Oracle Real Application Security (RAS) in APEX5

Pavel Glebov, Nikolaus Sperat
FRT Consulting GmbH
Graz

Schlüsselworte

Real Application Security (RAS), Oracle Application Express 5 (APEX), Oracle Database Enterprise Edition 12g

Einleitung

Oracle Application Express ermöglicht erweiternde Zugangskontrollmethoden, die auf der Benutzeroberfläche definiert werden können. Man kann zum Beispiel einfach unterschiedliche APEX-Componenten ein- oder ausblenden. Aber eine Daten-Zutrittskontrolle ist mit dieser Methode nicht abgedeckt. Wie könnte man zum Beispiel das Gehalt vor fremden Abteilungen verstecken? Diese Frage und mehr Informationen über Oracle Real Application Security (RAS) auf Datenniveau in APEX 5 werden im Vortrag beantwortet.

Zugriffsbeschränkungen in APEX

Zugriffsbeschränkungen in APEX sind ein großes Thema, die zwei wichtigsten Begriffe sind dafür sind Authentifizierung und Autorisierung. Als Authentifizierung bezeichnet man den Vorgang in dem der User seine Identität bestätigt.

Durch die Autorisierung-Prozesse definiert der Entwickler, welche Rechte ein User bekommt, welche Felder er sehen kann und welche Funktionalität ihm zu Verfügung steht. Diese Autorisierungsfunktion steht nicht nur für einzelne Eingabe- und Steuerungselemente wie Eingabefelder, Buttons und Menüeinträge zu Verfügung sondern damit können ganze Region oder Seiten gegen Zugriff geschützt werden.

Schwierigkeiten mit Datenzutrittskontrolle

In APEX fehlt es aber an einer eingebauten Datenzugriffkontrolle. Die Sicherheitsanforderungen benötigen oft eine Kontrolle auf sehr feiner Granularität – Zur Einschränkung welche Daten von welchem Benutzer betrachtet werden dürfen. Diese Beschränkungen können nicht nur bestimmte Tabellenzeilen betreffen sondern auch auf Spaltenebene eingeschränkt sein. Beispielsweise dürfen alle Mitarbeiter allgemeint zugängliche Information (Public Information) wie Vor- und Nachname oder E-Mail Adresse ihrer Kollegen aus der gleichen Abteilung sehen aber nur der Vorgesetzte und der Mitarbeiter selber darf sein eigenes Gehalt ansehen.

Gleichzeitig sollen die Maßnahmen nur einen geringen Einfluss auf die Datenbank nehmen. Speziell für solche Anforderungen bietet Oracle die Real Application Security (RAS) – Technologie an. Diese steht in der Enterprise Edition 12c zu Verfügung und ist auch gut in APEX 5 integriert.

Eine Rolle von RAS in typische WEB-Anwendung

Die wichtigste Herausforderung, die diese Technologie löst, ist die Darstellung von Applikation Users in der Datenbank. In modernen 3-Tier Web-Architekturen verbindet der Applikationsserver (zum

Beispiel Tomcat mit ORDS) mit der Datenbank als ein bestimmter Datenbankuser – im Fall von APEX heißt dieser APEX_PUBLIC_USER. Alle End-User der Web-Anwendung, die ganz unterschiedliche Rechte besitzen können, verbinden sich über diesen Datenbankuser. Das ist eine große potenzielle Sicherheitslücke, die durch manuell angelegte Views oder PL/SQL Prozeduren von den Entwicklern abgesichert werden soll. Oracle RAS kann diese Herausforderung leicht lösen. RAS stellt den Anwendungsuser und seine Rechte auf Datenbankebene dar und beschränkt damit den Zugang zu Datenbankobjekten und Daten.

Kurze Überblick von RAS–Grammatik und RASADM

In RAS verwendet man eigene spezielle Begriffe, die zum Verständnis des Prozessablaufs der Zugriffsbeschränkung nötig sind. RASADM bietet eine gute Möglichkeit sich einen Überblick über die RAS-Grammatik zu verschaffen. RASADM ist eine von Oracle entwickelte APEX-Applikation speziell für die Administration von RAS.

