RBAC mit OpenLDAP: OpenRBAC

Treffen des ZKI-AK Verzeichnisdienste,

Duisburg, 4.-5.10.2010

Peter Gietz, Markus Widmer,
DAASI International GmbH

Peter.gietz@daasi.de

DAASI International

Directory Applications
for Advanced Security
and Information Management

Agenda

- Motivationen für rollenbasierte Zugriffskontrolle
- Der RBAC-Standard
- XACML
- OpenRBAC
- Warum mit OpenLDAP

DAASI International
Directory Applications
for Advanced Security
and Information Management

Motivation für rollenbasierte Zugriffskontrolle

- Durch die Verwendung von Rollen wird die Zugriffskontrolle wesentlich übersichtlicher
- Wenn ein Benutzer die Rolle wechselt (z.B. von Student zu Systemadmin) bekommt er automatisch die entsprechenden Rechte
- Es werden keine Sonderlösungen eingeführt, die dann nicht dokumentiert sind und vergessen werden
- Die reale, gewachsene Organisationsstruktur wird durch Rollen (z.B. Student, Professor, Sekretärin) abgebildet
- Veränderungen in der Organisationsstruktur können einfach in das RBAC-System übernommen werden
- Es gibt bewährte Standards (RBAC und XACML)

DAASI International
Directory Applications for Advanced Security and Information Management

Voraussetzungen

- Klares Rollenkonzept
- Rollen müssen in Benutzerverwaltung abgebildet sein
- Anwendungen müssen entsprechende Informationen verwerten können
- Identity Management ist hierbei sehr hilfreich
- Auch über Föderationsinfrastrukturen (Shibboleth) können Rolleninformationen transportiert werden

DAASI International
Directory Applications for Advanced Security and Information Management

Der RBAC-Standard

- Der ANSI-Standard RBAC teilt sich auf in drei Bereiche:
  - RBAC Core
  - Hierarchical RBAC (zwei Typen)
  - Separation of Duty (zwei Typen, die im Standard als eigene Bereiche gezählt werden)
- Wichtige Komponenten sind:
  - Benutzer
  - Rolle
  - Session
  - Berechtigung
  - Ressource

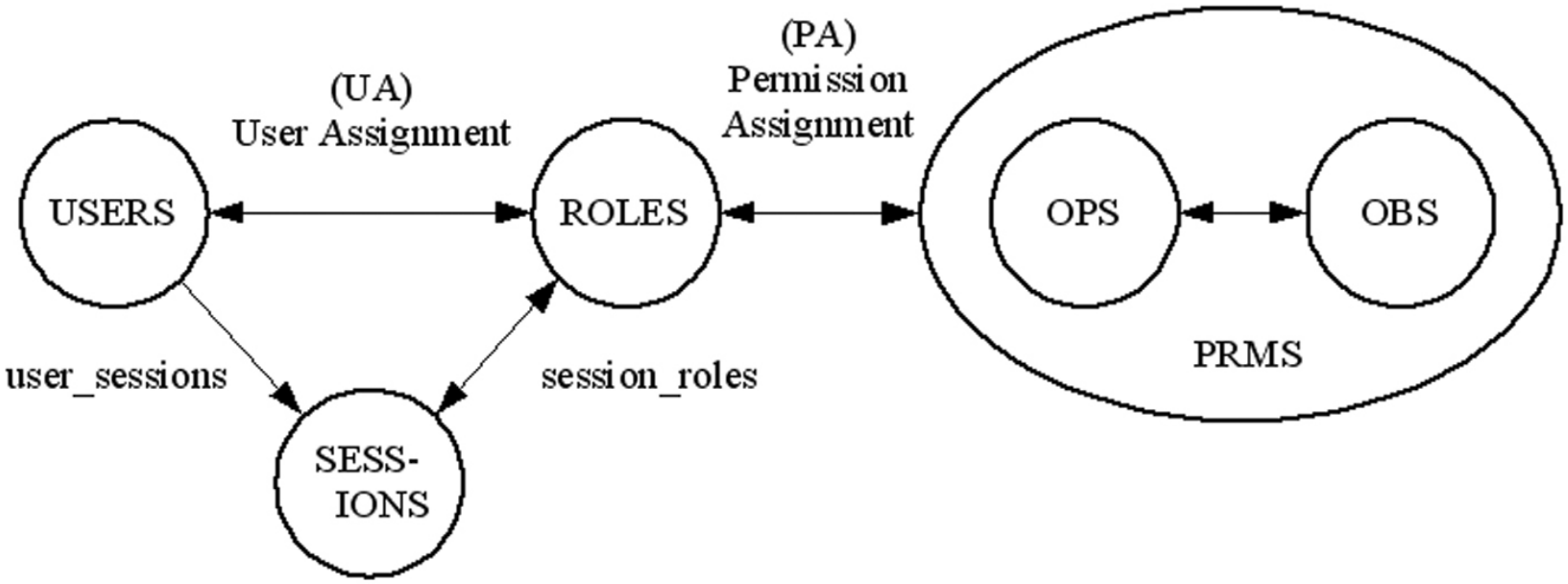
DAASI International
Directory Applications for Advanced Security and Information Management

RBAC-Core

- Definiert grundlegende Funktionen, die eine RBAC-Implementierung beinhalten muss. Dazu gehören:
  - Anlegen und Löschen eines Benutzers, einer Rolle oder einer Session
  - Hinzufügen und Entfernen von Berechtigungen auf Ressourcen
- Definiert die Funktion checkAccess, mit der eine Zugriffsentscheidung angefordert werden kann
- Definiert weitere Funktionen zum
  - Ändern von Beziehungen der Komponenten untereinander
  - Abfrage von Informationen zu den einzelnen Komponenten des Systems

DAASI International
Directory Applications for Advanced Security and Information Management

