eXtensible Access Control Markup Language (XACML) II

Nicola Zannone
XACML Recap

- OASIS standard
- Federate administration of policies about the same resource
- XACML Core Specification
  - attribute-based access control policy language
    - implemented in XML
    - default attributes and functions
  - language to express access requests and responses
  - policy evaluation process
  - high-level architecture (PEP, PDP, PIP, PAP, etc.)
XACML Architecture
Access Request & Policy Applicability

- Access Requests specify a context
  - Attributes of subject, object, action, environment
- A Rule/Policy/PolicySet has a target
  - Attributes of subject, object, action, environment
- Rule/Policy/PolicySet is applicable if the attributes in the access request “match” the attributes in the target

**REMARK:** Attributes in the target have an element *MustBePresent*, which determines the applicability of the Rule/Policy/PolicySet in case of missing attributes:

- If *MustBePresent* is “False” (default value), then a missing attribute results in “NotApplicable”
- If *MustBePresent* is “True”, then a missing attribute results in “Indeterminate”
Access Decision

- Permit
  - requested access is permitted
- Deny
  - requested access is denied
- Indeterminate
  - PDP is unable to evaluate the request
- NotApplicable
  - PDP does not have any policy that applies to the request
Extended *Indeterminate* Set

- Record potential effect value when errors occur
  - Indeterminate\{P\} (I\{P\})
  - Indeterminate\{D\} (I\{D\})
  - Indeterminate\{PD\} (I\{PD\})
- Used by some combining algorithms
Decision Set Projection

- $D_4 = \{ P, D, NA, I \}$
- $D_6 = \{ P, D, NA, I\{P\}, I\{D\}, I\{PD\}\}$

$D_6 \rightarrow D_4$

$I\{P\} \rightarrow I$

$I\{D\} \rightarrow I$

$I\{PD\} \rightarrow I$

$D_4 \rightarrow D_6$

$I \rightarrow I\{PD\}$
Combining Algorithms

- Deny-overrides
- Ordered-deny-overrides
- Permit-overrides
- Ordered-permit-overrides
- Permit-unless-deny
- Deny-unless-permit
- First-applicable
- Only-one-applicable
Obligations

- In XACML post obligations
- A policy or policy set may contain one or more obligations
- Obligations passed up to the next level of evaluation
- No obligations are passed up
  - if the policy or policy set is not evaluated
  - if result is “Indeterminate” or “NotApplicable”
  - if effect does not match FulfillOn attribute
Response

- Access Decision
- Obligations
Given the XACML policy and access request below, determine the access response.

**Hint:** If an attribute is missing (i.e., it is not provided in the request), then MustBePresent governs the applicability of the Rule/Policy/PolicySet.

- If MustBePresent is “False” (default value), then a missing attribute results in an empty bag. (i.e., the Rule/Policy/PolicysSet is “Not Applicable”)

- If MustBePresent is “True”, then a missing attribute results in an error. Some combining algorithms are defined in terms of an extended set of “Indeterminate” values (“Indeterminate(P)”, “Indeterminate(D)”, “Indeterminate(PD)”). The extended set associated with the “Indeterminate” contains the potential effect values which could have occurred if there would not have been an error causing the “Indeterminate”.
Solution

Access Request
Alice
clerk
accountant
FinancialRecord
read

PS
deny-overrides
R: FinancialRecord
O5
O6

P1
deny-overrides
O1
O2

R1
Deny
S1:manager
AT > 15:00 (M)
S2:clerk
AT > 15:00

R2
Permit
S:accountant
A:read

R3
Permit
S:accountant
AT > 15:00 (M)

R4
Permit
S:clerk
AT > 15:00 (M)

R5
Deny
S:accountant
A:read
Solution

Access Request
Alice
clerk
accountant
FinancialRecord
read

Applicable

P1
deny-overrides
O1
O2

P2
first-applicable
O3
O4

R1
Deny
S1:manager
AT > 15:00 (M)
S2:clerk
AT > 15:00

R2
Permit
S:accountant
A:read

R3
Permit
S:accountant
AT > 15:00 (M)

R4
Permit
S:clerk
AT > 15:00 (M)