Die Applikation ist auf verschiedene Tabs aufgeteilt, die jeweils eine eigene Ebene in der Zugriffsverwaltung darstellen.

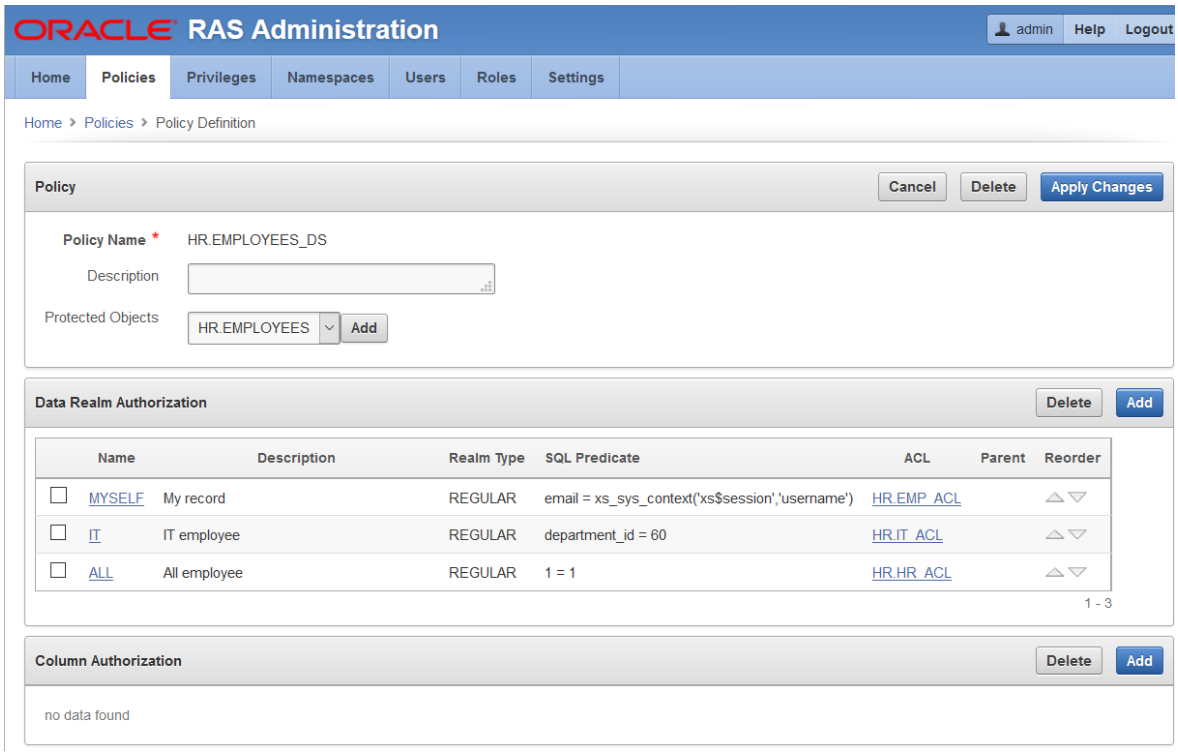

Abb. 1 RAS Administration Policies Tab

Data Realm ist eine Definition von Regeln, die den Zugang zum Datenbankobjekt- Zeilen, Spalten oder in einer anderen Weise beschränkt. Diese Einschränkung wird durch SQL-Prädikate definiert.

Policy Ist eine Sammlung von Data Realms. In RASADM kann man einfach zugehörige Data Realms einer bestimmten Tabelle zuweisen.

