

# eXtensible Access Control Markup Language

## XACML im Vergleich mit P3P und EPAL

Stefan Berthold

Technische Universität Dresden  
Fakultät Informatik

28. Juni 2004

# Thematischer Umriß

- 1 P3P und EPAL
- 2 XACML – eXtensible Access Control Markup Language
- 3 Rollenbasierte Zugriffssteuerung mit XACML

# Thematischer Umriß

- 1 P3P und EPAL
- 2 XACML – eXtensible Access Control Markup Language
- 3 Rollenbasierte Zugriffssteuerung mit XACML

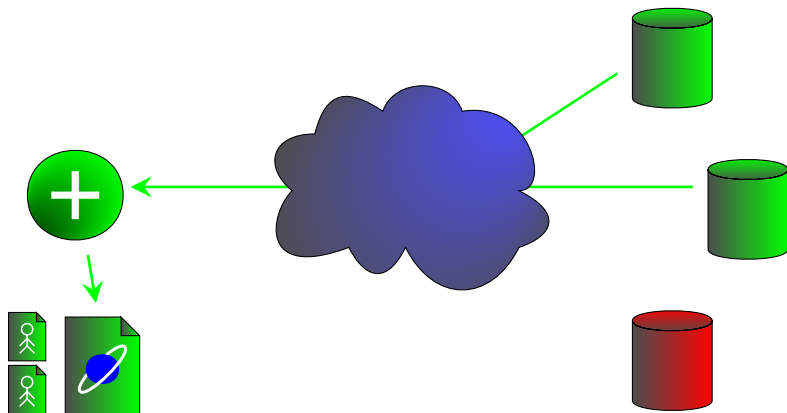
# Thematischer Umriß

- 1 P3P und EPAL
- 2 XACML – eXtensible Access Control Markup Language
- 3 Rollenbasierte Zugriffssteuerung mit XACML

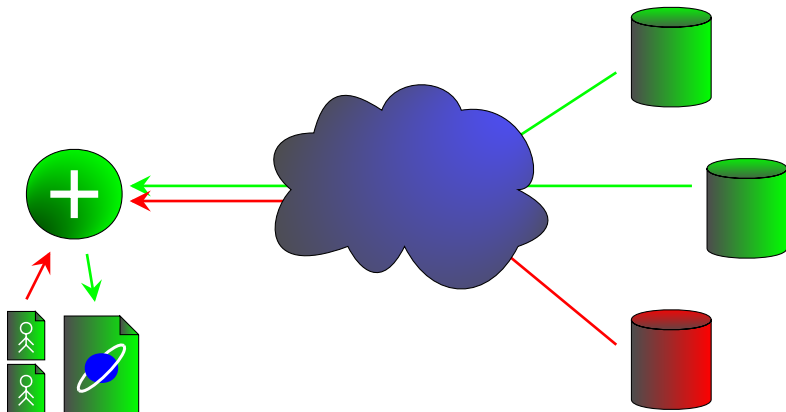
# Thematischer Umriß

- 1 P3P und EPAL
  - P3P – Platform for Privacy Preferences
  - EPAL – Enterprise Privacy Authorization Language
- 2 XACML – eXtensible Access Control Markup Language
- 3 Rollenbasierte Zugriffssteuerung mit XACML

# Motivationsbildchen (P3P)



# Motivationsbildchen (P3P)



# P3P als Grundlage automatisierter Unterhändler



## Automatische Aushandlung von Kommunikationsbedingungen:

- Selbstverpflichtung der Dienstleister u.a. in:
  - Sammlung personenbezogener Daten,
  - Verwendung,
  - Dauer der Speicherung,
  - Weitergabe
- Definition von Datenschutzrichtlinie durch Nutzer

Anwendung für WWW, dennoch sehr allgemein gehalten.

# P3P als Grundlage automatisierter Unterhändler



## Automatische Aushandlung von Kommunikationsbedingungen:

- Selbstverpflichtung der Dienstanbieter u.a. in:
  - Sammlung personenbezogener Daten,
  - Verwendung,
  - Dauer der Speicherung,
  - Weitergabe
- Definition von Datenschutzrichtlinie durch Nutzer

Anwendung für WWW, dennoch sehr allgemein gehalten.

# P3P als Grundlage automatisierter Unterhändler



## Automatische Aushandlung von Kommunikationsbedingungen:

- Selbstverpflichtung der Dienstleister u.a. in:
  - Sammlung personenbezogener Daten,
  - Verwendung,
  - Dauer der Speicherung,
  - Weitergabe
- Definition von Datenschutzrichtlinie durch Nutzer

Anwendung für WWW, dennoch sehr allgemein gehalten.

# P3P als Grundlage automatisierter Unterhändler

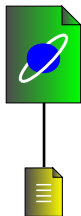


## Automatische Aushandlung von Kommunikationsbedingungen:

- Selbstverpflichtung der Dienstanbieter u.a. in:
  - Sammlung personenbezogener Daten,
  - Verwendung,
  - Dauer der Speicherung,
  - Weitergabe
- Definition von Datenschutzrichtlinie durch Nutzer

Anwendung für WWW, dennoch sehr allgemein gehalten.

