5. Aufgabenserie

Entwurfsbeschreibung OLAT

Inhaltsverzeichnis

1. Allgemeines
2. Produktübersicht
   2.1 Rollen
   2.2 Rechte der jeweiligen Rollen
3. Grundsätzliche Struktur- und Entwurfsprinzipien für das Gesamtsystem
   3.1 Policy-Konzept
   3.2 Einordnung des Rechtesystems in das Gesamtsystem
4. Grundsätzliche Struktur- und Entwurfsprinzipien einzelner Pakete
   1. Vorstellung wichtiger OLAT-Pakete
      1. org.olat.core.id
      2. org.olat.basesecurity
      3. org.olat.group
      4. Das Zusammenspiel von BusinessGroups und SecurityGroups
      5. org.olat.group.right
   2. Zugriff auf das Rechte- und Rollenmanagement von Erweiterungspunkten
   3. Erweiterung des Rechtesystems zur Realisierung einer elektronischen Studentenakte (ESA)

1. Allgemeines

Das Rechtesystem ist eine Kernfunktionalität von OLAT. Es ist wichtig, dass Benutzer nur die Funktionalitäten in Anspruch nehmen dürfen, für die sie Rechte haben. Die Rechte in OLAT sind additiv, d. h. alle Rechte der Gruppen, in denen ein Benutzer ist, werden zusammengezählt und ergeben die gesamten Rechte eines Benutzers. Die Rechte sind auch positiv, so dass sie eine *Erlaubnis* beschreiben, *etwas zu tun*.

2. Produktübersicht

2.1 Rollen

Es wird in OLAT zwischen den Rollen Gast, Benutzer, Autor, Gruppenverwalter, Benutzerverwalter und Administrator unterschieden.

2.2 Rechte der jeweiligen Rollen

Gast:

Ein Gast hat nur Zugriff auf Lernressourcen, die explizit für Gäste freigegeben sind.

Benutzer:

Ein Benutzer hat erweiterte Rechte gegenüber einem Gast. Er besitzt ein eigenes Profil mit individueller Oberfläche und persönlichen Daten. Weiterhin kann er sich an Kursen anmelden und eigene Arbeitsgruppen erstellen.

Autor:

Als Erweiterung zum Benutzer kann der Autor außerdem Lernressourcen erstellen und darin wiederum Lerngruppen und Lernressourcen erstellen.

Gruppenverwalter:

Der Gruppenverwalter kann zusätzlich zu den Benutzerrechten Gruppenkontexte, kursübergreifende Lerngruppen und kursübergreifende Rechtegruppen verwalten.

Benutzerverwalter:

Der Benutzerverwalter kann als spezieller Benutzer zusätzlich Benutzer erstellen, löschen, importieren und ihnen Rollen zuweisen.

Administrator:

Der Administrator vereinigt alle Rechte der vorher genannten Rollen. Zusätzlich kann er noch administrative Funktionen ausführen.

