APEX: Zentrale Rechteverwaltung aller Applikationen leicht gemacht

Dr. Alex Kohn
Pharma Research and Early Development Informatics
Roche Innovation Center Penzberg
Roche Diagnostics GmbH

Schlüsselworte

APEX, Security, Authorisierung, Authentifizierung, Single-Sign-On

Einleitung

APEX wird in mehr als 100 Applikationen im Bereich Pharma Research & Early Development (pRED) von Roche eingesetzt. Damit Standardaufgaben bei der Verwaltung existierender und der Entwicklung neuer Applikationen möglichst geringen Aufwand verursachen, haben wir die APEX - Plattform um neue Features erweitert. Von zentraler Bedeutung dabei ist das Rechte-Management. Für die gesamte APEX Instanz existiert hierzu eine zentrale Applikation zur Verwaltung von Benutzern, Gruppen und Rollen, welche benutzerfreundlich im APEX Application Builder verlinkt ist und zusätzlich in unserem Roche Application Template vorkonfiguriert ist. Dadurch wird NTLM-basiertes Single-Sign-On und Rechte-Management basierend auf Active Directory Gruppen für den Entwickler zum Kinderspiel.

Apex Security Management Applikation

Die Grundlage für die Verwaltung der Rechte ist die Apex Security Management Applikation (ASMA) – der zentrale Dreh- und Angelpunkt für die Definition von Sicherheitsregeln aller Applikationen einer APEX Instanz (vgl. Abb. 1). Die in ASMA definierten Regeln werden mit Hilfe des PL/SQL Pakets „APEX_AUTH“ von den Applikationen konsumiert.

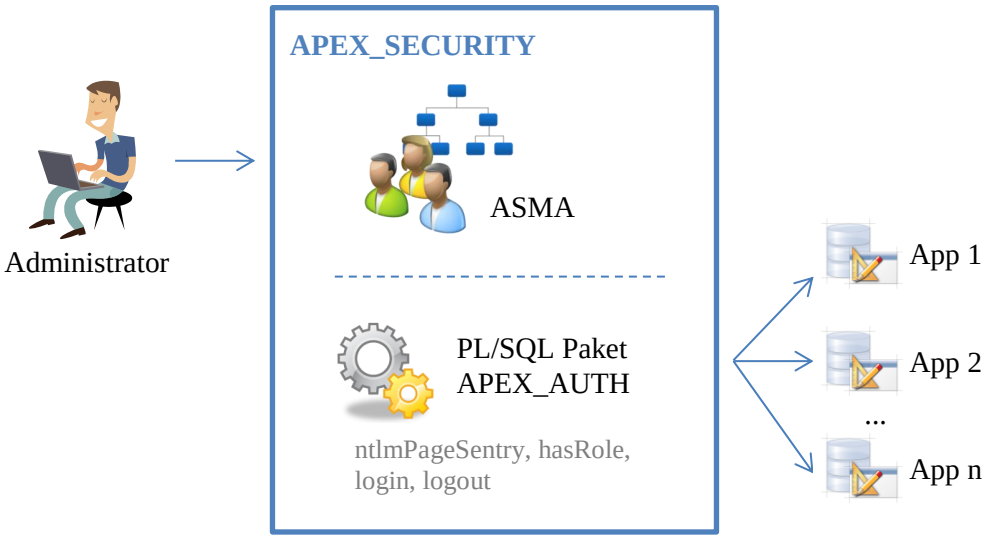

Abb. 1: Komponenten des Security Konzepts

Im Folgenden werden die Konzepte von ASMA beschrieben und anschließend die Funktionalität von APEX_AUTH.