# Anspruch auf Standard, der Vollständigkeit wegen



Formulierungsproblem:

- XML-Dialekt

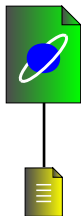
Transportproblem:

- Hypertext Transfer Protocol (HTTP)

Problem des Auffindens:

- „placed in an easy-to-locate manner“  
(vier Vorschläge im Standard erwähnt)

# Anspruch auf Standard, der Vollständigkeit wegen



Formulierungsproblem:

- XML-Dialekt

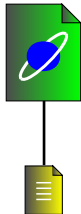
Transportproblem:

- Hypertext Transfer Protocol (HTTP)

Problem des Auffindens:

- „placed in an easy-to-locate manner“  
(vier Vorschläge im Standard erwähnt)

# Anspruch auf Standard, der Vollständigkeit wegen



Formulierungsproblem:

- XML-Dialekt

Transportproblem:

- Hypertext Transfer Protocol (HTTP)

Problem des Auffindens:

- „placed in an easy-to-locate manner“  
(vier Vorschläge im Standard erwähnt)

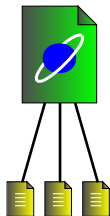
# Formulierung der Selbstverpflichtungen („Policies“)

Sicherheitsangaben sind:

- Einordnung der Daten („data-categories“)
- Empfänger („recipients“)
- Verwendung („uses“)

Notation:

- eine „Policy“ pro Datei
- mehrere zusammengefaßt
- beliebige Kombination



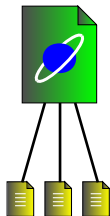
# Formulierung der Selbstverpflichtungen („Policies“)

Sicherheitsangaben sind:

- Einordnung der Daten („data-categories“)
- Empfänger („recipients“)
- Verwendung („uses“)

Notation:

- eine „Policy“ pro Datei
- mehrere zusammengefaßt
- beliebige Kombination



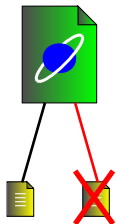
# Einschränkungen

## Beschränkungen des Standards:

- nur Zusagen möglich (keine Verneinung von Verwendungszwecken)
- Geltungsbereich: abgedeckte Daten

## Resultierende Praxis:

- Nutzer braucht Richtlinie für nicht abgedeckte Daten.
- Gesetzestreue der propagierten Aktion nicht nachprüfbar.



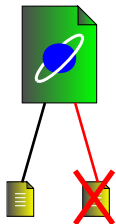
# Einschränkungen

## Beschränkungen des Standards:

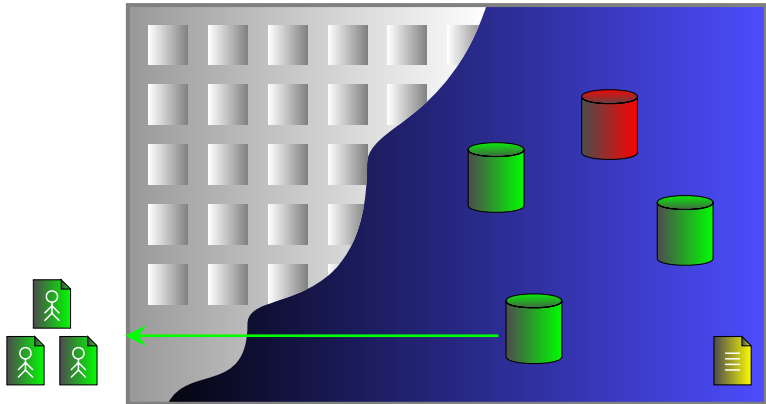
- nur Zusagen möglich (keine Verneinung von Verwendungszwecken)
- Geltungsbereich: abgedeckte Daten

## Resultierende Praxis:

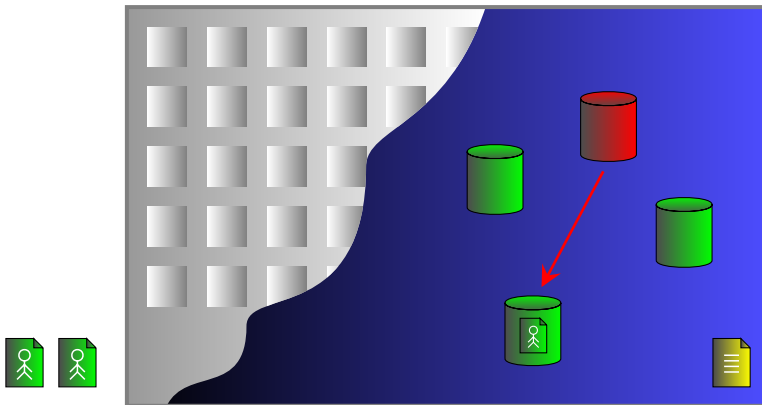
- Nutzer braucht Richtlinie für nicht abgedeckte Daten.
- Gesetzestreue der propagierten Aktion nicht nachprüfbar.



