check either. (how do you say that in XML? `minoccurs=1, maxoccurs=1?`).

536 For completeness the truth tables in the 4-valued logic are below for "and", "or" and "not", (ed
537 note: truth tables left out. See original email)

538 Champion: Anne

539 Status: Open

540 **ISSUE:[PM-1-06: result of <N-OF n=0> combinator expression]**

541 We all agreed that <N-OF n=[something greater than 0]> was an error if there were not at least n
542 predicates to be evaluated. We also agreed that the semantics of <N-OF> were "at least n of".
543 We did not agree on what should be the result of <N-OF n=0>.

544 Potential Resolution:

545 <N-OF n=0> results in TRUE, regardless of the results of the predicates in the combinator
546 expression.

547 Champion: Anne

548 Status: Open

549 **ISSUE:[PM-1-07: How can the set of combinators be extended?]**

550 We agreed at the March, 2002 F2F that XACML would define the <AND>, <OR>, <NOT>, <N-
551 OF>, and <ORDERED-[AND|OR|NOT|N-OF]> combinators. How can a policy writer extend
552 this set to define a new combinator, such as BEST-MATCH-OR?

553 Potential Resolution:

554 The set of Combinators may be extended by specifying a name for the new Combinator, a URI
555 that is associated with the semantics of the new Combinator, and a type that specifies the way the
556 URI is to be interpreted. Not all XACML PDPs will be able to interpret all extensions, but any
557 PDP that can handle the specified type and can access the specified URI can handle the specified
558 extended Combinator.

559 An example of a possible extended Combinator is BEST-MATCH-OR. The type for such an
560 extended Combinator might be "JavaClass". The URI for each might point to a Java class that
561 takes a set of Predicates as input and implements the semantics of the combinator to return a
562 result of TRUE, FALSE, NOT-APPLICABLE, or ERROR.]

563 Champion: Anne

564 Status: Open

ISSUE:[PM-1-08: syntax for <applicablePolicyReference>]

If a predicate in XACML references an <xacml:applicablePolicy>, what should the syntax for this reference be?

Potential Resolution:

The syntax should include a URI for <xacml:applicablePolicy> and a URI for the Policy Authority trusted to issue and sign this <xacml:applicablePolicy>. The name attribute in the referenced <xacml:applicablePolicy> must match the URI in the <applicablePolicyReference>. A chain of <applicablePolicyReference> that contains a cycle has a result of ERROR.

Champion: Anne

Status: Open

Group 2: Applicable Policy

ISSUE:[PM-2-01: Referencing Multiple Policies]

According to the current schema an Applicable Policy seems to refer to a single Policy. The discussions in the last conference call seem to assume that an Applicable Policy can refer to several Policies (distributed scenario and multiple issuers [Anne]). Is there agreement on this point? If so, the schema should be modified accordingly.

Group 1 issues are captured within this

[Tim] The current schema allows one possible way of achieving this. Separate applicable policies from independent PAPs (Policy Administration Points) may be combined in a single "applicable policy" by a PRP. This approach does, however, make the original PAPs anonymous.

Potential Resolutions:

[Tim] An XACML "applicable policy" will not reference external "applicable policies". However, it may "incorporate" external "applicable policies". [PM-2-01] [PM-3-01] [PM-5-03]

[Tim] An XACML "applicable policy" shall be capable of referencing an external "applicable policy", providing explicit rules for combining such policies. [PM-2-01] [PM-3-01] [PM-5-03]

Champion: Anne

Status: Open

593 **ISSUE:[PM-2-02: Target Specification]**

594 According to the current schema each applicable policy can have multiple targets, each of which
595 is an action and a URI identifying a set of resources (possibly with a transfer function to support
596 wildcards). One may want to specify the target with reference to resource attributes (e.g., this
597 policy applies to all files older than two years). How can I specify this?

598 [Tim] A different transform algorithm is all that is required. In the example, the "classification"
599 is "older than two years", and the transform algorithm specifies how to deduce the age of a file.

600 Potential Resolutions:

601 Ernesto suggests that this issue only mention retrieval of distributed policies and should be
602 updated to reflect the recent discussion and Anne's proposal (See PM-1-01A) about policy
603 combination. Anne volunteers to extend its wording in order to include policy combination as
604 well.

605 Simon will present counter deductions to Anne 's proposal at the F2F

606 Champion: Simon G.

607 Status: Open

608 **ISSUE:[PM-2-03: Meaningful Actions]**

609 There are pairings <resource,actions> which are not meaningful (e.g., execute a PDF file)
610 [Simon G.]. Should we control resource/action bindings in the language or refer to an external
611 authority?