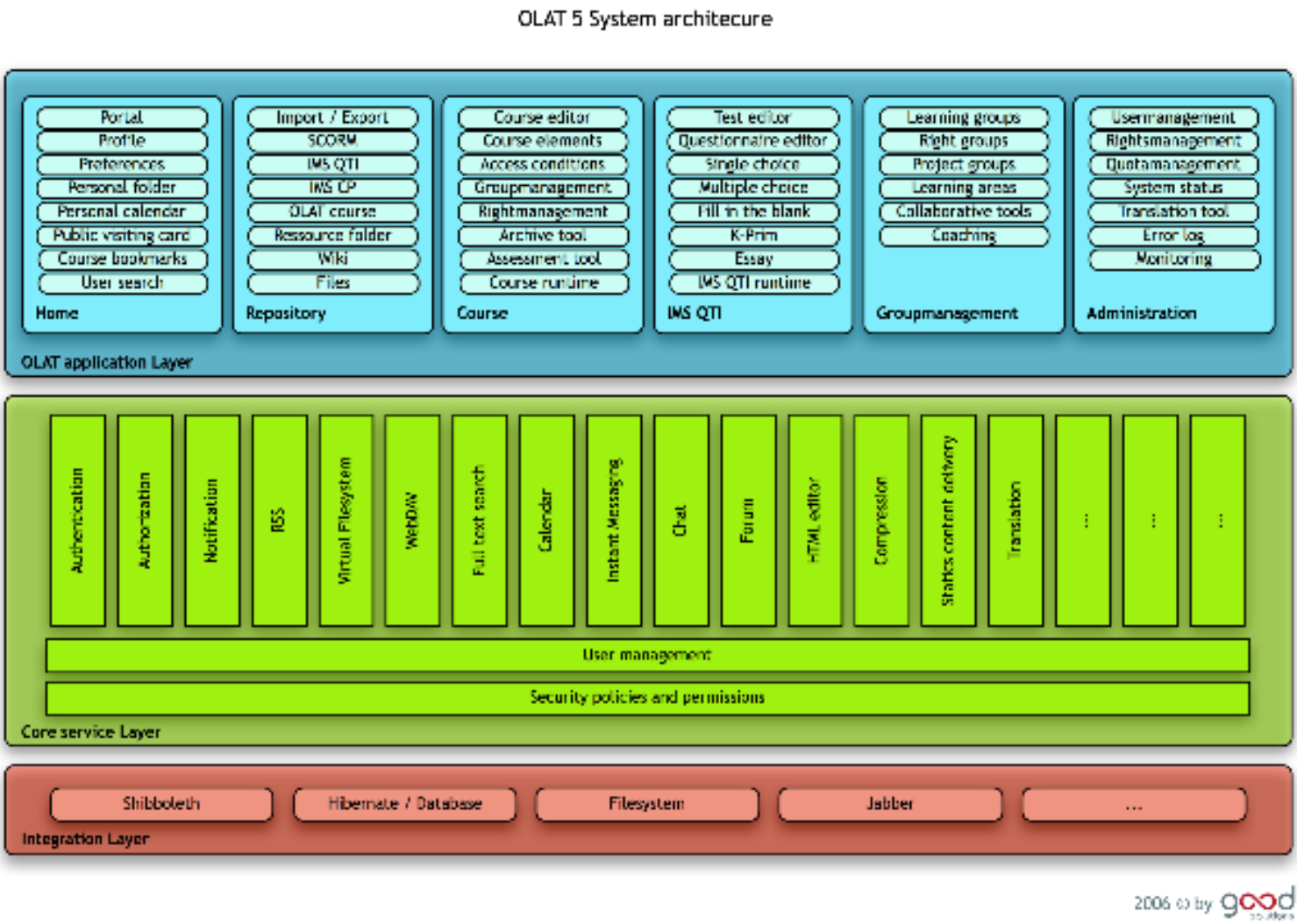
3. Grundsätzliche Struktur- und Entwurfsprinzipien für das Gesamtsystem

3.1 Policy-Konzept

Das sogenannte Policy-Konzept ist für das Rechtesystem in OLAT sehr wichtig. Eine Policy ist die Kombination von einem Recht, einer Gruppe und einer Ressource. Befindet sich ein Nutzer in einer SecurityGroup, so hat er Zugang zu den Ressourcen und die Rechte, die die Policy dieser SecurityGroup zuordnet. Dies könnten etwa Lese- oder Schreibrechte sein. Die Security Policy beschreibt, welche Gruppe welche Rechte auf welche Ressourcen hat.

3.2 Einordnung des Rechtesystems in das Gesamtsystem

Die Systemarchitektur von OLAT ist modular. Man kann in einer funktionalen Sichtweise zwischen den drei Schichten Applikationsschicht, Serviceschicht und Integrationsschicht unterscheiden. Wichtige Funktionen der Applikationsschicht sind das Home, die Gruppenverwaltung und die Systemadministration. Die Serviceschicht hat die Aufgabe, der Applikationsschicht wesentliche Dienste und Komponenten zur Verfügung zu stellen. Diese beinhalten das Rechtesystem von OLAT. So sind Authentifikation, Autorisierung und das Sicherheitsframework in dieser Schicht verankert. Weiterhin gibt es noch die Integrationsschicht, in der grundlegende Funktionalitäten wie das Dateisystem und der Datenbankzugriff verwaltet werden. Die folgende Grafik soll das Schichtenmodell verdeutlichen.