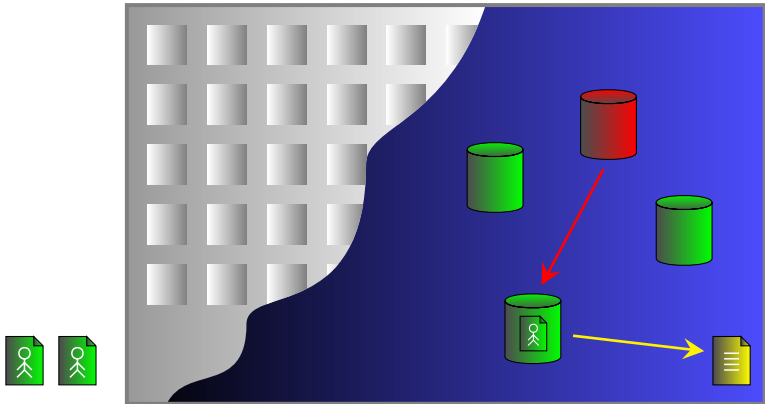
# Motivationsbildchen (EPAL)



# Motivationsbildchen (EPAL)



# Motivationsbildchen (EPAL)



# EPAL als Grundlage interner Zugriffssteuerung

Effektive Verarbeitung von Datenschutzinformationen durch:

- Standardisierung der Sprache für Datenschutzinformationen
- Maschinelle Verarbeitung und Durchsetzung innerhalb von Unternehmen
- Maschineller Austausch dieser Informationen über:
  - Abteilungsgrenzen
  - Filialen
  - Unternehmensgrenzen



# EPAL als Grundlage interner Zugriffssteuerung

Effektive Verarbeitung von Datenschutzinformationen durch:

- Standardisierung der Sprache für Datenschutzinformationen
- Maschinelle Verarbeitung und Durchsetzung innerhalb von Unternehmen
- Maschineller Austausch dieser Informationen über:
  - Abteilungsgrenzen
  - Filialen
  - Unternehmensgrenzen



# Umsetzung mit oder als Erweiterung von P3P

## Innerhalb des Unternehmens: EPAL



- an Unternehmensbedürfnisse anpassbarer Standard (selbst definierbares Vokabular)
- enthält P3P-Elemente zur leichteren Konvertierung

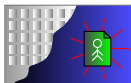
## Austausch von Informationen: P3P

- übergreifend festgeschriebener Standard



# Umsetzung mit oder als Erweiterung von P3P

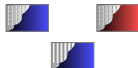
## Innerhalb des Unternehmens: EPAL



- an Unternehmensbedürfnisse anpassbarer Standard (selbst definierbares Vokabular)
- enthält P3P-Elemente zur leichteren Konvertierung

## Austausch von Informationen: P3P

- übergreifend festgeschriebener Standard



# Umsetzung mit oder als Erweiterung von P3P

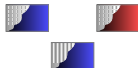
## Innerhalb des Unternehmens: EPAL



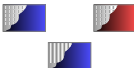
- an Unternehmensbedürfnisse anpassbarer Standard (selbst definierbares Vokabular)
- enthält P3P-Elemente zur leichteren Konvertierung

## Austausch von Informationen: P3P

- übergreifend festgeschriebener Standard



# Umsetzung mit oder als Erweiterung von P3P



## Innerhalb des Unternehmens: EPAL

- an Unternehmensbedürfnisse anpassbarer Standard (selbst definierbares Vokabular)
- enthält P3P-Elemente zur leichteren Konvertierung

## Austausch von Informationen: P3P

- übergreifend festgeschriebener Standard

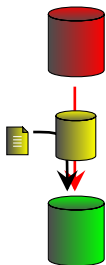
# Durchsetzung, der Vollständigkeit wegen



Zur Durchsetzung drei Methoden möglich:

- 1 Zielprogramm implementiert EPAL-Schnittstelle selbst.
- 2 Zielprogramm wird durch Monitor von Daten getrennt (Prinzip Firewall).
- 3 Zentrale Bewertung der eingehenden und ausgehenden Informationen von Zielprogrammen bzw. Netzwerken.

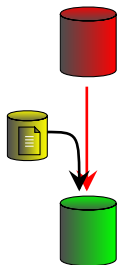
# Durchsetzung, der Vollständigkeit wegen



Zur Durchsetzung drei Methoden möglich:

- 1 Zielprogramm implementiert EPAL-Schnittstelle selbst.
- 2 Zielprogramm wird durch Monitor von Daten getrennt (Prinzip Firewall).
- 3 Zentrale Bewertung der eingehenden und ausgehenden Informationen von Zielprogrammen bzw. Netzwerken.

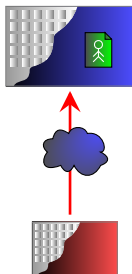
# Durchsetzung, der Vollständigkeit wegen



Zur Durchsetzung drei Methoden möglich:

- 1 Zielprogramm implementiert EPAL-Schnittstelle selbst.
- 2 Zielprogramm wird durch Monitor von Daten getrennt (Prinzip Firewall).
- 3 Zentrale Bewertung der eingehenden und ausgehenden Informationen von Zielprogrammen bzw. Netzwerken.

# P3P-Zusagen aus EPAL-Richtlinie



Die Konvertierung von EPAL-Richtlinien zu P3P-Zusagen ist...

- leicht, wenn Vokabular weitgehend übereinstimmt,
- etwas aufwendig, aber möglich, sonst.

