Outline

• Access control in the Grid
• XACML overview
• Use model
• Policy language
• Other features
• Future work
• More information
Access Control in the Grid

Virtual Org P:
Multidisciplinary design using programs and data at multiple locations

“Participants in P can run program A”
“Participants in P can read data D”
“Participants in P can run program B”

Virtual Org Q:
Ray tracing using cycles provided by cycle sharing consortium

“Participants in Q can use cycles if idle and budget not exceeded”

XACML Overview

- eXtensible Access Control Markup Language
  General-purpose access control policy and query languages.
- Version 1.0 OASIS Standard, February 2003
- Version 2.0 on standards track now
- Publicly available (C++, C#) and open source (Java) implementations
XACML Overview

• Designed to work in either a centralized or distributed, decentralized environment.

• Ties into legacy systems easily
  - No requirements on what supplies the attribute information
  - No requirements on actual query language
  - No requirements on transport, storage, etc.

• Extensible: new attribute types, new functions
Example

A policy in plain English:

“Only clients
  • Who are employed by DOE, AND
  • Who are part of the “FusionGrid” Virtual Organization, AND
  • Who are authenticated with an X509 public key certificate
are allowed access to Grid resources.”
Two-part example

1) Access decision request

2) Policy
Part 1: Access decision request

• A request to the PDP:
  
  Is this access permitted?

• Describes the access
Access Decision Request

<Request>
  <Subject>
    ... Attributes of the subject doing the access ...
  </Subject>
  <Resource>
    ... Attributes of the resource being accessed ...
  </Resource>
  <Action>
    ... Attributes of the action to be done on the resource ...
  </Action>
  <Environment>
    ... Attributes of the access environment ...
  </Environment>
</Request>
A Request Attribute

Attribute Identity: “employer”
Type: URI
Value: “urn:us:gov:DOE”
Part 2: Policy

1) Access decision request

2) Policy

what an acceptable access description looks like
Progressive example

1. Referring to an attribute in the request
2. Placing a constraint on an attribute
3. Combining constraints
4. Specifying a rule
5. Specifying a policy
6. Specifying a policy set
Referring to an attribute

```xml
<SubjectAttributeDesignator
  AttributeId="employer"
  DataType="anyURI" />
```

Alternative:

```xml
<AttributeSelector
  RequestContextPath="/employer/text()"
  DataType="anyURI" />
```

XPath expression
Constraining an attribute

<Apply FunctionId="anyURI-is-in">
  <AttributeValue
    DataType="anyURI">
    urn:us:gov:doe
  </AttributeValue>
  <SubjectAttributeDesignator
    AttributeId="employer"
    DataType="anyURI" />
</Apply>
Combining constraints

<Condition>

  <Apply FunctionId="and">
    <Apply>“must be a DOE employee”</Apply>
    <Apply>“must be member of FusionGrid”</Apply>
    <Apply>“must authenticate with X509 cert”</Apply>
  </Apply>

</Condition>
<Rule
  RuleId="Rule1"
  Effect ="Permit">
  <Target ... />
  <Condition .... />
</Rule>

If <Target> AND <Condition> are TRUE, returns Effect

If <Target> OR <Condition> is FALSE, returns “Not Applicable”

If error, returns “Indeterminate”
Combining Algorithm

Rule 1: Effect = "Permit"
On evaluation: returns "Not Applicable"

Rule 2: Effect = "Deny"
On evaluation: returns "Deny"

Rule 3: Effect = "Permit"
On evaluation: returns "Permit"

Rule 4: Effect = "Permit"
On evaluation: returns "Indeterminate"

Combining Algorithm → Result
Policy: combination of <Rule>s

Deny-overrides: return "Permit" only if <Target> is TRUE AND every <Rule> returns "Permit".

Obligations: optional attributes returned to the PEP.
PolicySet: combination of <Policy>s and other <PolicySet>s
Further information

• XACML is in the Globus ToolKit:
  3.9.3 Java WS Core only distribution

• “A Brief Introduction to XACML”
  http://www.oasis-open.org/committees/download.php/2713/Brief_Introduction_to_XACML.html

• OASIS Access Control (XACML) Technical Committee:
  all specifications and other documents
  http://www.oasis-open.org/committees/xacml

• Sun's XACML Open Source Implementation
  http://sunxacml.sourceforge.net

Anne Anderson <Anne.Anderson@sun.com>
Backup slides
Access Policy Enforcement

Applications → PEP (Policy Enforcement Point) → PDP (Policy Decision Point)

PEP:
- access interception
- decision enforcement
- obligation fulfillment
- some attribute collection

Data/resources → attributes → policies
## Attributes

<table>
<thead>
<tr>
<th>Subject + Category</th>
<th>Attribute Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>user, intermediary, recipient, codebase, requesting machine, ...</td>
<td>Subject's identity, role, clearance level, <code>&lt;wss:SecurityToken&gt;</code>, account id, IP address, ...</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Resource</th>
<th>Attribute Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>{+ optional XML ResourceContent}</code></td>
<td>Resource's identity, classification, location, size, value, ...</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Action</th>
<th>Attribute Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Action identity: read, write, execute, modify, open, move, ...; Action purpose, ...</td>
<td>time of day, date, vocabulary id, contract id, ...</td>
</tr>
</tbody>
</table>
Target

Optional way to pull out key “necessary” predicates (could do everything in `<Condition>`). Useful for indexing policies.

```xml
<Target>
  <Subjects><Subject><SubjectMatch
      MatchId="anyURI-equal"  DataType="anyURI">
    <AttributeValue  DataType="anyURI">
      urn:us:gov:doe
    </AttributeValue>
  </SubjectMatch></Subject>
</Target>

<Condition>... remaining two predicates ... </Condition>
```
PolicySet: combination of <Policy>s and other <PolicySet>s

```xml
<PolicySet
   PolicySetId="PolicySet1"
   PolicyCombiningAlgId="deny-overrides">
   <Target .... />
   <Policy1 ... />
   <Policy2 ... />
   <PolicySet2 ... />
</PolicySet>
```

Deny-overrides: return "Permit" only if <Target> is TRUE AND every <Policy> and <PolicySet> return "Permit".
Some other features

- Distributed policies: inclusion by reference
- Variable definitions and references (re-use constraints, etc.)
- XPath references to attributes from XML documents
XACML Profiles

- Hierarchical Resources
- Multiple Resources
- Role Based Access Control (RBAC)
- Privacy
- Security Assertion Markup Language (SAML)
- Digital Signature (DSig)
Future work

• Policy tools
  - Composition, editing
  - Analysis
  - Management

• Delegation chains