RBAC-Core



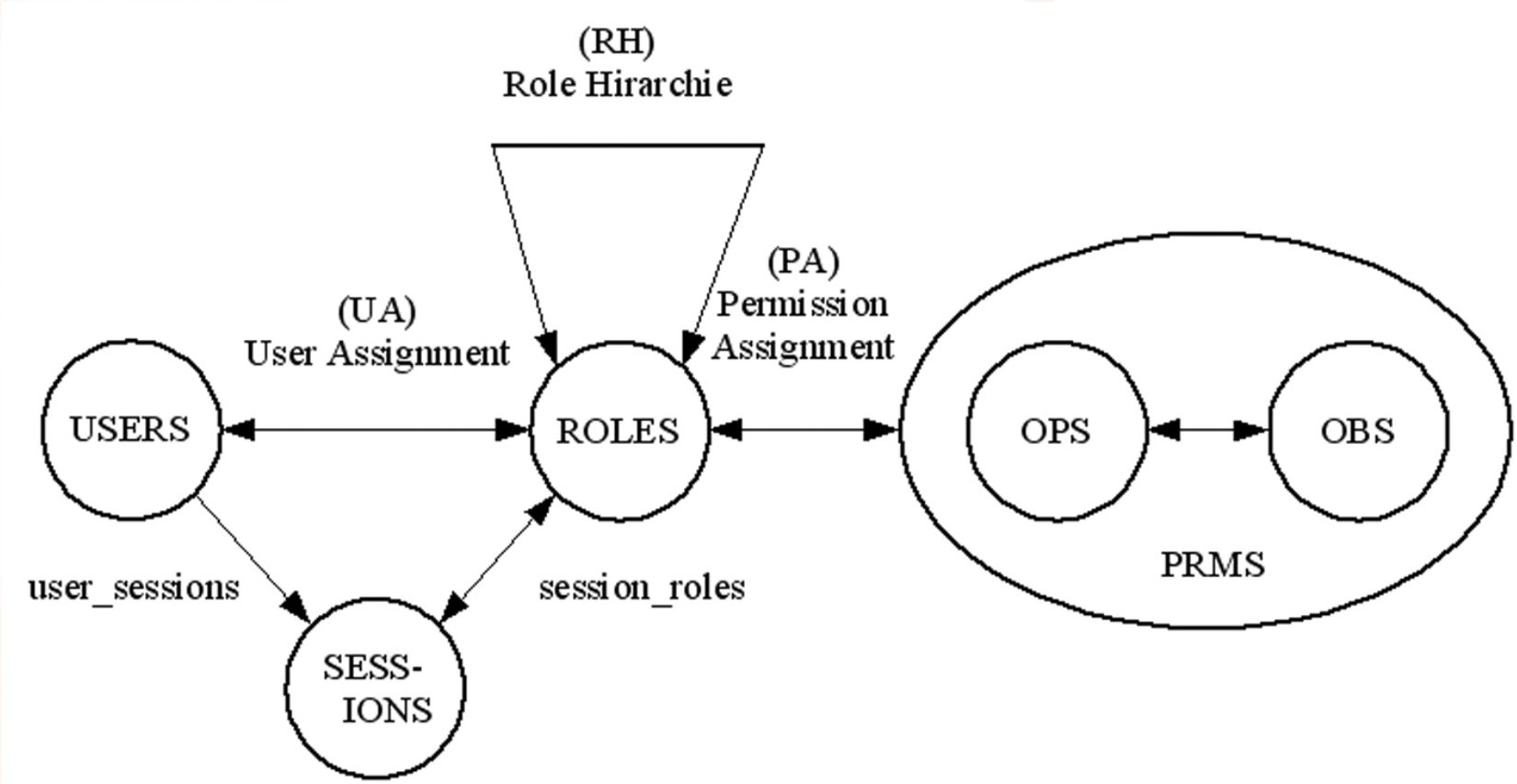
DAASI International
Directory Applications
for Advanced Security
and Information Management

Hierarchical RBAC

- Erweitert die Grundfunktionalität um Rollenhierarchien, wobei es zwei Typen gibt:
  - Limited Role Hierarchy: Rollen sind in Baumstrukturen geordnet (jeweils nur ein Elternknoten)
  - General Role Hierarchy: Rollen können in freien Graphen organisiert sein (beliebig viele Elternknoten)
- Hierbei werden einige im RBAC-Core definierte Funktionen angepasst:
  - z.B. die Funktion addActiveRole, die Rollen in einer Session aktiviert, muss nun auch über die Hierarchie implizit geerbte Rollen berücksichtigen
- Darüber hinaus werden neue Funktionen definiert zum Ändern der Hierarchien von Rollen

DAASI International
Directory Applications for Advanced Security and Information Management

Hierarchical RBAC



DAASI International
Directory Applications
for Advanced Security
and Information Management

Separation of Duty

- Um zu verhindern, dass ein Benutzer gleichzeitig sehr unterschiedliche Rollen ausübt, wird als weiterer Zusatz eine Pflichtentrennung eingeführt
- Über sogenannte Sets werden sich gegenseitig ausschließende Rollen definiert
- Es wird unterschieden zwischen
  - Static Separation of Duty
  - Dynamic Separation of Duty