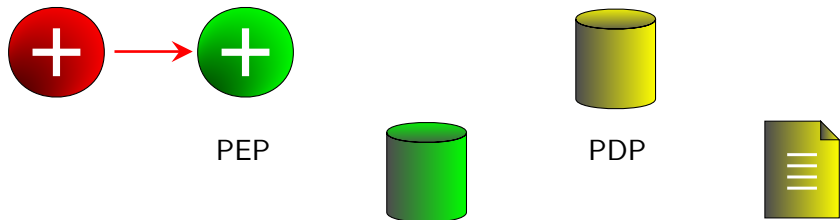
# Thematischer Umriß

- 1 P3P und EPAL
- 2 XACML – eXtensible Access Control Markup Language
  - Motivation
  - Anfrage-Sprache
  - Policy-Sprache
- 3 Rollenbasierte Zugriffssteuerung mit XACML

## Motivation (XACML)

XACML bietet erweiterbaren Rahmen für:

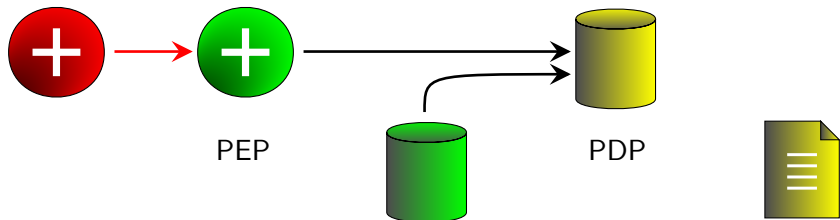
- Anfrage-Sprache (Request/Response)
- Policy-Sprache zur Zugriffssteuerung (und -kontrolle)



# Motivation (XACML)

XACML bietet erweiterbaren Rahmen für:

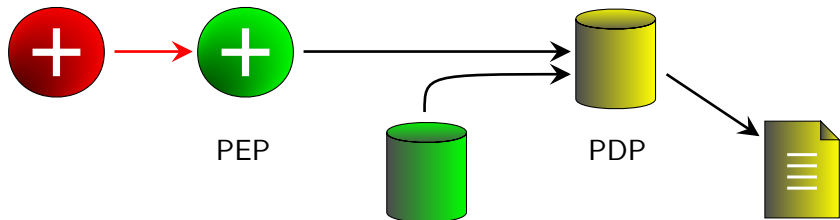
- Anfrage-Sprache (Request/Response)
- Policy-Sprache zur Zugriffssteuerung (und -kontrolle)



# Motivation (XACML)

XACML bietet erweiterbaren Rahmen für:

- Anfrage-Sprache (Request/Response)
- Policy-Sprache zur Zugriffssteuerung (und -kontrolle)



# Anfragen in XACML

## Beispiel (XACML Request)

```
<Request>  
  <Subject><Attribut/></Subject>  
  <Resource><Attribut/></Resource>  
  <Action><Attribut/></Action>  
  <Environment/>  
</Request>
```

## Frage

*Sind P3P-Zusicherungen in XACML-Anfragen formulierbar?*

# Anfragen in XACML

## Beispiel (XACML Request)

```
<Request>  
  <Subject><Attribut/></Subject>  
  <Resource><Attribut/></Resource>  
  <Action><Attribut/></Action>  
  <Environment/>  
</Request>
```

## Frage

*Sind P3P-Zusicherungen in XACML-Anfragen formulierbar?*

# Einfache Anpassung: OECD Privacy Guidelines

## Standard

- ...
- *Offenlegung des Zwecks einer Datenerhebungen*

## Beispiel

- Einführung: Request- und Policy-Purpose-Attribut
- Einführung einer Regel mit Effect="Permit"
- Bedingung: Purpose-Attribute stimmen überein

# Zugriffssteuerung in XACML

## Beispiel (XACML Policy)

```
<Policy RuleCombiningAlgId="permit-overrides">  
  <Description/>  
  <Target/>  
  <Rule Effect="Permit">  
    <Target><Subjects/></Target>  
  </Rule>  
</Policy>
```

## Frage

*Sind EPAL-Richtlinien in XACML-Anfragen formulierbar?*

# Zugriffssteuerung in XACML

## Beispiel (XACML Policy)

```
<Policy RuleCombiningAlgId="permit-overrides">  
  <Description/>  
  <Target/>  
  <Rule Effect="Permit">  
    <Target><Subjects/></Target>  
  </Rule>  
</Policy>
```

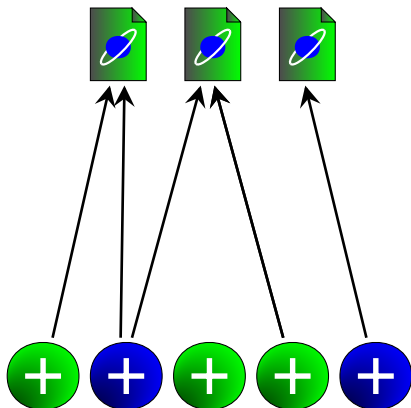
## Frage

*Sind EPAL-Richtlinien in XACML-Anfragen formulierbar?*

# Thematischer Umriß

- 1 P3P und EPAL
- 2 XACML – eXtensible Access Control Markup Language
- 3 Rollenbasierte Zugriffssteuerung mit XACML
  - Motivation
  - Funktionale Struktur von RBAC
  - Kritik an RBAC