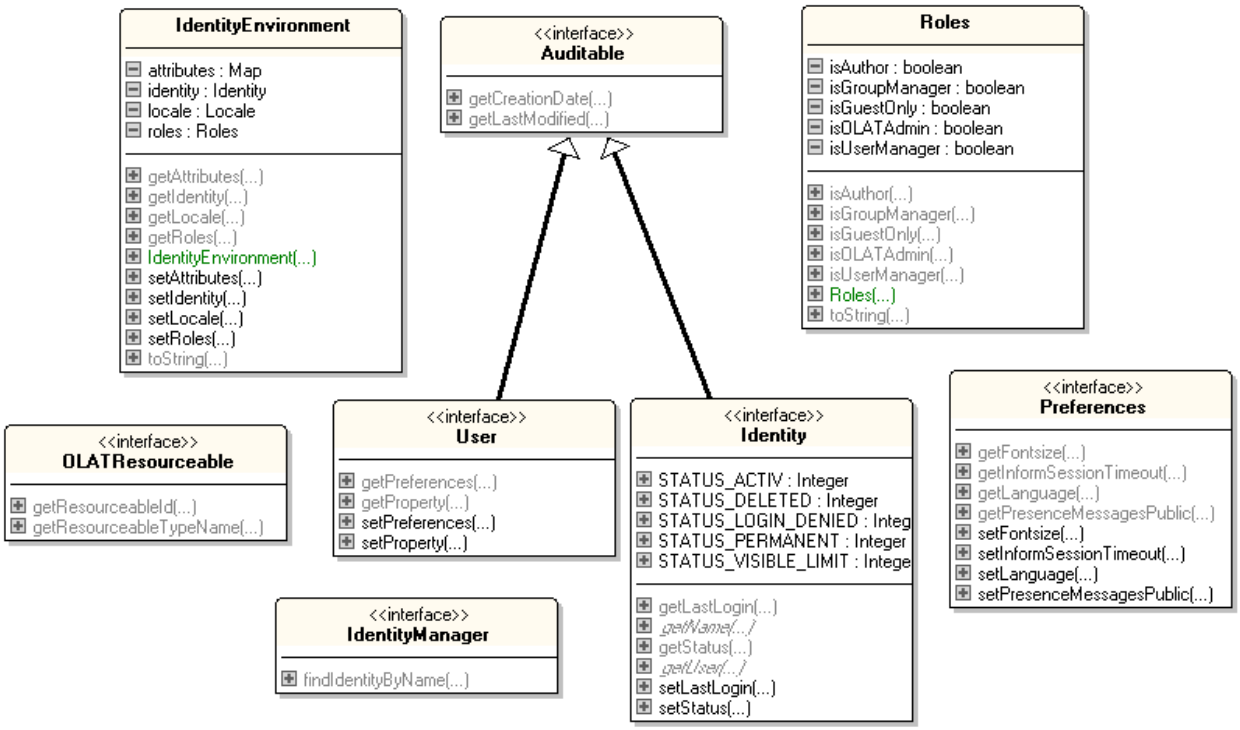


4. Grundsätzliche Struktur- und Entwurfsprinzipien einzelner Pakete

4.1. Vorstellung wichtiger OLAT-Pakete

4.1.1. org.olat.core.id

In diesem Paket befinden sich diverse Interfaces und Klassen, die dazu dienen, die Rollen und die damit verbundenen Rechte eines OLAT-Benutzers zu speichern und auszulesen. In der Klasse „IdentityEnvironment“ werden die Identität und die Rollenzugehörigkeit eines Benutzers verwaltet. Mit dem Interface „IdentityManager“ kann die Identität anhand des Namens eindeutig festgestellt werden. Das folgende UML-Schema zeigt den Paket-Aufbau:



4.1.2. org.olat.basesecurity

In diesem Paket befindet sich die Implementierung von SecurityGroups sowie der Policy, welche für das Rechtekonzept das Grundgerüst bildet. Eine Policy ist das Zusammenwirken einer SecurityGroup und Ressourcen, wobei Identitäten in der SecurityGroup aus der jeweiligen Policy der Zugriff auf die Ressourcen innerhalb derselben Policy gewährt wird.