R5
Deny
S:accountant
A:read

P1 deny-overrides

P2 first-applicable

PS
deny-overrides
R: FinancialRecord
O5
O6
Solution

Access Request

Alice
clerk
accountant
FinancialRecord
read

Applicable

PS
deny-overrides
R: FinancialRecord
O5
O6

R1
Deny
S1: manager
AT > 15:00 (M)
S2: clerk
AT > 15:00

R2
Permit
S: accountant
A: read

R3
Permit
S: accountant
AT > 15:00 (M)

R4
Permit
S: clerk
AT > 15:00 (M)

R5
Deny
S: accountant
A: read
Solution

Access Request
Alice clerk accountant FinancialRecord read

Applicable
PS deny-overrides R: FinancialRecord O5 O6

Applicable
P1 deny-overrides O1 O2

Applicable
P2 first-applicable O3 O4

R1 Deny S1:manager AT > 15:00 (M) S2:clerk AT > 15:00
R2 Permit S:accountant A:read
R3 Permit S:accountant AT > 15:00 (M)
R4 Permit S:clerk AT > 15:00 (M)
R5 Deny S:accountant A:read

P1 P2

N/A
### Solution

**Access Request**
- Alice
- clerk
- accountant
- FinancialRecord
- read

**Applicable**
- **P1**
  - deny-overrides
  - O1
  - O2

- **P2**
  - first-applicable
  - O3
  - O4

**N/A**

**R1**
- Deny
  - S1: manager
    - AT > 15:00 (M)
  - S2: clerk
    - AT > 15:00

**R2**
- Permit
  - S: accountant
    - A: read
    - AT > 15:00 (M)

**R3**
- Permit
  - S: accountant
    - AT > 15:00 (M)

**R4**
- Permit
  - S: clerk
    - AT > 15:00 (M)

**R5**
- Deny
  - S: accountant
    - A: read
Solution

**Access Request**

Alice
clerk
accountant
FinancialRecord
read

**Applicable**

**R1**

Deny
S1:manager
AT > 15:00 (M)
S2:clerk
AT > 15:00

**R2**

Permit
S:accountant
A:read

**R3**

Permit
S:accountant
AT > 15:00 (M)

**R4**

Permit
S:clerk
AT > 15:00 (M)

**R5**

Deny
S:accountant
A:read

PS
deny-overrides
R: FinancialRecord
O5
O6

P1
deny-overrides
O1
O2

P2
first-applicable
O3
O4

N/A

Permit

I{P}
Solution

Access Request
Alice
clerk
accountant
FinancialRecord
read

Applicable
PS
deny-overrides
R: FinancialRecord
O5
O6

Permit {O1}
P1
deny-overrides
O1
O2

N/A

Permit
R1
Deny
S1: manager
AT > 15:00 (M)
S2: clerk
AT > 15:00

Permit
R2
Permit
S: accountant
AT > 15:00 (M)

Permit
R3
Permit
S: accountant
AT > 15:00 (M)

I{P}

Permit
R4
Permit
S: clerk
AT > 15:00 (M)

Permit
R5
Deny
S: accountant
A: read

P1
deny-overrides
O1
O2

P2
first-applicable
O3
O4

PS
deny-overrides
R: FinancialRecord
O5
O6

N/A

R1
Deny
S1: manager
AT > 15:00 (M)
S2: clerk
AT > 15:00

R2
Permit
S: accountant
AT > 15:00 (M)
Solution

Access Request

Alice
clerk
accountant
FinancialRecord
read

Applicable

PS
deny-overrides
R: FinancialRecord
O5
O6

Permit {O1}

P1
deny-overrides
O1
O2

N/A

R1
Deny
S1: manager
AT > 15:00 (M)
S2: clerk
AT > 15:00

Permit

R2
Permit
S: accountant
A: read

I{P}

R3
Permit
S: accountant
AT > 15:00 (M)

Applicable

P2
first-applicable
O3
O4

R4
Permit
S: clerk
AT > 15:00 (M)

R5
Deny
S: accountant
A: read
Solution

Access Request
Alice
clerk
accountant
FinancialRecord
read

Applicable

PS
deny-overrides
R: FinancialRecord
O1
O2

Permit {O1}

P1
deny-overrides
O1
O2

Permit

N/A

R1
Deny
S1:manager
AT > 15:00 (M)
S2:clerk
AT > 15:00

R2
Permit
S:accountant
AT > 15:00 (M)

R3
Permit
S:accountant
AT > 15:00 (M)