612 Potential Resolutions:

613 [Tim] The administrative model in Figure 9 deals with this question, placing it out of scope for
614 the schema. If we do need to tackle this, I suggest leaving it for a later version.

615 [Tim] The XACML syntax shall not address the question of which actions are valid for a
616 particular resource classification. This matter shall be left for implementations to solve in a non-
617 standard way. [PM-2-03]

618 Champion: Simon G.

619 Status: Open

620 **ISSUE:[PM-2-04: Indexing Policy]**

621 Also related to target are indexing issues and how to retrieve, given a request, the applicable
622 policy for it [Tim].

623 Potential Resolutions:

624 [Tim] Section 6.4 of version 0.8 of the language proposal is reserved for tackling this question in
625 the LDAP case. Do we need to tackle other cases?

626 [Tim] The XACML specification shall provide normative, but non-mandatory to implement, text
627 that profiles LDAP for distribution of XACML instances. [PM-2-04]

628 [Tim] The XACML specification shall provide normative, but non-mandatory to implement, text
629 that profiles "the Web" for distribution of XACML instances. [PM-2-04]

630 Champion: Tim

631 Status: Open

632 **ISSUE:[PM-2-05: Ensuring Completeness]**

633 The applicable policy is defined as the ``complete" set of policies that apply to a resource. How
634 do I ensure completeness (meaning no two targets should intersect?)

635 Potential Resolutions:

636 [Tim] This is a job for the PRP and should (I think) be out of the scope for our specification. The
637 PRP has to be configured with the names and locations of the PAPs whose policies it recognizes.

638 [Tim] The XACML syntax shall not address the question of ensuring that "applicable policy" is
639 complete. This matter shall be left for PRP implementations to solve in a non-standard way.
640 [PM-2-05]

641 Proposed Resolution:

642 1. If a Base Policy is included in the Access Request, then that Base Policy is the only one that
643 will be applied to the Access Request. Otherwise,

644 2. If a PDP has a single Base Policy, then the PDP's Base Policy specifies the complete
645 <applicablePolicy> that will be used by that PDP in evaluating an Access Request. This
646 <applicablePolicy> may actually be a tree of <applicablePolicy> statements, where additional
647 statements are logically incorporated by the use of <referencedPolicy> predicates.

648 In this case, there are no overlapping targets. If the PDP's Base Policy has an empty "target"
649 element, then all Access Requests are evaluated against the <policy>. If the Base Policy has a
650 non-empty "target" element, then any Access Request that does not match the "target" returns a
651 result of "NOT-APPLICABLE" (=SAML INDETERMINATE). If the Access Request matches
652 the "target", then the result of the Access Request is the result of evaluating the <policy>.

653 3. If a PDP has multiple Base Policies, then the PDP must specify and publish its algorithm for
654 deciding which Base Policies to evaluate, in which order, and how target overlaps are resolved.

Champion: Pierangela

Status: Ready to Close

ISSUE:[PM-2-06:Encapsulation of XACML policy (was Policy Security)]

Resolution 4: An XACML "applicable policy" will contain its own security features (e.g. signature), rather than relying on an encapsulating saml assertion.

Potential Resolutions:

[Anne] XACML will be specified in two separate layers.

1. The first layer is the <applicablePolicy> syntax, and will contain no security provisions such as authentication (signature), integrity protection, or encryption.

2. The second layer is a specification of how the first layer can be embedded in another mechanism for security protection. The XACML TC will define such a mechanism using an encapsulating SAML assertion. OASIS members are free to propose other mechanisms, such as encapsulating an <applicablePolicy> inside an X.509 Attribute Certificate.

Implementations may be compliant with the first layer only, with both the first layer and with the XACML TC-defined second layer, or with the first layer and another specified mechanism for the second layer. Implementations must state which level of compliance they support.

Champion: Tim

Status: Open

ISSUE:[PM-2-07: valueRef type]

Resolution 5: XACML valueRef elements shall be of type "saml:AttributeValueType".

Potential Resolutions:

???

Champion: Tim

Status: Open

Group 3: Policy Composition

Assuming an Applicable Policy can refer to several Policy elements, we need to answer the following questions:

ISSUE:[PM-3-01: Combining Policy Elements]

How are the Policy Element combined? For instance, we could support Boolean expressions of policies. E.g., if there are three policies by independent issuers, I can say ``P1 AND (P2 OR P3)? This could fit well in the multiple issuers scenario Anne was envisioning. Should this be part of the core of the extension (external URI [Michiharu])?