4.1.3. org.olat.group

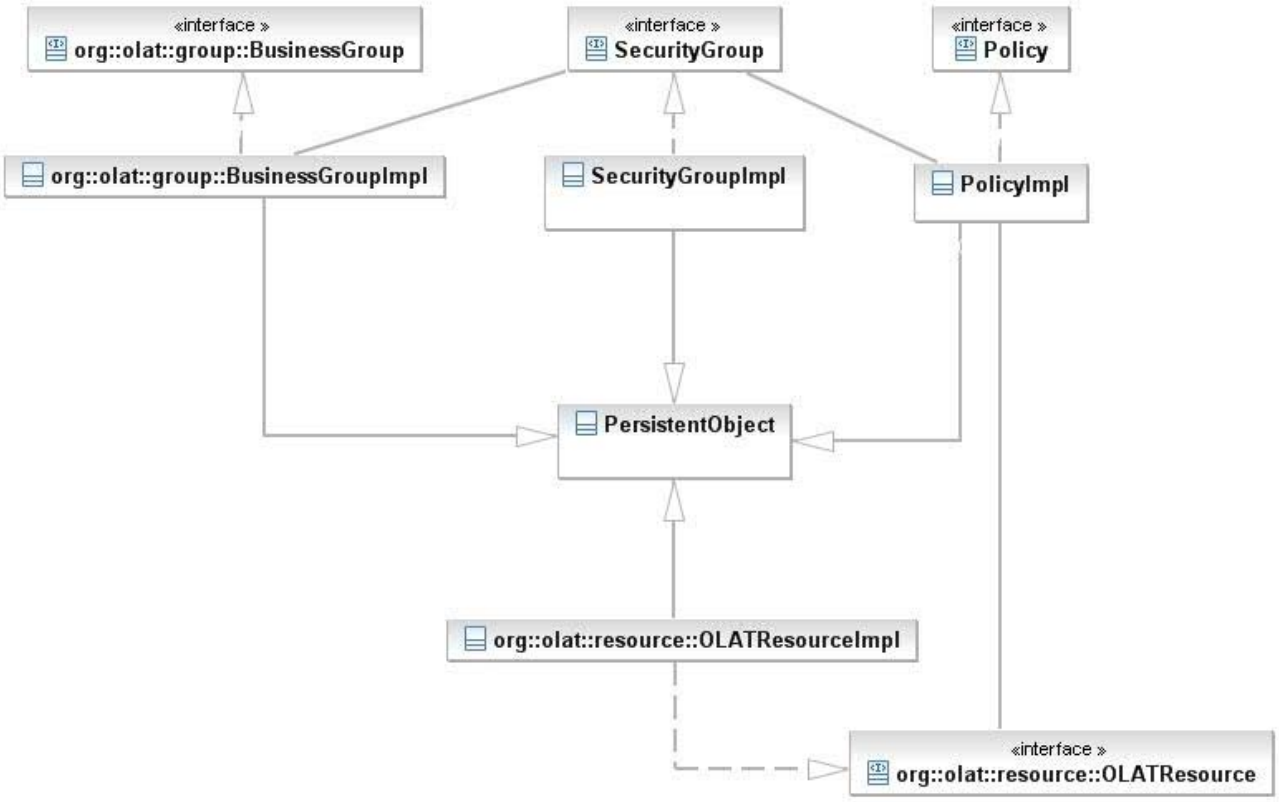
In diesem Paket befinden sich die Implementierung von BusinessGroups sowie die Verwaltung und die Rechteverwaltung der BusinessGroups.

(Siehe org.olat.group.right)