R4
Permit
S:clerk
AT > 15:00 (M)

R5
Deny
S:accountant
A:read
Solution

Access Request
Alice
clerk
accountant
FinancialRecord
read

Applicable
PS
deny-overrides
R: FinancialRecord
O5
O6

Permit {O1}
P1
deny-overrides
O1
O2

N/A

R1
Deny
S1:manager
AT > 15:00 (M)
S2:clerk
AT > 15:00

R2
Permit
S:accountant
A:read

R3
Permit
S:accountant
AT > 15:00 (M)

R4
Permit
S:clerk
AT > 15:00 (M)

R5
Deny
S:accountant
A:read

Applicable
P2
first-applicable
O3
O4

Permit
I{P}

Applicable
PS
deny-overrides
R: FinancialRecord
O5
O6

N/A

I{P} → I

Applicable
**Solution**

**Access Request**
- Alice
- clerk
- accountant
- FinancialRecord
- read

**Applicable**
- **P1**  
  - deny-overrides  
  - O1  
  - O2

**P2**  
- first-applicable  
- O3  
- O4

**R1**  
- Deny  
- S1: manager
  - AT > 15:00 (M)
- S2: clerk
  - AT > 15:00

**R2**  
- Permit  
- S: accountant
  - AT > 15:00 (M)

**R3**  
- Permit  
- S: accountant
  - AT > 15:00 (M)

**R4**  
- Permit  
- S: clerk
  - AT > 15:00 (M)

**R5**  
- Deny  
- S: accountant

---

**PS**  
- deny-overrides  
- R: FinancialRecord
- O5
- O6
Solution

**Access Request**
- Alice
- clerk
- accountant
- FinancialRecord
- read

**Applicable**
- P1 (deny-overrides)
  - O1
  - O2
- P2 (first-applicable)
  - O3
  - O4

**Permit**
- R1 (Deny)
  - S1: manager
    - AT > 15:00 (M)
  - S2: clerk
    - AT > 15:00
- R2 (Permit)
  - S: accountant
    - A: read
- R3 (Permit)
  - S: accountant
    - AT > 15:00 (M)
- R4 (Permit)
  - S: clerk
    - AT > 15:00 (M)
- R5 (Deny)
  - S: accountant
    - A: read
Solution

Access Request

Alice
clerk
accountant
FinancialRecord
read

PS
deny-overrides
R: FinancialRecord
O5
O6

I{PD}

P1
deny-overrides
O1
O2

Permit {O1}

P2
first-applicable
O3
O4

I → I{PD}

N/A

Permit

R1
Deny
S1:manager
AT > 15:00 (M)
S2:clerk
AT > 15:00

R2
Permit
S:accountant
A:read

R3
Permit
S:accountant
AT > 15:00 (M)

R4
Permit
S:clerk
AT > 15:00 (M)

R5
Deny
S:accountant
A:read

Permit {O}

I{P} → I

I{PD}
Solution

Access Request
Alice
clerk
accountant
FinancialRecord
read

Permit \{O1\}

\begin{align*}
\text{P1} & \quad \text{deny-override} \\
& \quad \text{O1} \\
& \quad \text{O2} \\
\end{align*}

\begin{align*}
\text{P2} & \quad \text{first-applicable} \\
& \quad \text{O3} \\
& \quad \text{O4} \\
\end{align*}

\begin{align*}
\text{R1} & \quad \text{Deny} \\
& \quad \text{S1: manager} \\
& \quad \text{AT} > 15:00 (M) \\
& \quad \text{S2: clerk} \\
& \quad \text{AT} > 15:00
\end{align*}

\begin{align*}
\text{R2} & \quad \text{Permit} \\
& \quad \text{S: accountant} \\
& \quad \text{AT} > 15:00 (M)
\end{align*}

\begin{align*}
\text{R3} & \quad \text{Permit} \\
& \quad \text{S: accountant} \\
& \quad \text{AT} > 15:00 (M)
\end{align*}

\begin{align*}
\text{R4} & \quad \text{Permit} \\
& \quad \text{S: clerk} \\
& \quad \text{AT} > 15:00 (M)
\end{align*}

\begin{align*}
\text{R5} & \quad \text{Deny} \\
& \quad \text{S: accountant} \\
& \quad \text{A: read}
\end{align*}
Outline

- RBAC Profile
- Privacy Profile
- Exercise
Facilitate policy specification and management

- $RBAC_0$
  - Roles based on job functions
  - Users assigned to roles
  - Permissions assigned to roles
- $RBAC_1 = RBAC_0 + $Role hierarchy$
- $RBAC_2 = RBAC_0 + $Constraints$
- $RBAC_3 = RBAC_1 + RBAC_2$
RBAC Profile – Scope

RBAC profile answers three types of questions:

1. If a subject $X$ has roles $R_1, R_2, \ldots, R_n$ enabled, is $X$ allowed to perform a given action on a given resource?