Potential Resolutions:

[Tim] We could add "policy" to the "sequence" in "rule". Then we would have to give policies unique identifiers, not just string names. Perhaps, we should add "applicable policy", instead of "policy".

[Tim] An XACML "applicable policy" will not reference external "applicable policies". However, it may "incorporate" external "applicable policies". [PM-2-01] [PM-3-01] [PM-5-03]

[Tim] An XACML "applicable policy" shall be capable of referencing an external "applicable policy", providing explicit rules for combining such policies. [PM-2-01] [PM-3-01] [PM-5-03]

Champion: Michiharu

Status: Open

ISSUE:[PM-3-02: Specifying Policy Outcome]

How the policy outcome should be specified. Possibilities are 2-valued (access decision is ``grant"/"deny") or 3-valued (policy outcome is ``grant"/"deny"/nothing). Note the ``nothing" means that no rule applies, to be solved according to default. (Related work on composition...?)

How does the PEP interpret the answer I don't know?

Potential Resolutions:

[Tim] Ultimately, the PEP has to know whether or not to grant access. So, someone has to decide, and (by definition) it is the PDP. So, the "don't care" response isn't helpful. However, saml should have an error code to indicate that the PDP is not the appropriate PDP to render a decision on a particular request.

[Tim] The XACML specification shall specify when a PDP should return saml:decision attributes with the values "permit" and "deny". If the PDP is unable to render a decision, then a saml status code shall be returned. No decision value shall be supplied in this case. [PM-3-02]

Champion: Simon

Status: Open

ISSUE:[PM-3-03: multiple Base Policies]

Can a PDP have more than one Base Policy?

Potential Resolutions:

Alternative 1:

A PDP MAY have multiple Base Policies, but such Base Policies SHOULD have non-overlapping <xacml:target> elements. The XACML specification does not specify the order in which multiple Base Policies are evaluated, or the result if two or more Base Policies have overlapping <xacml:target> elements.

A PDP that has multiple Base Policies MUST publish its algorithm for the order in which Base Policies are evaluated and the result where two or more Base Policies have overlapping <xacml:target> elements.

Alternative 2:

Base Policies have restricted <target> elements that are easily compared for overlap. In this alternative, the case where base policies overlap is an ERROR. Note that the 0.8 syntax favors this alternative and allows Alternative 3.

Alternative 3:

There is only one Base Policy. Either it has no <target>, and applies to all Resources or it has a <target> element that specifies the set of resources which this PDP is prepared to handle and returns NOT-APPLICABLE if a resource does match that target.

Champion: Anne (who supports Alternative 3)

Status: Open

ISSUE:[PM-3-03: default PDP result]

If no Base Policy applies to a given Access Request (i.e. all Base Policy evaluations return NOT-APPLICABLE), does the PDP return NOT-APPLICABLE (=SAML INDETERMINATE) to the PEP, or is the PDP configured with a default result to return (e.g. TRUE or FALSE)?

Potential Resolution:

If no Base Policy applies to a given Access Request, then the PDP returns NOT-APPLICABLE (=SAML INDETERMINATE) to the PEP.

Champion: Anne

Status: Open

Group 4: Syntax

ISSUE:[PM-4-01: Triplet Syntax (was Syntactic Sugar)]

The current schema assumes authorizations are specified as a pre-condition which is an expression made of predicates on SAML attributes (conditions on principal, resource and environment can be interspersed), let's call it Option ``pre-cond" [Carlisle, Tim, Anne, ...]. In the last conference call it was agreed to leave as an open issue whether to group conditions about principal, resource, and environment in three different elements, let's call it Option ``triplet" [Michiharu, Ernesto, Simon,]. The argument for Option ``pre-cond" is that there are predicates that involve both principal and resource attributes (e.g., an authorization that states that users can read the files they own). The counter-objection to this is that you can naturally include all predicates on resources in the resource condition element (which can also refer to principal attributes). The argument for the triplet is that it makes authorization specifications conceptually clearer and closer to current approaches.

[Tim] In the 0.8 schema, valueRef has an attribute to indicate the entity to which it applies (principal, resource, etc.). It only has to be consulted if the attribute type identifier is ambiguous.