Die Berechtigungen einer Applikation werden durch *Benutzer*, *Gruppen* und *Rollen* geregelt. Die Bezeichner der Benutzer und der Gruppen leiten sich mittels LDAP vom Active Directory ab. Benutzer und Gruppen können in einer frei definierbaren Rollenhierarchie organisiert werden. Die Rollen werden in applikationsspezifischen Authorisierungs Schemas verwendet, was eine granulare Zugriffssteuerung ermöglicht. Zwei häufig verwendete Rollen – *Contributor* und *Administrator* – sind in ASMA bereits vordefiniert. Die Rolle Contributor ist für den Zugriff auf Elemente konzipiert die Schreibrechte benötigen (z.B. das Erstellen neuer Datensätze). Die Rolle Administrator gewährt Zugriff auf alle Funktionen einer Applikation. Außerdem genießt diese Rolle einen besonderen Stellenwert in ASMA: Ein Administrator darf Benutzer, Gruppen und die Rollenhierarchie verwalten. Folglich können die Anwender im Self-Service die Rechte verwalten ohne den Umweg über die IT-Abteilung gehen zu müssen. Unsere Erfahrung hat gezeigt, dass diese Möglichkeit der Prozessvereinfachung sehr gut von den Endanwendern angenommen wird. Neben dem *Applikations-Administrator* unterscheidet ASMA zwei weitere Arten von Administratoren für die Rechteverwaltung, welche v.a. für die Entwickler relevant sind:

- *Instanz*-Administratoren können Workspace-Administratoren sowie die Rechte aller Applikationen verwalten.
- *Workspace*-Administratoren können die Rechte aller Applikationen innerhalb ihres Workspaces verwalten.

Eine typische Rechte-Konfiguration ist in Abb. 2 dargestellt. Im Beispiel sind der drei Benutzer für den Lesezugriff berechtigt: Erika, Robert und Hans. Schreibzugriff haben nur die Benutzer Erika und Robert. Erika erbt die Schreibrechte der Rolle Contributor aufgrund der Rollenhierarchie. Der Benutzer Hans verfügt über reine Leserechte. Da Erika Administratorin ist, darf sie außerdem im Self-Service die Rechte der Applikation verwalten.

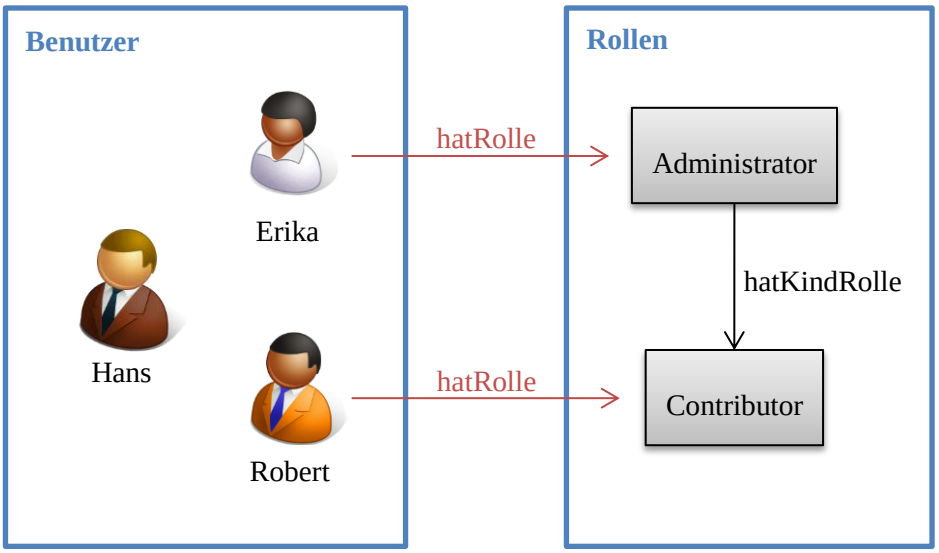

Abb. 2: Beispiel einer Konfiguration in ASMA

Üblicherweise will man, wie in dem Beispiel gezeigt, den Zugriff auf die genannten Benutzer und Gruppen limitieren. Es gibt aber auch Fälle in denen Teile einer Applikation für alle Mitglieder einer Organisation zugänglich sein sollen, während andere Teile nur von bestimmten Rollen geöffnet

werden dürfen. Diesen Sonderfall, d.h. die Deaktivierung der Authorisierung auf Applikationsebene, lässt sich mittels einer Konfigurationsoption in ASMA abdecken.