2. Is subject $X$ allowed to have role $R_i$ enabled?

3. If a subject $X$ has roles $R_1, R_2, \ldots, R_n$ enabled, does that mean $X$ will have permissions associated with a given role $R'$?
   
   ▶ In other word, is role $R'$ either equal to or junior to any of roles $R_1, R_2, \ldots, R_n$?

NOT supported

▶ Separation of Duty
RBAC\(_0\)

- File system operations: read, write and execute
- DBMS operations: Insert, delete, and update

- User is associated with a session
- Gives roles activated by the session

Many-to-many relationship

One-to-many relationship
Policies in RBAC Profile

- **Role**: `<PolicySet>` that associates permissions with a role
- **Permission**: `<PolicySet>` that define permissions
- **Role Assignment**: `<Policy>` or `<PolicySet>` that defines which roles can be assigned to a subject
  - Actual assignment of roles to users outside scope of XACML PDP
- **HasPrivilegesOfRole**: `<Policy>` in a Permission `<PolicySet>` that supports requests asking whether a subject has the privileges associated with a given role
**RBAC\textsubscript{0} in XACML**

- **User** expressed using XACML Subjects
- **Roles**
  - Single attribute identifier and an attribute value for each role (recommended)
    - urn:oasis:names:tc:xacml:2.0:subject:role
  - An attribute identifier for each role (empty value)
    - urn:someapp:attributes:employee-role
  - Same representation within a policy domain
- **Objects** expressed using XACML Resources
- **Operations** expressed using XACML Actions
- **Permissions** expressed using Permission `<PolicySet>`
- **Permissions-Role Assignment** expressed using Role `<PolicySet>`
Role <PolicySet>

Determine whether a subject holds a certain role

- <Target> of Role <PolicySet> limits the applicability of the <PolicySet> to subjects holding a certain role

- Each Role <PolicySet> references a single Permission <PolicySet>
  - No contain or reference any other <Policy> or <PolicySet>
Permission \(<\text{PolicySet}\rangle\)

Define the permissions associated to a certain role

- **One Permission \(<\text{PolicySet}\rangle\) for each role**

- **Permission \(<\text{PolicySet}\rangle\) contains**
  \(<\text{Policy}\rangle\) and \(<\text{Rule}\rangle\) elements that describe the resources and actions that subjects are permitted to access

- **\(<\text{Target}\rangle\)** MUST NOT limit the subjects to which the \(<\text{PolicySet}\rangle\) is applicable
  
  - including contained or referenced PolicySet/Policy/Rule
Employee has the permission to create a purchase order
Role <PolicySet> for employees

```
<PolicySet xmlns="urn:oasis:names:tc:xacml:2.0:policy:schema:os"
    PolicySetId="RPS:employee:role"
    PolicyCombiningAlgId="permit-overrides">

    <Target>
        <AnyOf>
            <AllOf>
                <Match MatchId="&function;anyURI-equal">
                    <AttributeValue
                        DataType="&xml;anyURI">employee</AttributeValue>
                    <AttributeDesignator
                        MustBePresent="false"
                        Category="&subject-category;access-subject"
                        AttributeId="role"
                        DataType="&xml;anyURI"/>
                </Match>
            </AllOf>
        </AnyOf>
    </Target>

    <!-- Use permissions associated with the employee role -->
    <PolicySetIdReference>PPS:employee:role</PolicySetIdReference>

</PolicySet>
```
Permission <PolicySet> for employees

```xml
<PolicySet xmlns="urn:oasis:names:tc:xacml:2.0:policy:schema:os"
    PolicySetId="PPS:employee:role"
    PolicyCombiningAlgId="permit-overrides">

<!-- Permissions specifically for the employee role -->
<Policy PolicyId="Permissions:specifically:for:the:employee:role"
    RuleCombiningAlgId="permit-overrides">

<!-- Permission to create a purchase order -->
<Rule RuleId="Permission:to:create:a:purchase:order" Effect="Permit">
   <Target>
      <AnyOf>
         <AllOf>
            <Match MatchId="&function;string-equal">
               <AttributeValue DataType="&xml;string">purchase order</AttributeValue>
               <AttributeDesignator DataType="&xml;string"
                    Category="&category;resource"
                    AttributeId="&resource;resource-id"/>
            </Match>
         </AllOf>
      </AnyOf>
      <AnyOf>
         <AllOf>
            <Match MatchId="&function;string-equal">
               <AttributeValue DataType="&xml;string">create</AttributeValue>
               <AttributeDesignator DataType="&xml;string"
                    Category="&category;action"
                    AttributeId="&action;action-id"/>
            </Match>
         </AllOf>
      </AnyOf>
   </Target>
</Rule>
</Policy>
</PolicySet>
```
Evaluation