Potential Resolutions:

[Tim] The XACML syntax will differentiate between model entities (principal, resource, etc.) in its attribute elements, rather than in its rule elements. [PM-4-01]

Champion: Pierangela

Status: Open

ISSUE:[PM-4-02: Policy names as URIs]

Policy names are strings. Should we make them URIs?

Potential Resolutions:

Proposed Resolution:

Policy names should be URIs.

Champion: Tim

Status: Ready to Close

ISSUE:[PM-4-03: Required type in policy]

The "rec:patient/patientName" element is a complex type. So, how should we indicate the required type in the policy?

[From PM-4-09] This only allows for simple types. Do we need to support values of complex type?

Potential Resolutions:

???

Champion: Tim

Status: Open

ISSUE:[PM-4-04:syntax extension]

Issue: should this element be an extension point to which other policy syntaxes can be added?

Potential Resolutions:

Propose Resolution:

Close this issue. It is incompletely specified: which element? Extension issues are in a separate section.

Champion: Tim

Status: Ready to Close

ISSUE:[PM-4-05:Policy Name a URI]

Issue: should we make policy name a URI?

Potential Resolutions:

See PM-4-02

Champion: Tim

Status: Closed as Duplicate

ISSUE:[PM-4-06:Comment element]

Issue: Should we include a "comment" element?

Potential Resolutions:

Proposed Resolution:

We should include a "comment" element.

Champion: Tim

Status: Ready to Close

ISSUE:[PM-4-07:policy element in a rule]

Issue: Should we allow a policy element in a rule? Then the same schema could express the policy for combining policies. If so, should it be policy or applicable policy?

Potential Resolutions:

See PM-3-01

Champion: Tim

Status: Closed as Duplicate

ISSUE:[PM-4-08:XML elements include xsi:type]

Issue: Should we require XML elements compared in this way to include an xsi:type attribute?

Potential Resolutions:

???

Champion: Tim

Status: Open

ISSUE:[PM-4-09:complex types]

Issue: This only allows for simple types. Do we need to support values of complex type?

Potential Resolutions:

See PM-4-03

Champion: Tim

Status: Closed as Duplicate

ISSUE:[PM-4-10:preserve PAP identity]

Issue: Should the identities and/or signatures of the PAPs be preserved in the composed policy?

Colors: Gray Blue Yellow

821 Potential Resolutions:

822 ???

823 Champion: Tim

824 Status: Open

825

826 **Group 5: SAML Related**

827 In the current schema attributes on resources and principals, which can be used in the Target (for
828 resources) and in predicates, are retrieved using URIs pointing to SAML dataflow.

829 [ISSUE:\[PM-5-01: Non-SAML Input\]](#)

830 Can this mechanism be extended to point to non-SAML authorities as required in the Java
831 environment [Sehkar]?

832 At a minimum, extending SAML expressions but broader to other authorities.

833 Potential Resolutions:

834 [Tim] The XACML specification shall be closely coupled to saml entities. However, the use of
835 saml namespace identifiers is not intended to imply that all attributes must be retrieved from
836 saml messages and assertions. [PM-5-01]

837 Champion: Sehkar

838 Status: Open

839 [ISSUE:\[PM-5-02: Wildcards on Resource Hierarchies\]](#)

840 How do we express wildcards on the resource hierarchies [Simon G.]?

841 The current schema includes ResourceToClassificationTransform to this purpose. Is this
842 sufficient?

843 Potential Resolutions:

844 [Tim] We should register an OASIS identifier for the use of regular expressions in this context.

845 [Tim] The XACML syntax shall use registered URIs to identify algorithms for processing
846 resource classification wildcards. [PM-5-02]

847 Proposed Resolution:

848 Use "ResourceToClassificationTransform". Register a URI with OASIS for the use of regular
849 expressions in this context. Other transform algorithms may be specified by the use of other
850 URIs to be registered with OASIS.

851 Champion: Simon G.

852 Status: Ready to Close

853 **ISSUE:[PM-5-03: Roles and Group Hierarchies]**

854 Are roles and groups hierarchies available via SAML [Simon G.]? Hierarchies could be needed,
855 in case of support of negative rules, for resolving conflicts based on more-specific-takes-
856 precedence. Note: policy resolution conflicts fit well when the principal is a group, they may be
857 difficult to apply in case of principal's expressions.