## Motivation (ACL vs. RBAC)



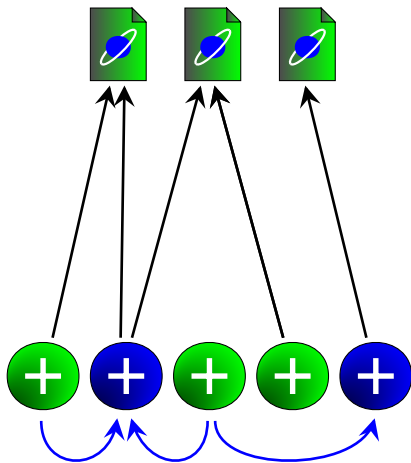
Beispiel (ACL)

Gruppen bündeln Nutzer.

Beispiel (RBAC)

Rollen bündeln **Aufgaben** für Nutzer.

## Motivation (ACL vs. RBAC)



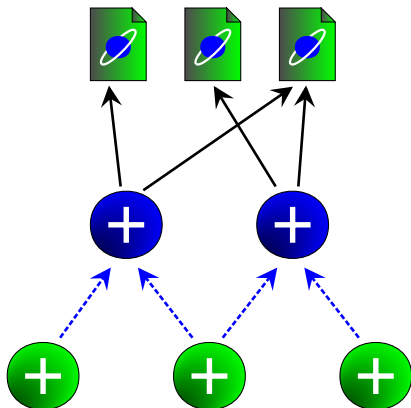
### Beispiel (ACL)

Gruppen bündeln Nutzer.

### Beispiel (RBAC)

Rollen bündeln **Aufgaben** für Nutzer.

## Motivation (ACL vs. RBAC)



Beispiel (ACL)

Gruppen bündeln Nutzer.

Beispiel (RBAC)

Rollen bündeln **Aufgaben** für Nutzer.

# Grundkonzept (RBAC<sub>0</sub>)

## Standard

- *Nutzer*
- *Rollen*
- *Objekte*
- *Operationen*
- *Berechtigungen*
- *Rechteverteilung*
- *Rollenverteilung*
- *Sitzungen*

# Grundkonzept (RBAC<sub>0</sub>)

## Standard

- *Nutzer*
- *Rollen*
- *Objekte*
- *Operationen*
- *Berechtigungen*  $\subseteq$  *Operationen*  $\times$  *Objekte*
- *Rechteverteilung*
- *Rollenverteilung*
- *Sitzungen*

# Grundkonzept (RBAC<sub>0</sub>)

## Standard

- *Nutzer*
- *Rollen*
- *Objekte*
- *Operationen*
- *Berechtigungen*  $\subseteq$  *Operationen*  $\times$  *Objekte*
- *Rechteverteilung*  $\subseteq$  *Rollen*  $\times$  *Berechtigungen*
- *Rollenverteilung*
- *Sitzungen*

# Grundkonzept (RBAC<sub>0</sub>)

## Standard

- *Nutzer*
- *Rollen*
- *Objekte*
- *Operationen*
- *Berechtigungen*  $\subseteq$  *Operationen*  $\times$  *Objekte*
- *Rechteverteilung*  $\subseteq$  *Rollen*  $\times$  *Berechtigungen*
- *Rollenverteilung*  $\subseteq$  *Nutzer*  $\times$  *Rollen*
- *Sitzungen*

# Grundkonzept (RBAC<sub>0</sub>)

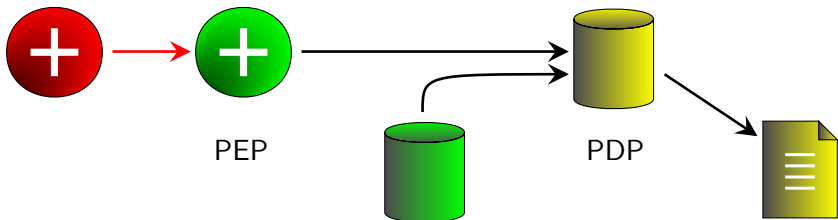
## Standard

- *Nutzer*
- *Rollen*
- *Objekte*
- *Operationen*
- *Berechtigungen*  $\subseteq$  *Operationen*  $\times$  *Objekte*
- *Rechteverteilung*  $\subseteq$  *Rollen*  $\times$  *Berechtigungen*
- *Rollenverteilung*  $\subseteq$  *Nutzer*  $\times$  *Rollen*
- *Sitzungen*

# Motivation (XACML)

XACML bietet erweiterbaren Rahmen für:

- Anfrage-Sprache (Request/Response)
- Policy-Sprache zur Zugriffssteuerung (und -kontrolle)



# RBAC<sub>0</sub> in XACML

## Standard

- *Nutzer*
- *Rollen*
- *Objekte*
- *Operationen*
- *Berechtigungen*
- *Rechteverteilung*
- *Rollenverteilung*
- *Sitzungen*