- Permission `<PolicySet>` cannot be used as initial policy by PDP

- Permission `<PolicySet>` MUST be reachable only through the corresponding Role Permission `<PolicySet>`
  - To support inheritance, Permission `<PolicySet>` must be applicable to any subject
Role inheritance defined by policy structure

Permissions associated with one role includes permissions associated with more general roles
**Permission** `<PolicySet>`

Define the permissions associated to a certain role

- One Permission `<PolicySet>` for each role

- Permission `<PolicySet>` contains `<Policy>` and `<Rule>` elements that describe the resources and actions that subjects are permitted to access

- `<Target>`, if present, MUST NOT limit the subjects to which the `<PolicySet>` is applicable
  - including contained or referenced PolicySet/Policy/Rule

- Permission `<PolicySet>` MAY also contain references to Permission `<PolicySet>`s associated with other roles that are generalizations of the given role
  - Inherit all permissions associated with the role of the referenced Permission `<PolicySet>`
Example

- Two roles:
  - **Employee** has the permission to create a purchase order
  - **Manager** has the permission to sign a purchase order

- Role hierarchy
  - Manager is a specialization of Employee

- Manager has permission associated with Employee

```
Employee's Role <PolicySet>
  <Target> employee
  Reference to Employee's Permission <PolicySet>
  Employee's Permission <PolicySet>
    <Target>
    <PolicySet> create purchase order

Manager's Role <PolicySet>
  <Target> manager
  Reference to Manager's Permission <PolicySet>
  Manager's Permission <PolicySet>
    <Target>
    <PolicySet> sign purchase order
    Reference to Employee's Permission <PolicySet>
```
Multi-Role Permissions

A user must hold several roles simultaneously to gain access to a certain permission

- <Target> of Role <PolicySet> requires specifying all roles
  - A single <AllOf> element containing multiple <Match> elements

- Associated Permission <PolicySet> specifies permission associated with subjects who have all role enabled

- Associated Permission <PolicySet> may refer to Permission <PolicySet>s associated with individual roles

- All necessary roles have to be specified in Request context

Subjects has the permissions associated with

- each role individually
- multi roles
HasPrivilegesOfRole <Policy>

- Used to allow queries asking if a subject “has the privileges of” a specific role

- Included in a Permission <PolicySet>

- If this type of request is to be supported, then a HasPrivilegesOfRole <Policy> MUST be included in each Permission <PolicySet>
HasPrivilegesOfRole < Policy > for managers

<Policy PolicyId="Permission:to:have:manager:role:permissions"
         RuleCombiningAlgId="permit-overrides">

<!-- Permission to have manager role permissions -->
<Rule RuleId="Permission:to:have:manager:permissions" Effect="Permit">

  <Condition>
    <Apply FunctionId="and">
      <Apply FunctionId="anyURI-is-in">
        <AttributeValue
          DataType="anyURI">manager</AttributeValue>
        <ResourceAttributeDesignator
          AttributeId="role" DataTypes="anyURI"/>
      </Apply>
      <Apply FunctionId="anyURI-is-in">
        <AttributeValue
          DataType="anyURI">hasPrivilegesofRole</AttributeValue>
        <ActionAttributeDesignator
          AttributeId="action-id" DataTypes="anyURI"/>
      </Apply>
    </Apply>
  </Condition>