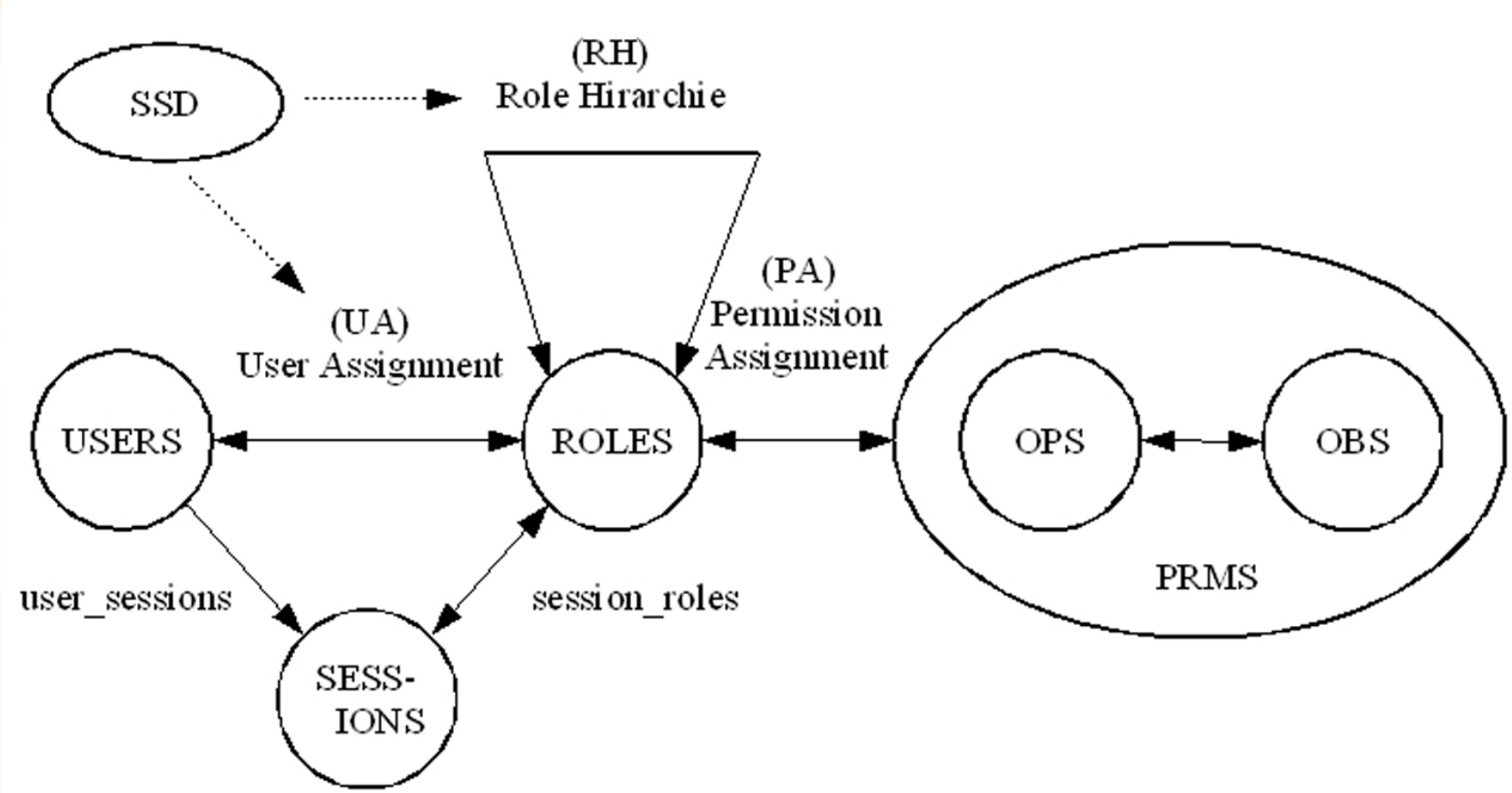
DAASI International
Directory Applications for Advanced Security and Information Management

Static Separation of Duty (SSD)

- Statisch bedeutet hier: „über die Zeit nahezu unveränderlich“
- SSD-Sets definieren sich grundsätzlich ausschließende Rollen, die kein Benutzer gleichzeitig innehaben darf
- Die in SSD-Sets definierten Einschränkungen werden bei jeder Zuweisung von einer Rolle zu einem Benutzer angewendet

DAASI International
Directory Applications for Advanced Security and Information Management

Static Separation of Duty (SSD)



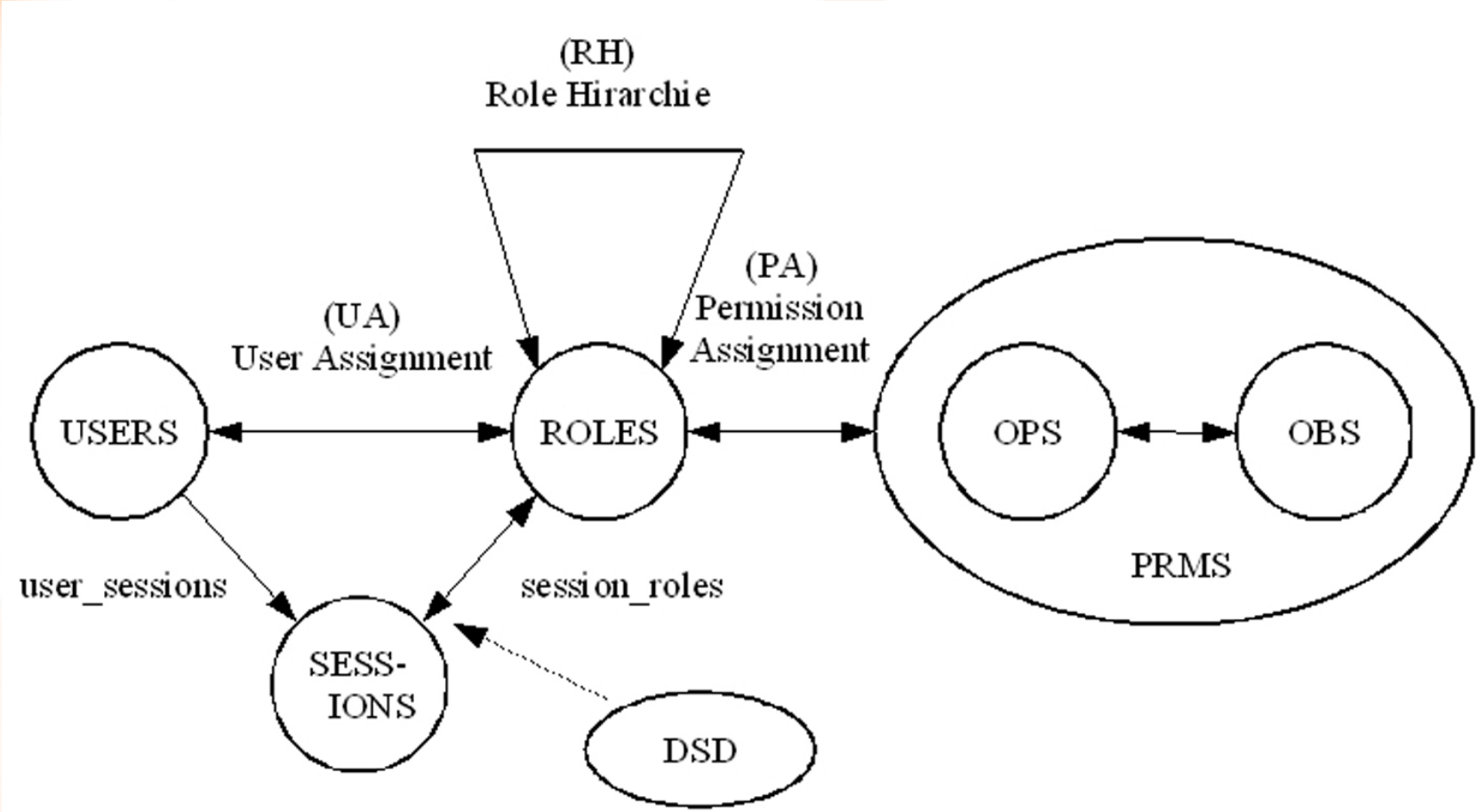
DAASI International
Directory Applications
for Advanced Security
and Information Management