- XACML <Subject>

# RBAC<sub>0</sub> in XACML

## Standard

- *Nutzer*
- *Rollen*
- *Objekte*
- *Operationen*
- *Berechtigungen*
- *Rechteverteilung*
- *Rollenverteilung*
- *Sitzungen*

- XACML Attribut des `<Subject>`

# RBAC<sub>0</sub> in XACML

## Standard

- *Nutzer*
- *Rollen*
- *Objekte*
- *Operationen*
- *Berechtigungen*
- *Rechteverteilung*
- *Rollenverteilung*
- *Sitzungen*

- XACML <Resource>

# RBAC<sub>0</sub> in XACML

## Standard

- *Nutzer*
- *Rollen*
- *Objekte*
- *Operationen*
- *Berechtigungen*
- *Rechteverteilung*
- *Rollenverteilung*
- *Sitzungen*

- XACML `<Action>`

# RBAC<sub>0</sub> in XACML

## Standard

- Nutzer
- Rollen
- Objekte
- Operationen
- **Berechtigungen**
- Rechteverteilung
- Rollenverteilung
- Sitzungen

- Rollen-<PolicySet>
- <PolicySet>-Instanz für  
eigentliche Berechtigungen

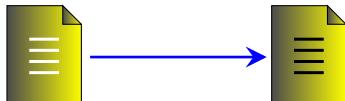


# RBAC<sub>0</sub> in XACML

## Standard

- Nutzer
- Rollen
- Objekte
- Operationen
- Berechtigungen
- **Rechteverteilung**
- Rollenverteilung
- Sitzungen

- Verbinden der PolicySets von Rolle und Berechtigung mit `<PolicySetIdReference>`



# RBAC<sub>0</sub> in XACML

## Standard

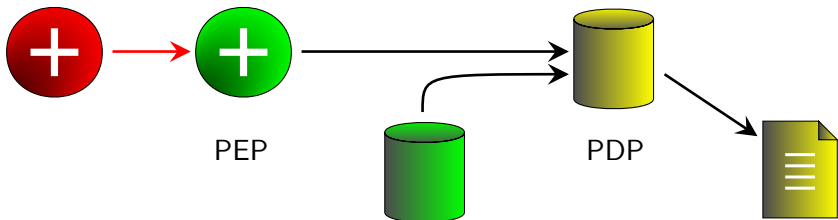
- *Nutzer*
- *Rollen*
- *Objekte*
- *Operationen*
- *Berechtigungen*
- *Rechteverteilung*
- *Rollenverteilung*
- *Sitzungen*

Nicht in PDP!

# Motivation (XACML)

XACML bietet erweiterbaren Rahmen für:

- Anfrage-Sprache (Request/Response)
- Policy-Sprache zur Zugriffssteuerung (und -kontrolle)



# RBAC<sub>0</sub> in XACML

## Standard

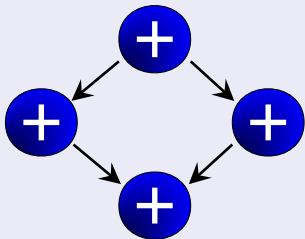
- *Nutzer*
- *Rollen*
- *Objekte*
- *Operationen*
- *Berechtigungen*
- *Rechteverteilung*
- *Rollenverteilung*
- *Sitzungen*

Nicht beschrieben!

## Hierarchiekonzept (RBAC<sub>1</sub>)

### Standard

- *Hinzunahme einer Rollenhierarchie*
- *Vererbung der Berechtigungen*



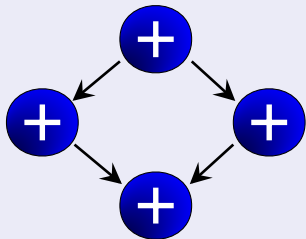
<PolicySetIdReference> in  
Senior- zur Junior-Rolle



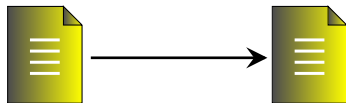
## Hierarchiekonzept (RBAC<sub>1</sub>)

### Standard

- *Hinzunahme einer Rollenhierarchie*
- *Vererbung der Berechtigungen*



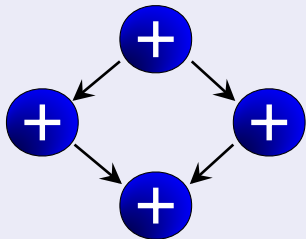
<PolicySetIdReference> in  
Senior- zur Junior-Rolle



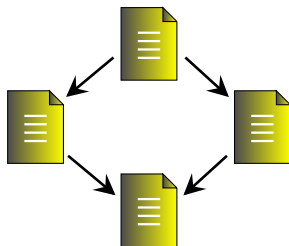
## Hierarchiekonzept (RBAC<sub>1</sub>)

### Standard

- *Hinzunahme einer Rollenhierarchie*
- *Vererbung der Berechtigungen*



`<PolicySetIdReference>` in  
Senior- zur Junior-Rolle



# Aufteilung der Zuständigkeit (RBAC<sub>2</sub>)

„Separation of Duty“

## Standard

- *Statisch*
  - *Beschränkung der Anzahl der Berechtigungen pro Nutzer*
  - *Definition von Rollen-Konflikten*
- *Dynamisch*
  - *Beschränkung der Berechtigungen in Situationen, wie*
    - *vorheriger Besitz von Rollen oder Berechtigungen*
    - *Situation anderer Nutzer*