</Rule>
</Policy>
HasPrivilegesOfRole<Request>

<Request>
   <Attributes Category="&subject-category;access-subject">
      <Attribute AttributeId="&subject;subject-id"
         IncludeInResult="false">
         <AttributeValue DataType="&xml;string">
            Anne
         </AttributeValue>
      </Attribute>
   </Attributes>
   <Attributes Category="&category;resource">
      <Attribute AttributeId="&role;"
         IncludeInResult="false">
         <AttributeValue DataType="&xml;anyURI">
            &roles;manager
         </AttributeValue>
      </Attribute>
   </Attributes>
   <Attributes Category="&category;action">
      <Attribute AttributeId="&action;action-id"
         IncludeInResult="false">
         <AttributeValue DataType="&xml;anyURI">
            &actions;hasPrivilegesOfRole
         </AttributeValue>
      </Attribute>
   </Attributes>
</Request>
Outline

▸ RBAC Profile
▸ Privacy Profile
▸ Exercise
Privacy-aware Access Control

Access control policies include

- **Subject**
- **Object**
- **Actions**

Privacy-aware access control policies also include

- **Purpose**: the intended use of data
- **Obligations**: mandatory requirements to be fulfilled
- **Conditions**: restrictions under which a policy is applied
Privacy Profile

- No element for purpose

- Two optional purpose attributes:
  - \texttt{urn:oasis:names:tc:xacml:2.0:resource:purpose}
    - purpose for which data were collected
  - \texttt{urn:oasis:names:tc:xacml:2.0:action:purpose}
    - purpose for which data are requested

- No purpose hierarchy
<?xml version="1.0" encoding="UTF-8"?>
<Rule xmlns="urn:oasis:names:tc:xacml:2.0:policy:schema:os"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="urn:oasis:names:tc:xacml:2.0:policy:schema:os"
    RuleId="urn:oasis:names:tc:xacml:2.0:matching-purpose"
    Effect="Permit">
  <Condition>
    <FunctionId="urn:oasis:names:tc:xacml:2.0:function:regexp-string-match">
      <ResourceAttributeDesignator>
        <AttributeId="urn:oasis:names:tc:xacml:2.0:resource:purpose"
          DataType="http://www.w3.org/2001/XMLSchema#string"/>
      </ResourceAttributeDesignator>
      <ActionAttributeDesignator>
        <AttributeId="urn:oasis:names:tc:xacml:2.0:action:purpose"
          DataType="http://www.w3.org/2001/XMLSchema#string"/>
      </ActionAttributeDesignator>
    </FunctionId>
  </Condition>
</Rule>
XSPA Profile

Cross-Enterprise Security and Privacy Authorization (XSPA) Profile of XACML v2.0 for Healthcare

- Healthcare Treatment
- Emergency Treatment
- System Administration
- Operations
- Payment
- Research
- Marketing
- Public Health
Summary

XACML can be extended to cope with specific issues

- RBAC profile
- Privacy profile
- Others
  - XACML v3.0 Administration and Delegation Profile Version 1.0
  - XACML v3.0 Hierarchical Resource Profile Version 1.0
  - SAML 2.0 Profile of XACML, Version 2.0
  - XACML v3.0 XML Digital Signature Profile Version 1.0
Text is in the repository
Hints

- Build policy hierarchy
- Annotate it with relevant information
  - Target
  - Combination algorithm
  - Obligations (and FulfillOn)
- Match target with request context
- Response = decision + set of obligations
Solution
Solution (accountant)

Alice
take
accountant
engineer
make
order

RPS: accountant permit-overrides

PPS: accountant permit-overrides

P: accountant permit-overrides

R: accountant permit-overrides

PPS: employee permit-overrides

P: employee permit-overrides

R: employee permit read order
Solution (accountant + engineer)

Alice
accountant
engineer
make
order
Solution (engineer)

Alice
accountant
engineer
make
order

RPS:engineer
permit-overrides

PPS:engineer
permit-overrides

P:engineer
permit-overrides

R:engineer
permit
make
order

P:employee
permit-overrides

PPS:employee
permit-overrides

R:employee
Permit
read
order
References

- Cross-Enterprise Security and Privacy Authorization (XSPA) Profile of XACML v2.0 for Healthcare (Committee Draft)