4.1.4. Das Zusammenspiel von BusinessGroups und SecurityGroups

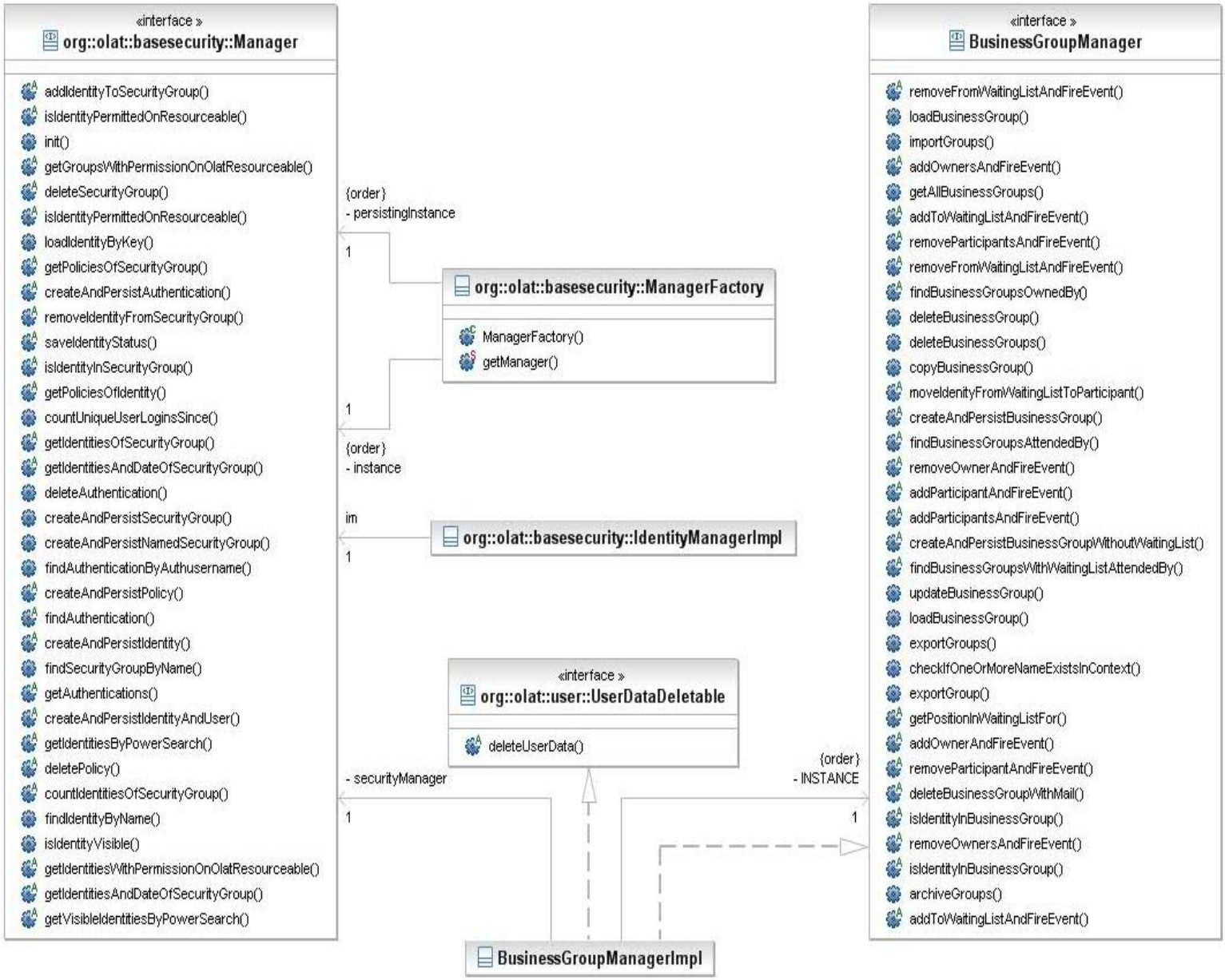
Auch in einer BusinessGroup wird auf das Konzept der SecurityGroup zurückgegriffen. So enthält die Implementierte BusinessGroup 3 Instanzen einer SecurityGroup.

Die Implementierten Klassen werden von PersistentObject abgeleitet, um die Persistenz der Daten zu bewahren. Dies zeigt das folgende Schema:



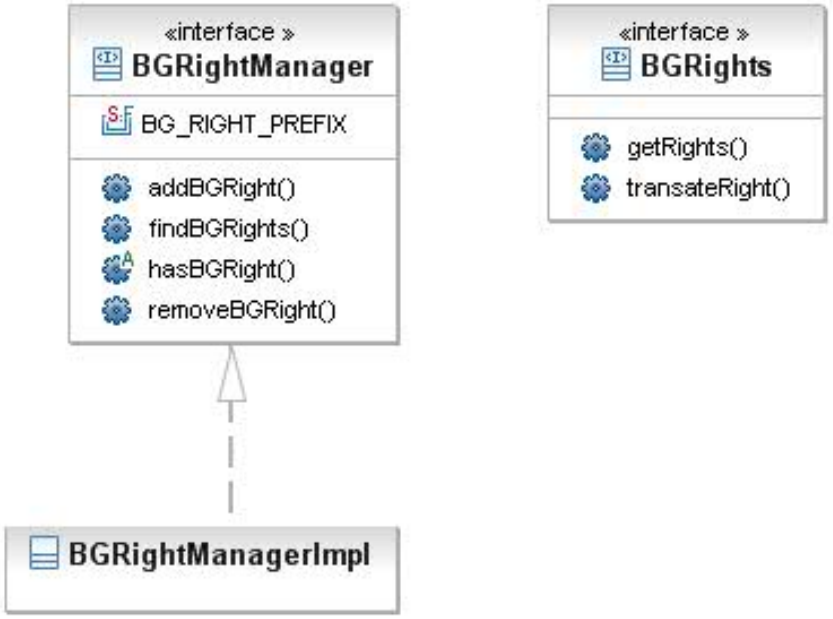
Über Manager werden die Verwaltung von SecurityGroups und BusinessGroups ermöglicht, sowie verschiedene Methoden zur Manipulation und Verwaltung bereitgestellt.