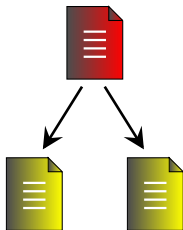
# Aufteilung der Zuständigkeit (RBAC<sub>2</sub>)

„Separation of Duty“

## Standard

- *Statisch*
  - *Beschränkung der Anzahl der Berechtigungen pro Nutzer*
  - *Definition von Rollen-Konflikten*
- *Dynamisch*
  - *Beschränkung der Berechtigungen in Situationen, wie*
    - *vorheriger Besitz von Rollen oder Berechtigungen*
    - *Situation anderer Nutzer*

# Statische Aufteilung der Zuständigkeit in XACML



## Beispiel

Zwei Rollen sollen nicht gleichzeitig benutzbar sein:

- Einführung eines zusätzlichen `<PolicySet>`
- `<Target>`: beide `<Subject>`-Attribute per UND-Verknüpfung
- `<Rule>` mit `Effect="Deny"`
- `<PolicySetIdReference>` für jede betroffene Rolle

# Aufteilung der Zuständigkeit (RBAC<sub>2</sub>)

„Separation of Duty“

## Standard

- *Statisch*
  - *Beschränkung der Anzahl der Berechtigungen pro Nutzer*
  - *Definition von Rollen-Konflikten*
- *Dynamisch*
  - *Beschränkung der Berechtigungen in Situationen, wie*
    - *vorheriger Besitz von Rollen oder Berechtigungen*
    - *Situation anderer Nutzer*

# Dynamische Aufteilung der Zuständigkeit in XACML

## Problem

- in „Policy Decision Point“ keine Rollen-Zuweisung möglich
- dynamische Trennung verlangt aber genau das

## Beispiel (Regel für Rollenzuweisung)

Seth oder Anne sollen max. 2 Rollen gleichzeitig aktivieren dürfen.

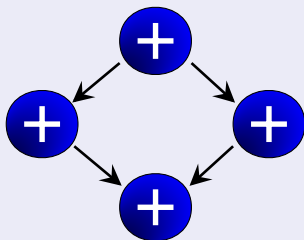
- neue <Rule> mit Effect="Permit"
- <Target> mit
  - zwei <Subject> für Seth und Anne
  - <Action> sucht nach enable
- <Condition> vergleicht Größe der „Bag“ auf  $< 2$ .

## Hierarchische Einschränkungen (RBAC<sub>3</sub>)

### Standard

*Hinzunahme von Zusicherungen für Hierarchie, bspw.:*

- *keine Mehrfachvererbung*
- *Zulassung höchstens einer mächtigeren Rolle*



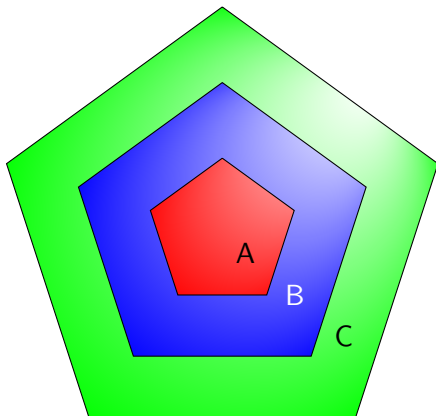
- **Nicht beschrieben!**

# Kritik an RBAC



- Keine Zonen-Separation durch Rollen möglich, da mehr als eine Rolle aktiv sein darf und keine Wege durch die Rollen definierbar (*Rollengeschichte* wird verworfen)
- Datenfluß bei komplexen Systemen nicht klar nachvollziehbar (beispielsweise):
  - 1 Umschalten auf Rolle mit mehr Rechten
  - 2 Lesen von Daten mithilfe der erworbenen Rechte
  - 3 Verzicht auf die zusätzlichen RechteDaten bleiben im geringeren Sicherheitsniveau weiterhin verfügbar!

# Zonen-Separation



## Definition (erlaubte Aktionen)

- schreiben nur von außen nach innen
- lesen nur von innen nach außen

# Kritik an RBAC



- Keine Zonen-Separation durch Rollen möglich, da mehr als eine Rolle aktiv sein darf und keine Wege durch die Rollen definierbar (*Rollengeschichte* wird verworfen)
- Datenfluß bei komplexen Systemen nicht klar nachvollziehbar (beispielsweise):

- 1 Umschalten auf Rolle mit mehr Rechten
- 2 Lesen von Daten mithilfe der erworbenen Rechte
- 3 Verzicht auf die zusätzlichen Rechte

Daten bleiben im geringeren Sicherheitsniveau weiterhin verfügbar!

# Kritik an RBAC

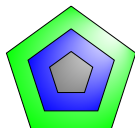


- Keine Zonen-Separation durch Rollen möglich, da mehr als eine Rolle aktiv sein darf und keine Wege durch die Rollen definierbar (*Rollengeschichte* wird verworfen)
- Datenfluß bei komplexen Systemen nicht klar nachvollziehbar (beispielsweise):

- 1 Umschalten auf Rolle mit mehr Rechten
- 2 Lesen von Daten mithilfe der erworbenen Rechte
- 3 Verzicht auf die zusätzlichen Rechte

Daten bleiben im geringeren Sicherheitsniveau weiterhin verfügbar!