Alle in ASMA vorgenommenen Änderungen werden im APEX_SECURITY Schema materialisiert (s. Abb. 3, gelbe Markierung). In dem selben Schema befinden sich auch die Konfigurationseinstellungen für die Verbindung mit dem LDAP Server (s. Abb. 3, graue Markierung). Diese sind für das Paket APEX_AUTH relevant.

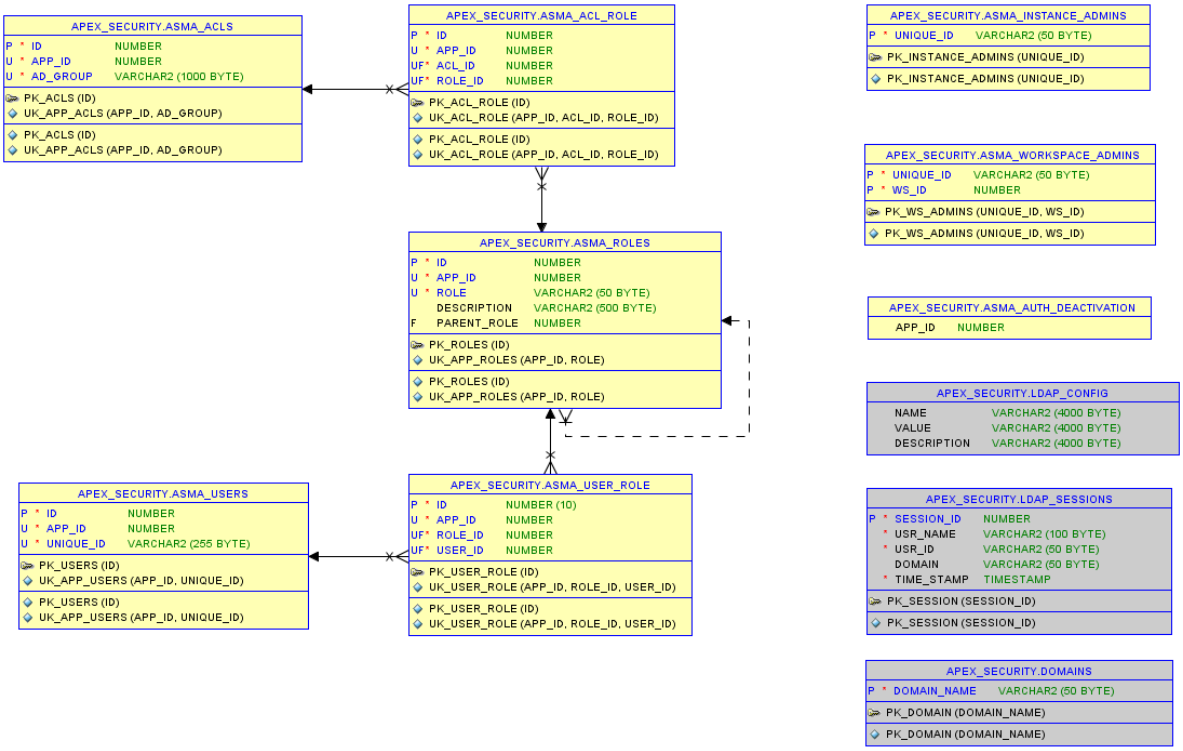

Abb. 3: Datenbankschema von APEX_SECURITY

PL/SQL Paket APEX_AUTH

Das Paket APEX_AUTH stellt die Schnittstelle zwischen ASMA und den APEX Applikationen und bietet dazu vier wesentliche Methoden an:

login

Diese Funktion hat zwei Betriebsarten: Single-Sign-On (SSO) und manueller Login. SSO liegt vor, wenn die REMOTE_USER Variable gesetzt ist (vgl. nächstes Kapitel). Andernfalls überprüft die Funktion die Kombination aus Benutzernamen und Passwort gegen den LDAP Server.

logout

Diese Prozedur wird aufgerufen nachdem der Benutzer auf den logout Link geklickt hat. Momentan hat diese Prozedur keine zusätzliche Aufgabe. Offene Sessions werden automatisch nach einem Time-Out gelöscht.