858 Potential Resolutions:

859 [Tim] An XACML "applicable policy" will not reference external "applicable policies".
860 However, it may "incorporate" external "applicable policies". [PM-2-01] [PM-3-01] [PM-5-03]

861 [Tim] An XACML "applicable policy" shall be capable of referencing an external "applicable
862 policy", providing explicit rules for combining such policies. [PM-2-01] [PM-3-01] [PM-5-03]

863 Proposed Resolution:

864 XACML will not support role and group hierarchies in the policy language. Attribute authorities
865 may support role and group hierarchies.

866 Champion: Simon G.

867 Status: Ready to Close

868 **ISSUE:[PM-5-04: SAML Assertions URI]**

869 From the schema it seems that expressions are predicates whose arguments are always URI or
870 value. Are SAML assertions always URI?

871 Potential Resolutions:

872 [Tim] Attributes in saml assertions are identified by a namespace, which is a URI, and a name,
873 which is a string.

874 Simon suggests that the current solution is in general enough, as the URI+XPath combination
875 specifies a schema (via the URI) and allows to retrieve a value (via the XPath). XPaths guarantee
876 that values are uniquely identified. This technique smoothly applies not only to SAML but also
877 to other formats like LDAP.

878 Hal observes that this is not always the case, as there may be attribute namespaces which are not
879 URI.

880 Anne remarks that besides a pointer to the schema, a pointer to an instance is also needed. Simon
881 agrees to provide a full explanation of this scenario at the F2F.

882 This issue conflates two separate issues:

883 1. Are SAML assertions always URI?

884 2. references to attributes in XACML predicates. (See new issue PM-1-04)

885 Proposed Resolution:

886 Attributes in SAML assertions are identified by a namespace, which is a URI, and a name, which
887 is a string.

888 Champion: Simon

889 Status: Ready to Close

890 [ISSUE:\[PM-5-05: XPath\]](#)

891 Use of Xpath for identifying SAML constructs and the use of Xpath operators

892

893 Potential Resolutions:

894 Simon clarifies that the position he will take is that while the use of Xpaths to extract nodeset is
895 just fine, they do not make good values in expression. The solution in the current schema is
896 cleaner.

897 Anne offers to look into the issue to provide an alternative point of view.

898

899 Champion: Simon

900 Status: Open

901 [ISSUE:\[PM-5-06: Multiple actions in single request\]](#)

902 In the SAML issues document, [http://www.oasis-open.org/committees/security/docs/draft-sstc-](http://www.oasis-open.org/committees/security/docs/draft-sstc-core-discussion-01.doc)
903 [core-discussion-01.doc](http://www.oasis-open.org/committees/security/docs/draft-sstc-core-discussion-01.doc)

904 ... Issue 5.1.15.2 seeks guidance on whether multiple "actions" can be specified in a single
905 decision request.

906 Potential Resolutions:

907 [Tim] I feel that XACML should answer this question and send its conclusion in a liaison to
908 SAML. My feeling is that the answer is "No". If "applicable policy" is to be identified with the
909 resource/action pair, then multiple "applicable policies" are involved when multiple actions are
910 involved. Much "cleaner" for there to be a single "applicable policy" for each decision request.
911 And, therefore, a single action per decision request. It is no great hardship to submit multiple
912 decision requests, in the event that you need a decision for each of several actions.

913 [Hal] Personally I am in favor of limiting this, but I will state the counter argument for the
914 record. If the possible Actions correspond to what can be in the request, then this works fine. The
915 only reason for multiple actions would be some sort of policy provisioning requirement.
916 However, if the Actions are more like privileges or permission bits, and do not match allowable
917 requests one for one, then some requests may require the AND or OR of several actions. I
918 believe this is the motive behind suggesting multiple actions.

919 I don't see any rush on this as we are not close to proposing changes to the decision protocol yet.

920 Champion: Tim

921 Status: Open

922 [ISSUE:\[PM-5-07: Delegation\]](#)

923 [Polar] Has anybody thought about how delegation can be reasoned about in XACML? It
924 appears that SAML only asserts a flat list of attributes with a single principal, or am I off base
925 here? Can I support policies on such operations as:

926 Paul for Peter says debit Peter's account?

927 Which mean that Paul (or some other party trusted to do so) has issued Paul the authorization to
928 act on behalf of Peter, in this case to access Peter's account. Or such things, like WebServer
929 quoting JohnDoe says lookup in customer database. Where the WebServer may be trusted to
930 authenticate JohnDoe, but no such proof is necessary other than the WebServer merely claiming
931 to be acting on JohnDoe's behalf?