Dynamic Separation of Duty (DSD)

- Im Unterschied zum SSD werden die Einschränkungen erst beim Aktivieren von Rollen (also „zur Laufzeit“) in einer Session geprüft
- Hierdurch kann es einem Benutzer z.B. möglich sein die Rollen „Antragssteller“ und „Antragsprüfer“ auszufüllen, jedoch nicht gleichzeitig
- Eine aktive Rolle zu haben bedeutet, im Unterschied zu einer nicht aktiven Rolle, dass diese in einer Session eingetragen ist

DAASI International
Directory Applications for Advanced Security and Information Management

Dynamic Separation of Duty (DSD)



DAASI International
Directory Applications
for Advanced Security
and Information Management

Erweiterbarkeit von RBAC

- Das durch den Standard definierte RBAC stellt bereits eine sehr umfangreiche Bibliothek zur Verfügung, mit der viele Autorisierunganforderungen erfüllt werden können
- Diese Funktionalität kann aber auch sehr leicht erweitert werden, da der Standard selbst bereits in einzelne Bereiche gegliedert ist, die aufeinander aufbauen.
- Ein Beispiel hierfür ist „Multi-session Separation of Duties“ (David Chadwick, 2006):
  - Eine Erweiterung, in der ein Benutzer immer nur eine Session gleichzeitig starten kann

DAASI International
Directory Applications for Advanced Security and Information Management

XACML

- Extended Access Control Markup Language
- OASIS-Standard
- Es gibt XACML-Profile für SAML und LDAP/DSML, sowie für RBAC
- Generisch: Zugriffsregeln (Policy) können anwendungsunabhängig spezifiziert werden
- Verteilt: Eine Policy kann sich auf eine andere an einem anderen Ort gespeicherte Policy beziehen
- Komplex: Im Prinzip eine eigene Programmiersprache
- Setzt sich deshalb nur langsam durch

DAASI International
Directory Applications for Advanced Security and Information Management

XACML-Elemente

- PolicySet: Container für Policies bzw. weitere PolicySets
- Policies und PolicySets können mit Algorithmen kombiniert werden
- Bedingungen setzen sich zusammen aus Subjekt, Ressource und Aktion
- Environment
- Policy Decision Point (PDP)
- Policy Enforcement Point (PEP)
- Target: Sammlung vereinfachter Bedingungen
- Rule: Access Control Regeln
- Attribute

DAASI International
Directory Applications for Advanced Security and Information Management

XACML-Elemente

- XACML spezifiziert:
  - Element <Policy> in dem Policy-Regeln abgebildet werden
  - Element <Request> in dem eine Zugriffskontrollabfrage spezifiziert werden kann

DAASI International
Directory Applications for Advanced Security and Information Management

XACML-Policy

- Eine Policy besteht aus zwei Teilen:
  - Einleitender Bereich im Element <Target>
  - Die einzelnen Regeln im Element <Rule>

DAASI International
Directory Applications for Advanced Security and Information Management

XACML-Policy-Target

- Im Target-Element sind die Elemente <Subjects>, <Resources> und <Actions> enthalten
- Diese Angaben sind keine Pflicht, sondern bezeichnen die Anwendbarkeit der gesamten Policy ohne die einzelnen in der Policy enthaltenen Regeln zu betrachten
- Werden die darin enthaltenen Bedingungen zum boolschen Wert TRUE ausgewertet, so wird die Policy angewendet
- Werden sie zu FALSE ausgewertet, werden die in dieser Policy spezifizierten Regeln nicht weiter ausgewertet
- Wird eines der genannten Elemente nicht angegeben, so wird es zu TRUE ausgewertet

DAASI International
Directory Applications for Advanced Security and Information Management

XACML-Policy-Rules

- Im Rule-Element sind die Elemente <Target> und <Condition> enthalten
- Jedes <Target> wiederum besteht aus den Elementen <Subjects>, <Resources> und <Actions>.
- Werden diese nicht näher spezifiziert, werden sie zum boolschen Wert TRUE ausgewertet
- Das <Condition>-Element enthält Vergleichsoperationen auf Attributwerten des XACML-Requests und logische Verknüpfungen mehrerer solcher Vergleichsoperationen
- Insgesamt wird <Condition> wieder zu einem boolschen Wert ausgewertet

DAASI International
Directory Applications for Advanced Security and Information Management

XACML-Policy-Beispiel 1/2

```
<Policy PolicyId="SamplePolicy"
        RuleCombiningAlgId="urn:oasis:names:tc:xacml:1.0:
        rule-combining-algorithm:permit-overrides">
  <Target>
      <Subjects/>
      <Resources>
        <ResourceMatch
MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
         <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">SampleServer
         </AttributeValue>
         <ResourceAttributeDesignator
DataType="http://www.w3.org/2001/XMLSchema#string"

AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-
id"/>
        </ResourceMatch>
      </Resources>
      <Actions>
        <AnyAction/>
      </Actions>
  </Target>
```

DAASI International
Directory Applications for Advanced Security and Information Management

XACML-Policy Beispiel 2/2

```
<Rule RuleId="Rule-X1-1" Effect="Permit">
  <Target>
    <Subjects />
    <Resources />
    <Actions />
  </Target>
  <Condition>
    <Expression />
  </Condition>
</Rule>
<Rule>...</Rule>
</Policy>
```


DAASI International
Directory Applications for Advanced Security and Information Management

XACML-Request

- XACML-Request besteht aus drei Elementen:
  - Subject: Das Objekt (Person), das Zugriff auf eine Ressource erhalten möchte. Es wird mit den zur Auswertung der Policies notwendigen Attributen übergeben. Im Kontext von RBAC sind dies vor allem die aktivierten Rollen.
  - Resource: Das Objekt, auf das zugegriffen werden soll. Auch dieses wird mit den zur Auswertung der Policies notwendigen Attributen übergeben.
  - Action: Die Operation, die ausgeführt werden soll. Eine solche Operation wäre z.B. lesender Zugriff auf die Ressource.