ntlmPageSentry

Diese Funktion wird von Apex bei jedem Request aufgerufen, beispielsweise bevor eine Seite dargestellt / prozessiert wird oder vor einem Ajax Request. Es überprüft ob der aktuelle Benutzer

authentifiziert und authorisiert ist, das betreffende Element (Seite, Knopf, etc.) der Applikation zu öffnen. Die Authorisierung überprüft hierzu, ob der Benutzername für die aktuelle Applikation in der Tabelle ASMA_USERS hinterlegt ist oder ob der Benutzer Mitglied einer in der Tabelle ASMA_ACL eingetragenen Active Directory Gruppe ist.

hasRole
Diese Funktion überprüft ob der Benutzer / die Gruppe Mitglied der angegebenen Rolle ist. Diese Methode wird verwendet, um fein-granulare Authorisierungs Schemas in einer APEX Applikation zu erstellen.

Single-Sign-On

Ein weiterer Schritt zur Verbesserung des Benutzererlebnisses ist Single-Sign-On (SSO): Der Benutzer öffnet eine APEX Applikation und wird ohne manuelle Zwischenschritte automatisch authentifiziert.

Im folgenden wird skizziert, wie NTLM-basiertes SSO in APEX realisiert werden kann. NTLM ist ein Authentifizierungs Protokoll, welches in Windows Server Arbeitsgruppen eingesetzt wird und z.B. vom Internet Explorer oder FireFox WebBrowser implementiert wird.

Damit NTLM mit APEX verwendet werden kann, bedarf es noch der Unterstützung des Protokolls auf dem Applikationsserver (Tomcat, WebLogic, etc.). Zwei Optionen haben sich dafür in der Praxis als robust erwiesen. Für Windows Server ist Waffle (http://dblock.github.io/waffle/) eine sehr gute Wahl. Unter Linux kann Waffle nicht verwendet werden da es Abhängigkeiten zu Windows Bibliotheken hat. Hier gibt es mit Jespa (http://www.ioplex.com/) eine sehr gute Alternative, die jedoch kostenpflichtig ist.

Beim Deployment der Oracle Rest Data Service müssen die Waffle bzw. Jespa Bibliotheken eingebunden werden und der Deployment Descriptor (context.xml und web.xml) muss entsprechend der Anleitung erweitert werden. Wenn das Deployment abgeschlossen ist, ist auf dem Applikationsserver ein Filter aktiv, der bei jedem Aufruf den Benutzer in die REMOTE_USER Variable schreibt. Diese Information kann anschließend von unserer Prozedur in APEX_AUTH verwendet werden um den Benutzer automatisch zu identifizieren und einzuloggen.

Template Applikation

Als nächstes wird gezeigt wie die Anbindung von ASMA und von APEX_AUTH in den Entwicklungsprozess integriert wird. Der Aufwand hierfür soll minimal gehalten werden. Deshalb definieren wir eine Template Applikation, von der alle neuen Applikationen ableiten. Die Anbindung an ASMA wird also nur einmalig im Template konfiguriert: Die Konsumenten müssen sich nicht um die Details kümmern. Außerdem kann man durch diesen Mechanismus Änderungen am Master-Template durch einen Klick an alle Konsumente weitergeben.

Im Template wird eine neue Authentifizierungsfunktionen namens NTLM erstellt und als Standard definiert, welche die Methoden von APEX_AUTH aufruft (Abb. 4). Außerdem werden zwei vordefinierte Authorisierungsschemas *isAdministrator* und *isContributor* erstellt, welche mit den entsprechenden Standardrollen in ASMA verknüpft sind (Abb. 5).