932 Potential Resolutions:

933 [Hal] With regards to SAML, the Access Decision Request was deliberately kept simple with the
934 idea that XACML would give us the tools to do the job properly. I have proposed (see my use
935 cases) that XACML not only be able to express policies, but the method of expressing policy
936 inputs be rolled back into the SAML Access Decision Request (and Assertion).

937 In my opinion, XACML policies should be able to contain predicates about zero or more of the
938 following subjects:

939 Requestor Subject

940 Recipient Subject (can be different from requestor)

941 Intermediary Subject (can be more than one for a given request)

942 I propose a single construct for Subjects and their attributes and some kind of modifier indicating
943 the type (refrain from using "role" here) of subject.

944 [Tim] Delegation could be expressed in attribute assertions. The very issuance of an attribute
945 assertion is a form of delegation. So, XACML should not have to concern itself with the process
946 by which an entity obtained an attribute.

947 Champion: Polar/Hal

948 Status: Open

949 **ISSUE:[PM-5-08: saml:Action is a "string"]**

950 These are some of the potential SAML issues. Most of them were found when attempting to
951 write J2SE policy files in XACML syntax. Further discussion is needed on these issues.

952 saml:Action is currently specified as a "string". Making Action an abstract type would allow it
953 to be extended. This would allow the content model to be defined by a schema external to the
954 SAML spec.

955 Thus what constitutes an action could be determined by the J2SE schema.

956 Potential Resolutions:

957 [Toshi] In SAML, saml:Action is used only in saml:Actions and saml:Actions have Namespace
958 as an attribute. So it is possible to write action(s) such as:

959 <saml:Actions Namespace="urn:J2SEPermission:java.io.FilePermission">
960 <saml:Action>write</saml:Action>
961 </saml:Actions>

962 or

963 <saml:Actions Namespace="urn:J2SEPermission">
964 <saml:Action>java.io.FilePermission:write</saml:Action>
965 </saml:Actions>

966 But it will be useful if we can write something like:

967 <saml:Action>
968 <J2SEPermission class="java.io.FilePermission">write</J2SEPermission>

969 </saml:Action>

970 Champion: Sekhar

971 Status: Open

972 **ISSUE:[PM-5-09: saml:AuthorizationQuery requires actions]**

973 If actions are optional for XACML, then why should <saml:Actions> be required in
974 <saml:AuthorizationQuery> ? Both the wording in the SAML assertions draft as well as the
975 SAML schema places such a requirement. saml:Actions should be optional in the
976 AuthorizationQuery to accommodate queries without actions. At least for now, I don't anticipate
977 this as an issue for J2SE.

978 Potential Resolutions:

979 [Toshi] In the latest SAML spec (core-25), AuthorizationDecisionQuery element has Resource
980 attribute and Actions element and both of them are "required". Does this cause many problems?

981 (Resource attribute is "optional" for AuthorizationDecisionStatement element.)

982 As for J2SE case, I think there is an issue in terminology.

983 Champion: Sekhar

984 Status: Open

985 **ISSUE:[PM-5-10: single subject in AuthorizationQuery]**

986 [editor note: Is this issue covered somewhere else?]

987 saml:AuthorizationQuery currently only contains a single Subject. While a saml:Subject can
988 support multiple NameIdentifier or SubjectConfirmation or AssertionSpecifier elements, it is
989 required that they all belong to the same principal. So a single subject cannot be used for
990 unrelated principals. In J2SE, there is a need to base access control on multiple principals which
991 are not related and this therefore points to a need for more than one Subject in the
992 saml:AuthorizationQuery

993 Potential Resolutions:

994 The way out of this appears to be extend SubjectQueryAbstractType.

995 Champion: Hal

996 Status: Open

997 [ISSUE:\[PM-5-11:XACML container in SAML\]](#)

998 Issue: should we use a SAML assertion as a container for an XACML applicable policy?

999 Potential Resolutions:

1000 ???

1001 Champion: Tim

1002 Status: Open

1003 [ISSUE:\[PM-5-12:derive attribute from saml:AttributeValueType\]](#)

1004 Issue: Should we derive the attribute from saml:AttributeValueType? This seems to make sense,
1005 but the resulting attribute will have to become an element, with start and stop tags, making it
1006 larger and less readable.

1007 Potential Resolutions:

1008 ???