DAASI International
Directory Applications for Advanced Security and Information Management

XACML-Request-Protokoll

- Als Request-Response-Protokoll kann SAML über SOAP verwendet werden
- So kann ein XACML Policy Decision Point über einen Web Service abgefragt werden

DAASI International
Directory Applications for Advanced Security and Information Management

XACML-Request Beispiel 1/2

```
<?xml version="1.0" encoding="UTF-8"?>
<Request xmlns="urn:oasis:names:tc:xacml:2.0:context:schema:os"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="urn:oasis:names:tc:xacml:2.0:context:schema
:os
      http://docs.oasis-open.org/xacml/access_control-xacml-
2.0-context-schema-os.xsd">
  <Subject>
    <Attribute AttributeId=
           "urn:oasis:names:tc:xacml:1.0:subject:subject-id"
           DataType="http://www.w3.org/2001/XMLSchema#string">
      <AttributeValue>foo</AttributeValue>
    </Attribute>
    <Attribute> … </Attribute>
  </Subject>
  <Resource>
    <Attribute AttributeId=
        "info:fooproject:resource:item-id"
        DataType="http://www.w3.org/2001/XMLSchema#anyURI">
      <AttributeValue>f6839544-38f6-4a87-874d-b0435cbbc699
      </AttributeValue
    </Attribute>
  </Resource>
```

DAASI International
Directory Applications for Advanced Security and Information Management

XACML-Request Beispiel 2/2

```
<Action>
  <Attribute AttributeId=
     "urn:oasis:names:tc:xacml:1.0:action:action-id"
      DataType="http://www.w3.org/2001/XMLSchema#string">
    <AttributeValue>
      info:fooproject:action:retrieve-content
    </AttributeValue>
  </Attribute>
</Action>
<Environment/>
</Request>
```

DAASI International

Directory Applications for Advanced Security and Information Management

OpenRBAC

- OpenRBAC ist eine OpenSource-Implementierung des RBAC-Standards
- Im Rahmen einer Diplomarbeit von Markus Widmer bei DAASI International entstanden
- Im Rahmen von mehreren Grid-Forschungsprojekten von DAASI eingesetzt und weiterentwickelt worden
- Implementiert den gesamten Standard außer General Role Hierarchy (Da die Rollenhierarchien im LDAP-Baum abgebildet werden)
- Die RBAC-Funktionen werden von OpenRBAC selbst geschützt
- Stabile Version und Dokumentation unter http://www.openrbac.de

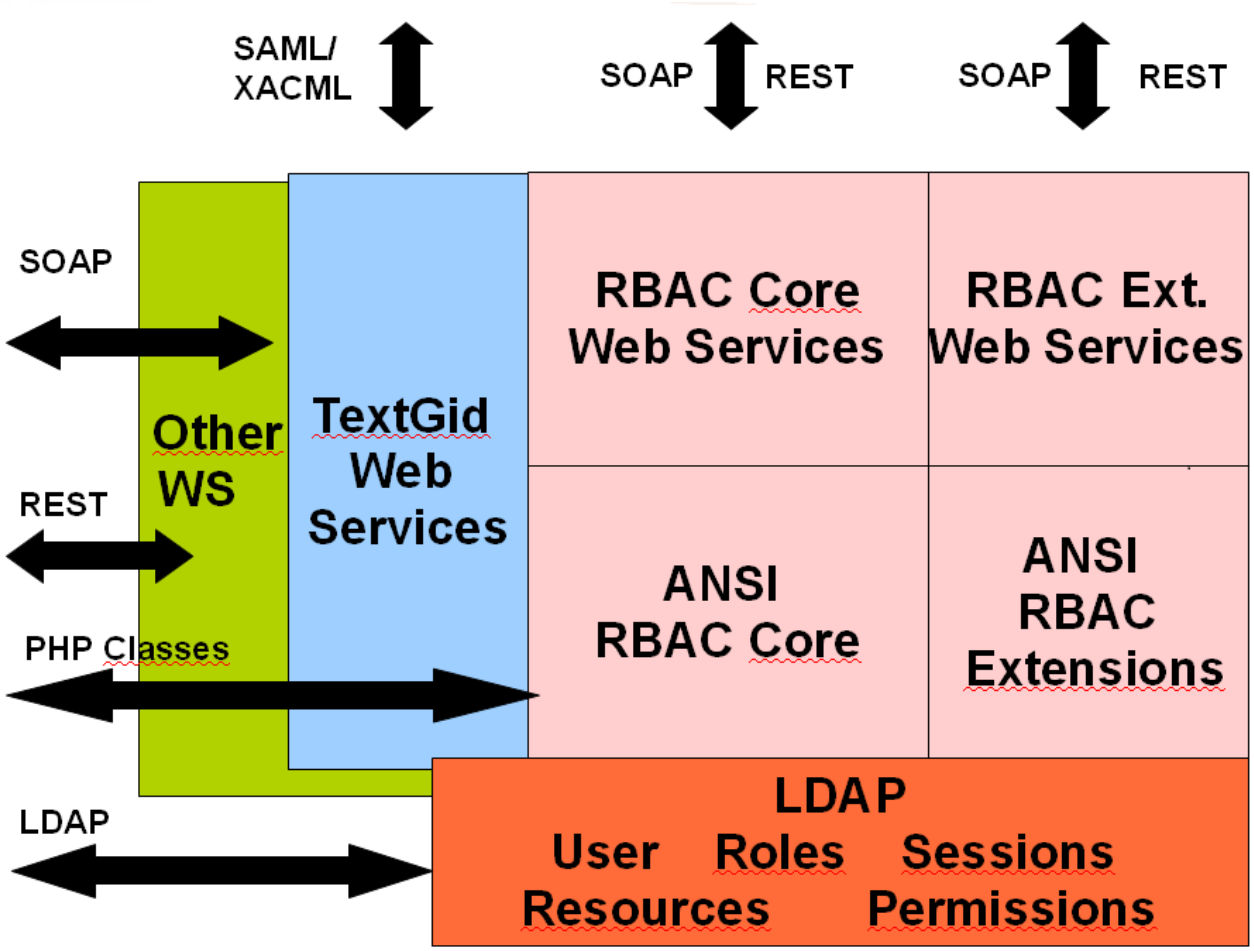
DAASI International
Directory Applications for Advanced Security and Information Management

OpenRBAC-Schichten-Modell

- OpenRBAC ist in verschiedenen Schichten implementiert:
  - Kern (Datenbackend) ist ein OpenLDAP-Server mit entsprechend definiertem DIT und Schema
  - Die RBAC-Funktionen sind als PHP-Klassen implementiert, wobei die drei Bereiche Core, Hierarchical und Separation of Duty gekapselt sind
  - Die PHP-Klassen können über Web-Service Wrapper als Web Services aufgerufen werden
  - Erweiterte Webservices können ebenfalls auf die einzelnen RBAC-Methoden zugreifen
  - Die Funktion Check-Access ist auch über das XACML/SAML-Protokoll abfragbar