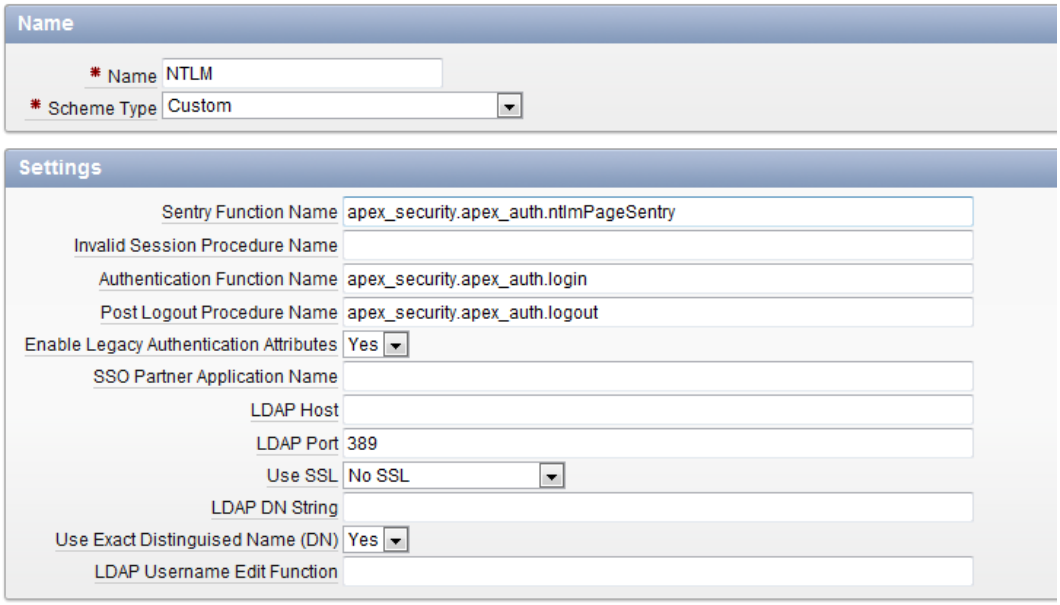

Abb. 4: NTLM Authentifizierung

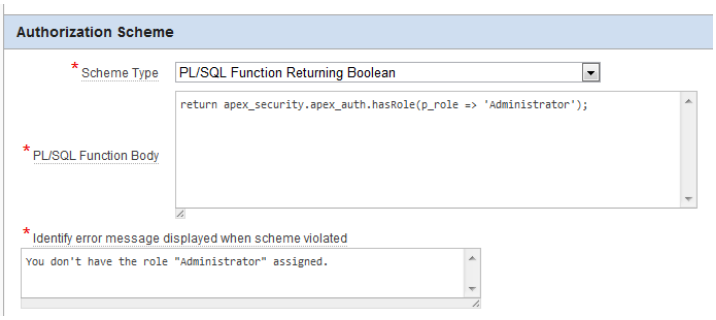

Abb. 5: Authorisierungsschema isAdministrator

Integration von ASMA in den APEX Application Builder

Um den Zugriff auf ASMA möglichst einfach zu gestalten wird die Toolbar des Application Builders um ein zusätzliches Icon „Manage Access Control“ erweitert. Die Erweiterung der Toolbar wird durchgeführt wie es Roel Hartman in seinem Vortrag XFILES auf der DOAG 2011 beschrieben hat (https://www.doag.org/termine/termine.php?tid=423128&stream=2844828). Im INTERNAL Workspace angemeldet, öffnet man „Manage Instance“ und anschließend „Define System Message“. Dort wird mittels JavaScript die existierende Toolbar erweitert. Anschließend wird das neue Icon in der Toolbar des Application Builders angezeigt. Das Icon verlinkt auf das Rechte-Management in ASMA für die aktuell ausgewählte Applikation (Abb. 6).

Abb. 6: Integration von ASMA in den Application Builder

Fazit

Mit der Apex Security Management Applikation, dem Paket APEX_AUTH, der SSO Erweiterung des Applikationsservers, dem Applikations Template und der Integration von ASMA in die Toolbar des Application Builders ist eine sowohl für die Endanwender als auch für die Entwickler benutzerfreundliche Lösung für die zentrale Verwaltung der Zugangsregeln geschaffen worden.

Kontaktadresse:

Dr. Alex Kohn

Pharma Research and Early Development Informatics

Roche Innovation Center Penzberg

Roche Diagnostics GmbH

Nonnenwald 2

D-82377 Penzberg

Telefon: +49 (0) 8856-6019138

E-Mail alex.kohn@roche.com

Internet: www.roche.de