1009 Champion: Tim

1010 Status: Open

1011 [ISSUE:\[PM-5-13: Base Policy supplied as part of AuthorizationDecisionQuery\]](#)

1012 Some PEPs have knowledge of the policy associated with a resource (example: a typical
1013 FileSystem knows the ACLs associated with a file or directory). To support this case, can a Base
1014 Policy or <referencedPolicy> be supplied as part of the SAML AuthorizationDecisionQuery?

1015 Possible Resolutions:

1016 Default policy:

1017 A Base Policy or <referencedPolicy> for evaluating a particular Access Request may be
1018 specified as part of the Access Request. If a PDP has no Base Policy(s), then the result of
1019 evaluating an Access Request that does not specify a Base Policy to use is NOT-APPLICABLE
1020 (=SAML INDETERMINATE).

1021 Champion: Anne

1022 Status: Open

1023 **Group 6: Predicate Cononicalization**

1024 **ISSUE:[PM-6-01: SAML Assertions URI]**

1025 Values used in predicates can refer to various standard formats (e.g, X.509 [Anne]) that could
1026 make the predicates evaluation difficult. For instance, if a principal's name is expressed in X.500
1027 syntax you cannot compare it against a simple string. How do we make the representations
1028 canonical?

1029 Potential Resolutions:

1030 [Tim] Policy environments have to use consistent type definitions for the attributes they use.

1031 Champion: Anne

1032 Status: Open

1033 **Group 7: Extensibility**

1034 **ISSUE:[PM-7-01: XACML extensions]**

1035 XACML Extension Model that defines what portion of the XACML specification is a core and
1036 to what extent the XACML specification can be extended. Based on this proposal, XACML
1037 policy administrators can represent much broader access control policies by extending the core
1038 portion of the XACML specification.

1039 This extension model is designed to support an XACML extensibility property stated in the
1040 XACML charter. This proposal is based on the current language proposal document but includes
1041 several modifications.

1042 Potential Resolutions:

1043 See <http://lists.oasis-open.org/archives/xacml/200112/msg00076.html>

1044 Champion: Michiharu

1045 Status: Open

Miscellaneous Issues

Group 1: Glossary

ISSUE:[MI-1-01: Consistency]

Pierangela mentioned something discussed in PM group that may not coincide with glossary concerning pre and post conditions.

Potential Resolutions:

???

Champion: Pierangela

Status: Open

Group 2: Conformance

ISSUE:[MI-2-01: Successfully Using]

XACML definition of OASIS requirement to successfully use the specification

Potential Resolutions:

"Successfully Using the XACML Specification"

XACML is an XML schema for representing authorization and entitlement policies. However, it is important to note that a compliant Policy Decision Point (PDP) may choose an entirely different representation for its internal evaluation and decision-making processes. That is, it is entirely permissible for XACML to be regarded simply as a policy interchange format, with any given implementation translating the XACML policy to its own local/native/proprietary/alternate policy language sometime prior to evaluation.

A set of test cases (each test case consisting of a specific XACML policy instance, along with all relevant inputs to the policy decision and the corresponding PDP output decision) will be devised and included on the XACML Web site.

In order to be "successfully using the XACML specification", an implementation MUST, for each test case, have a "policy evaluation component" that can consume the policy instance and the inputs and produce the specified output.

Furthermore, the implementation MUST have a "policy creation component" that allows it to generate schema-valid XACML policy instances that can be consumed/processed by other PDPs.

Note that, aside from the XACML policy instance itself, all PDP inputs and outputs MUST be

1075 SAML-compliant (i.e., conform with the assertions and protocol messages defined in the SS-TC
1076 SAML specification), although other syntaxes/formats for the PDP input and output MAY be
1077 supported in addition to this.

1078 Champion: Carlisle

1079 Status: Closed

1080 **Group 3: Patents, IP**

1081 **ISSUE:**[\[MI-3-01: XrML\]](#)

1082 [Ernesto] As I recollect, OASIS requested us to evaluate whether any XACML specification
1083 might fall in the scope of patents held by others. I quote from a Dec 13th addition to
1084 announcements regarding Xerox's XrML:

1085 (<http://xml.coverpages.org/xrml.html>) :

1086 "ContentGuard's strategy appears to be to make money by licensing the technology -- whatever
1087 some outside body defines it to be. It can do this because its patents cover the idea of a rights
1088 language in general, no matter what the specifics of the language are".

1089 I know XrML has already been mentioned in our discussions from the technical point of view,
1090 but the wording of this announcements makes me suspect that we should explore the matter
1091 further from the patents' point of view.