So kann z.B. geprüft werden, ob eine Identity in einer BusinessGroup oder SecurityGroup ist.



4. 1.5. org.olat.group.right

In diesem Paket werden die Rechte von BusinessGroups verwaltet. Man kann einer BusinessGroup BusinessGroup-Rechte zuordnen, verwehren oder diese auflisten.



4.2. Zugriff auf das Rechte- und Rollenmanagement von Erweiterungspunkten

In der Demo Extension wird nur an einer Stelle auf das Rechtesystem zugegriffen. Dieser Zugriff erfolgt bei dem Initialisieren der DemoExtension-Seitendefinition „DemoSiteDef“.

Hierzu wird folgender Befehl genutzt:

***if** (ureq.getUserSession().getRoles().isGuestOnly()) **return null**;*

Es ist zu erkennen, dass die Rolle, die der Nutzer besitzt, in dessen aktueller Session gespeichert wird. Mittels „.getRoles()“ werden die Rollen des Benutzers aus dieser herausgelesen.

In der DemoExtension wird gleichzeitig mittels „.isGuestOnly“ überprüft, ob es sich um einen Gast des Systems handelt. Ist dies der Fall wird der „null“-Wert zurückgegeben, wodurch keine Instanz der DemoExtension-Seite erzeugt und somit dem Gast diese Seite nicht angezeigt wird.

Somit kann man zusammenfassend sagen, dass von einem Erweiterungspunkt über die mit dem UserRequest verbundene Session auf das Rechte- und Rollenmanagement zugegriffen werden kann.

4.3. Erweiterung des Rechtesystems zur Realisierung einer elektronischen Studentenakte (ESA)

Man könnte nur dem Benutzer in der bestehenden Rolle des „*Administrators*“ gestatten, die ESA zu editieren. Jedoch kann es möglich und wahrscheinlich auch gewollt sein, Benutzern ohne Administrator-Rechte das Editieren der ESA zu gewähren.

Weiterhin gibt es noch die Rolle des „*UserManager*“, welcher bereits die Möglichkeit besitzt, Benutzer des Systems zu editieren. Aber auch hier besteht dasselbe Problem, da nicht jeder Nutzer, der die ESA bearbeiten kann, auch gleichzeitig andere Einstellungen des Nutzers (wie z.B. sein Passwort) ändern können sollte.

Somit wird ersichtlich, dass, um eine ESA als Erweiterung in das OLAT-System zu integrieren, eine neue Rolle geschaffen werden muss (hier *ESAEditor* genannt).

Da nun die Rollen „*Administrator*“ und „*UserManager*“ effektiv Boolean-Werte sind, muss man bei der Erweiterung des Rechtesystems darauf achten, dass bei Nutzern der „*Administrator*“- und „*UserManager*“-Rolle auch der Wert für „*isESAEditor*“ gesetzt wird. Denn zum Einen muss der Administrator Vollzugriff auf das System besitzen und zum Anderen sollte ein Benutzer, der bereits Benutzer editieren kann, auch die Möglichkeit haben, die ESA zu editieren.

Mit der bereits bestehenden Rolle „*user*“ kann man einem Benutzer einen lesenden Zugriff auf die ESA gewährleisten, indem man Nutzern dieser Rolle den Inhalt der ESA ausgibt. Sollte ein Nutzer weiterhin die Rolle des „*ESAEditor*“ besitzen, kann man nun weiter Buttons und Links zum Editieren der ESA ausgeben.