# Kritik an RBAC



- Keine Zonen-Separation durch Rollen möglich, da mehr als eine Rolle aktiv sein darf und keine Wege durch die Rollen definierbar (*Rollengeschichte* wird verworfen)
- Datenfluß bei komplexen Systemen nicht klar nachvollziehbar (beispielsweise):
  - 1 Umschalten auf Rolle mit mehr Rechten
  - 2 **Lesen von Daten mithilfe der erworbenen Rechte**
  - 3 Verzicht auf die zusätzlichen RechteDaten bleiben im geringeren Sicherheitsniveau weiterhin verfügbar!

# Kritik an RBAC

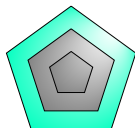


- Keine Zonen-Separation durch Rollen möglich, da mehr als eine Rolle aktiv sein darf und keine Wege durch die Rollen definierbar (*Rollengeschichte* wird verworfen)
- Datenfluß bei komplexen Systemen nicht klar nachvollziehbar (beispielsweise):

- 1 Umschalten auf Rolle mit mehr Rechten
- 2 Lesen von Daten mithilfe der erworbenen Rechte
- 3 **Verzicht auf die zusätzlichen Rechte**

Daten bleiben im geringeren Sicherheitsniveau weiterhin verfügbar!

# Kritik an RBAC



- Keine Zonen-Separation durch Rollen möglich, da mehr als eine Rolle aktiv sein darf und keine Wege durch die Rollen definierbar (*Rollengeschichte* wird verworfen)
- Datenfluß bei komplexen Systemen nicht klar nachvollziehbar (beispielsweise):
  - 1 Umschalten auf Rolle mit mehr Rechten
  - 2 Lesen von Daten mithilfe der erworbenen Rechte
  - 3 Verzicht auf die zusätzlichen RechteDaten bleiben im geringeren Sicherheitsniveau weiterhin verfügbar!

## Einschränkungen zur Abhilfe



Umgehung wäre folgendermaßen möglich:

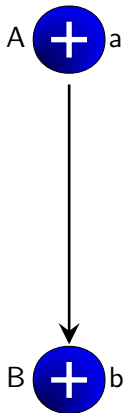
- Beschränkung der jeweils aktiven Rollen pro Nutzer auf **eine**
- Einführung einer „Kompatibilität“ von Rollen („Übergangsmatrix“, nicht notwendigerweise reflexiv)

Das ist in RBAC nicht möglich!

Frage

*Ist die Umgehung in XACML beschreibbar?*

## Einschränkungen zur Abhilfe



Umgehung wäre folgendermaßen möglich:

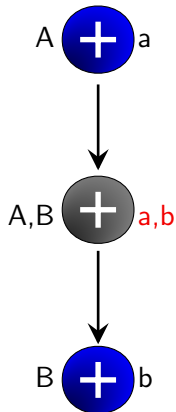
- Beschränkung der jeweils aktiven Rollen pro Nutzer auf **eine**
- Einführung einer „Kompatibilität“ von Rollen („Übergangsmatrix“, nicht notwendigerweise reflexiv)

Das ist in RBAC nicht möglich!

Frage

*Ist die Umgehung in XACML beschreibbar?*

# Abhilfe durch XACML



## Beispiel (mögliche Abhilfe durch XACML)

- Nutzer in zwei Zuständen möglich:
  - ① eine Rolle, deren Berechtigungen
  - ② zwei Rollen, keine aktiven Berechtigungen
- Kompatibilitäts- $\langle$ PolicySet $\rangle$ , pro kompatiblen Paar Rollen:
  - Einführung einer  $\langle$ Policy $\rangle$
  - $\langle$ Target $\rangle$  die beiden Rollen
  - $\langle$ Rule $\rangle$  mit Effect="Permit"
  - $\langle$ Target $\rangle$  „Ursprungsrolle“
  - $\langle$ Action $\rangle$  ist disable

# Zusammenfassung

XACML im Vergleich zu anderen Zugriffssteuerungssprachen:

- als Anfrage-Sprache besser als P3P
- als Policy-Sprache ungefähr genauso gut wie EPAL

XACML in Verwendung komplexer Zugriffssteuerung:

- RBAC bis auf Rollenzuweisung und RBAC<sub>3</sub>
- mit Verknüpfung zweier XACML-Systeme möglicherweise verbessertes RBAC komplett

## Zum Weiterlesen...



### XACML-Standard

[http://www.oasis-open.org/committees/tc\\_home.php?wg\\_abbrev=xacml](http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=xacml)



### P3P-Standard

<http://www.w3.org/TR/P3P/>



### EPAL

<http://www.zurich.ibm.com/security/enterprise-privacy/epal/Specification/index.html>



### RBAC-Standard

<http://csrc.nist.gov/rbac/>



### RC-Modell (RSBAC)

<http://www.rsbac.org/documentation/rc-nordsec2002/index.html>