1092 Potential Resolutions:

1093 Oasis has a specific IPR policy and ContentGuard needs to make Oasis aware of any IP as it
1094 relates to XACML or other technical committees in accordance with that policy.

1095 [Hal] Paragraph (C) of OASIS.IPR.3.2. makes the following points:

1096 If OASIS knows about something they "shall attempt to obtain from the claimant of such rights a
1097 written assurance ..."

1098 However, "results of this procedure shall not affect advancement of a specification..."

1099 Except that "The results will, however, be recorded..." and "...may also direct that a summary of
1100 the results be included in any OASIS document published containing the specification." It also
1101 says elsewhere that they will not go out of their way to find IPR that has not been drawn to their
1102 attention.

1103 Champion: Ernesto

1104 Status: Open

Group 4: Other Standards

ISSUE:[MI-4-01: RuleML]

Should XACML look at RuleML?

[Edwin] XACML folks, Since XACML is about defining "rules" for Authorization -- would it make sense to leverage work done by the RuleML folks?

RuleML folks, You may want to checkout XACML as an application of RuleML. Here is a standard that will be real within the next year!]

Potential Resolutions:

The issue is a generic suggestion about XACML to be a possible application of a general setting for rule representation, RuleML.

Anne proposes that at the F2F every suggestion of taking into account related languages should be mandatory accompanied by a presentation

After a brief discussion on RuleML, the issue is voted closed. It should be deleted from the next version of the issues document

Champion: Edwin

Status: Closed

ISSUE:[MI-4-02: RAD]

Should XACML look at RAD?

[Polar] In response to some query about the expressiveness of evaluation of policies from different places, I would like to point the group to the CORBA Resource Access Decision specification (RAD).

<http://www.omg.org/cgi-bin/doc?formal/01-04-11.pdf>

and we may want to include it the document repository. It has in it an Access Decision model in which not only policies are located, but also, a policy evaluation combinator is located for a

particular resource. Note, there is no language component to this specification.

However, it does present a model by which policy can be distributed and evaluated. A combinator, which has an interface operation of "evaluate_policies" takes the list of located policies for the resource, the attribute list of the subject, and the operation (i.e. Action) on the resource) and evaluates the decision.

1134 That way, depending the semantics of the combinator you choose for the resource, your
1135 combinator may choose to ignore, or evaluate only some policies based on the evaluations of
1136 other policies.

1137 Potential Resolutions:

1138 Polar will bring that one to the discussion, with special reference to policy combination.

1139 Champion: Polar

1140 Status: Open

1141 [ISSUE:\[MI-4-03: DSML\]](#)

1142 Transformations from XACML to DSML

1143 [Gil] Since the last time we talked I had the chance to play with DSML a little. It seems to me
1144 that it is theoretically possible to transform an XACML policy document into a DSML document
1145 and import that document into LDAP. The DSML document could contain elements that
1146 described the (LDAP) schema necessary to store the authorization policy entries in case the
1147 target LDAP

1148 didn't already have this schema. It is also possible to export some LDAP entries into a DSML
1149 document and transform that DSML document in XACML.

1150 What I don't know (having nothing more than a cursory understanding of XSL/XSLT) is how
1151 difficult such transformations would be and if there are any "gotchas" that would keep this from
1152 really working.

1153 Potential Resolutions:

1154 [Gil] What I think the XACML spec should do is:

1155 1.) Describe the LDAP schema necessary to store authorization policies. This should be done in
1156 "LDAP fashion" with dn's, classnames, etc.

1157 2.) (if possible) Provide the XSLT necessary to transform XACML to DSML and vice versa.

1158 That way people who don't want to be bothered with DSML can work out their own way to store
1159 and retrieve XACML data to and from the defined schema.

1160 Champion: Gil

1161 Status: Open

1162 **ISSUE:[MI-4-04: Java Security Model]**

1163 Hal says he is not clear about whether XACML should be able to represent the Java security
1164 model. Gil comments that XACML would be limited if it cannot express it. Hal notes that what
1165 XACML should be able to represent are the same requirements that Java security model
1166 represents, but not necessarily in the same way (i.e., representing the same authorizations).

1167 Potential Resolutions:

1168 ???

1169 Champion: Sekhar

1170 Status: Open

1171 **Document History**

- 1172 • 7 Jan 2002 First Version Published
- 1173 • 21 Jan 2002 Major edits and additions. Every open item updated.
- 1174 • 18 Feb 2002 Edits based on F2F and Anne's edits