DAASI International
Directory Applications for Advanced Security and Information Management

OpenRBAC Schichten Modell



DAASI International
Directory Applications
for Advanced Security
and Information Management

Warum OpenLDAP als Backend

- Teile der Informationen (über die Benutzer) sind meistens ohnehin schon in einem LDAP-Server
- Die einzelnen Informationen sind in verschiedenen Teilbäumen organisiert (ou=Roles, ou=Resources, ou=Sessions, etc) und können, wo sinnvoll auch auf verschiedenen LDAP-Servern liegen (z.B. User auf dem Authentifizierungsserver und der Rest auf einem eigenen OpenRBAC-Server
- Das entwickelte Datenmodell ist grundsätzlich mit XACML kompatibel

DAASI International
Directory Applications for Advanced Security and Information Management

Warum OpenLDAP als Backend

- (Open)LDAP kann sehr schnell Antworten auf Access-Fragen geben
  - CheckAccess ist über einen einzigen LDAP-Filter realisierbar
  - Ein OpenLDAP-Server auf einem starken Rechner (Multi-Core mit 48 GB RAM) kann auch bei großen Datenmengen (z.B. 1 Million Ressourcen) über 60.000 Abfragen pro Sekunde beantworten

DAASI International
Directory Applications for Advanced Security and Information Management

OpenRBAC im Projekt TextGrid

- TextGrid entwickelt eine Virtuelle Forschungsumgebung für Geisteswissenschaftler, augenblicklich für:
  - Editionsphilologen, Linguisten, Musikwissenschaftler und Kunsthistoriker
- Die TextGrid-Software besteht aus zwei Bausteinen
  - TextGridLab : GUI und Webservices basierte Workbench
  - TextGridRep: Middleware zur Verwaltung der Informationsobjekte im Grid
- Für die Authentifizierungs- und Autorisierungsinfrastruktur werden LDAP, OpenRBAC und Shibboleth eingesetzt

DAASI International
Directory Applications for Advanced Security and Information Management

TextGridLab

- Das TextGridLab (Laboratory) bietet Zugriff auf fachwissenschaftliche Werkzeuge, Services und Inhalte
  - GUI über Eclipse Rich Client (RCP) und verteilte als WebServices realisierte Werkzeuge
  - Webbasierte GUI für Retrieval
  - Workbench beliebig über neue Web Services erweiterbar

DAASI International
Directory Applications for Advanced Security and Information Management

TextGridRep

- Das TextGridRep (Repository) ist ein im Grid verteiltes Repository für geisteswissenschaftliche Forschungsdaten und zielt auf langfristige Verfügbarkeit und Zugänglichkeit
- Hält wohl definierte (Web Service-) Schnittstellen für TextGridLab bereit
- Besteht aus:
  - TG-auth* für Autorisierung und Authentifizierung (Shibboleth und OpenRBAC)
  - TG-search: XML-Datenbank für Metadaten und Volltexte, RDF-Datenbank für Relationen
  - TG-crud Service (create/retrieve/update/delete)
- Überbrückt die LDAP/Shibboleth-AAI und die PKI-basierte Grid Sicherheitsinfrastruktur (GSI)

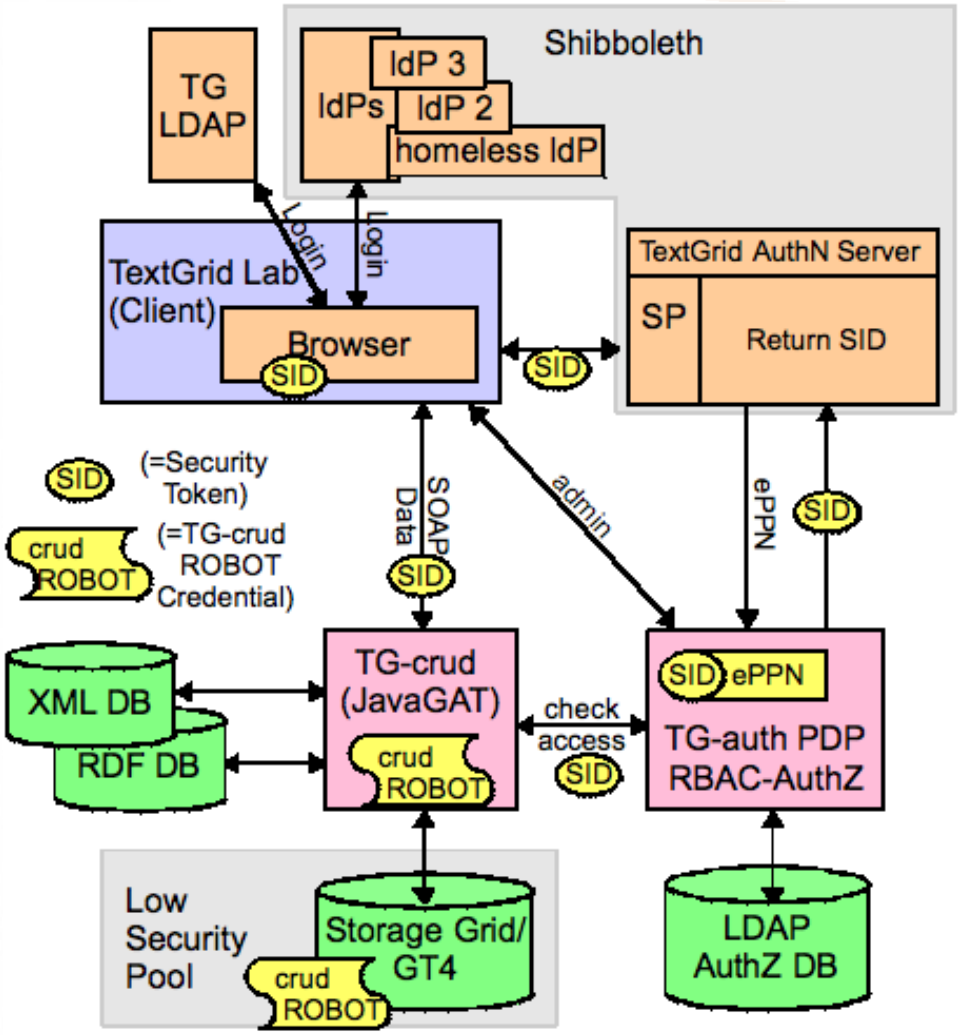
DAASI International
Directory Applications for Advanced Security and Information Management

TextGrid-Szenarien für Einbindung eines externen PDP (OpenRBAC)

- Kontaktierung des PDP über TG-Crud (unabhängig von Globus-Grid-Middleware)
  - E-Mail-Verifikation (TextGrid LDAP) / Shibboleth-Authentifizierung
  - TG-crud authentifiziert sich im Grid über ROBOT-Zertifikat
- Zusätzl. Mapping der PDP-Policy auf POSIX ACLs
  - Shibboleth-Authentifizierung und SLC (Short Lived Credential)
  - Userrechte werden auf Dateiebene durchgesetzt
  - ROBOT-Zertifikat wird nur noch für anonyme Zugriffe verwendet
- Zusätzl. direkte Kontaktierung des PDP durch Globus (XACML-Callout)
  - Shibboleth-Authentifizierung und SLC (Short Lived Credential)
  - Userrechte werden auf Globus-Ebene durchgesetzt

DAASI International
Directory Applications for Advanced Security and Information Management

Szenario 1



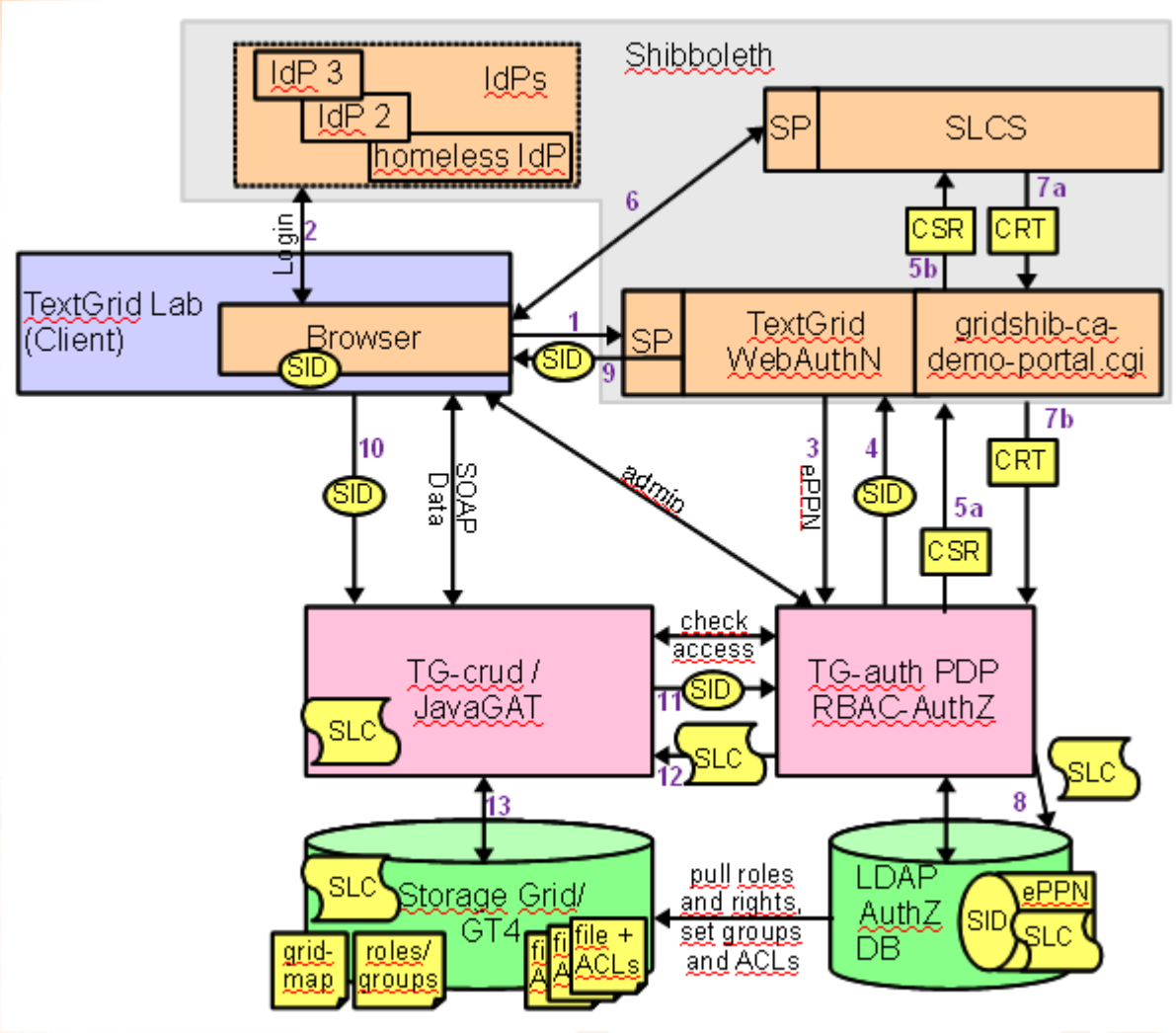
DAASI International
Directory Applications for Advanced Security and Information Management

Szenario 2

- Nutzer authentifiziert sich über Shibboleth (TG-auth*)
- TG-auth* generiert Schlüssel und SLC-Zertifikat-Request (private key ist nur im RAM, nicht auf Festplate)
- Portal leitet Request an SLCS weiter, SLCS signiert SLC
- signiertes Zertifikat wird im TG-auth* abgelegt
- TG-crud holt sich SLC von TG-auth*
- Daten werden im Heimat-Verzeichnis des SLC-Nutzers abgelegt
- Zugriffsrechte werden von TG-auth* (RBAC) in das Dateisystem der Grid-Middleware gemappt (ACLs)

DAASI International
Directory Applications for Advanced Security and Information Management

Szenario 2 Architektur



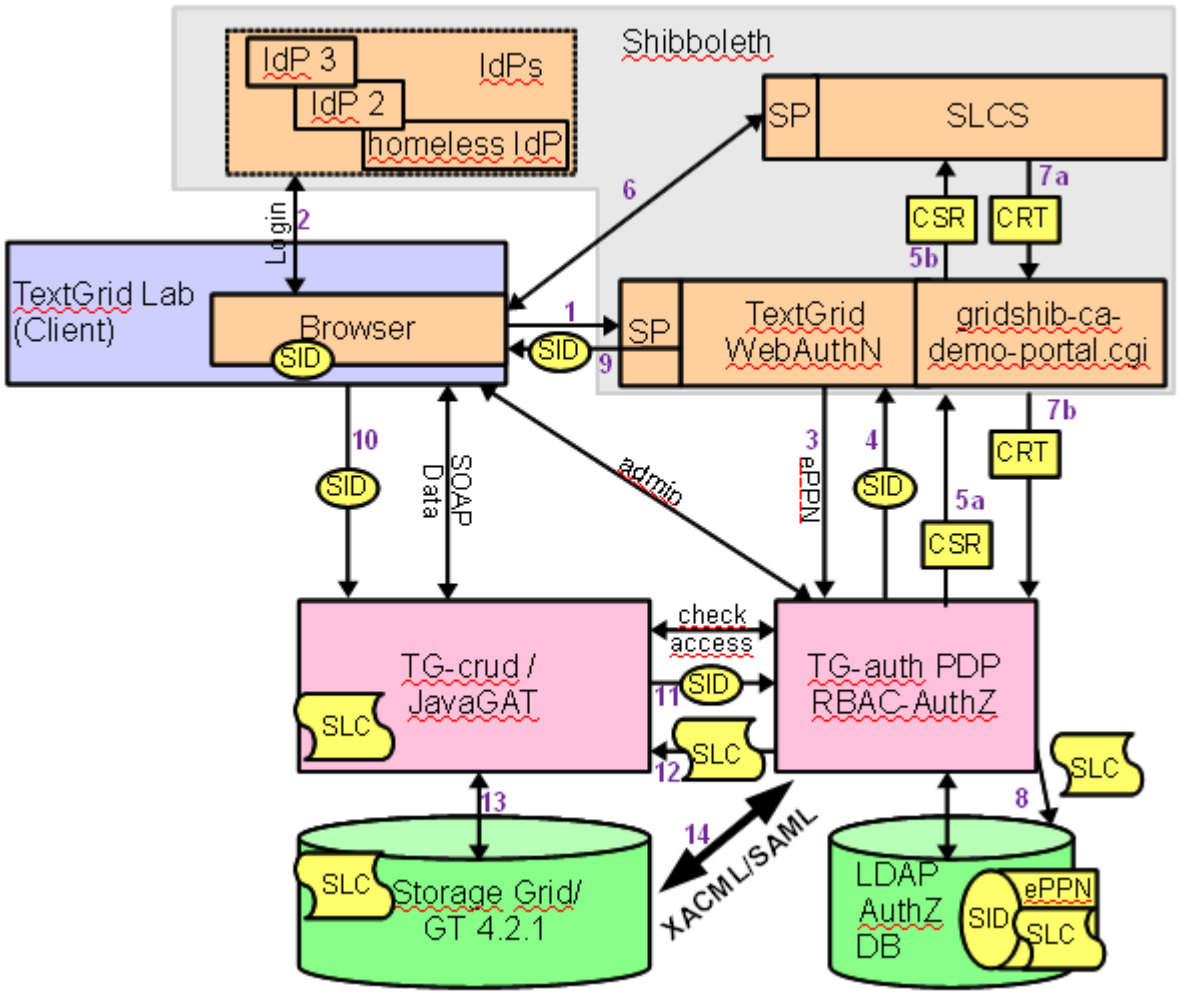
DAASI International
Directory Applications
for Advanced Security
and Information Management

Szenario 3 - XACML-SAML

- SLCs wie in Szenario 2
- Daten werden ebenso im Heimat-Verzeichnis des SLC-Nutzers abgelegt
- Zugriffsrechte werden von Globus als PEP direkt bei TG-auth* als PDP über das XACML-SAML-Request-Response-Protokoll abgefragt, wobei Folgendes mitgegeben wird:
  - Subject-DN aus dem Zertifikat
  - Name der Ressource (Datei)
  - gewünschte Operation für Zugriff
- Auf Dateiebene erhält Globus über Gruppenzugehörigkeit alle Rechte auf die von ihm verwalteten Ressourcen

DAASI International
Directory Applications for Advanced Security and Information Management

Szenario 3 - XACML-SAML



DAASI International
Directory Applications
for Advanced Security
and Information Management

Bewertung

- In allen Szenarien hat sich OpenRBAC als flexibel einsetzbar bewährt
- Vorteile Szenario 2 und 3:
  - Ressource weiß, welcher Nutzer was macht
  - Gridknoten müssen nicht einzeln konfiguriert werden sondern können zentral über einen PDP verwaltet werden
- Nachteil Szenario 2:
  - Replikation der ACLs benötigt Root-Rechte
  - Nicht alle Policyregeln des PDP sind in ACLs abbildbar
  - z.B. keine delete permission
- Vorteile Szenario 3:
  - Es findet keinerlei zeitverzögerte Replikation statt (weder Gridmapfile noch ACLs)
  - Alle Regeln sind prinzipiell abbildbar, der gesamte Funktionsumfang von RBAC kann genutzt werden

DAASI International
Directory Applications for Advanced Security and Information Management

Diskussion

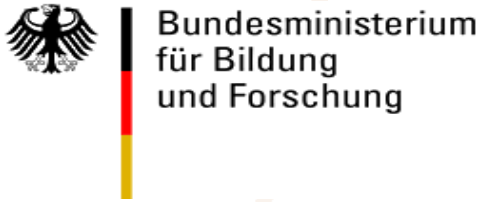
- OpenRBAC ist reine Backend-Infrastruktur
- Graphen können mit einem gewissen Aufwand nachrüstbar
- Es ist für OpenRBAC transparent:
  - ob Rollen zentral oder dezentral vergeben werden
    - Die OpenRBAC-Methoden sind von OpenRBAC geschützte Ressourcen
    - „Zentraler PDP“ ist zentral für die Anwendungen, nicht Verwaltung
  - ob Vieraugenprinzip eingeführt wird, muss im Frontend implementiert werden
- Auch eXtreme Role Engineering ist im Frontend

DAASI International
Directory Applications for Advanced Security and Information Management

Vielen Dank für Ihre Aufmerksamkeit!

- Noch Fragen?

- DAASI International GmbH
  - www.daasi.de
  - Info@daasi.de



DAASI International
Directory Applications for Advanced Security and Information Management