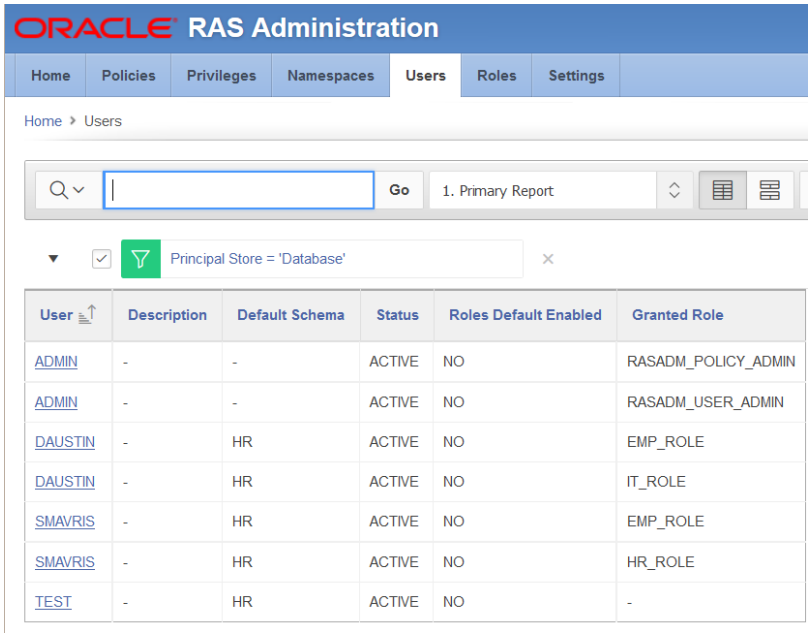

Abb. 2 RAS Administration Users Tab

Application User ist ein normaler Web-Applikationsbenutzer, wie Arnold S. oder Nicolaus K.. Diese werden durch den Benutzernamen definiert und können daher kein Datenbankschema oder irgendwelche Datenbankobjekte und Ressourcen besitzen. Der Entwickler sollte aber dem User ein Default-Datenbankschema zuweisen.

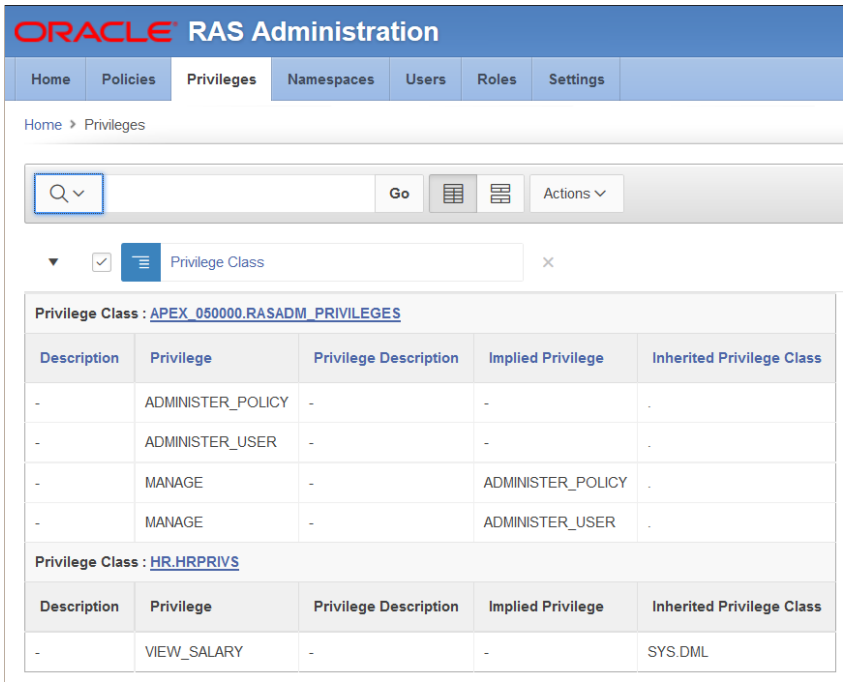

Abb. 3 RAS Administration Privileges Tab

Application Privelege Dieser Begriff hat in RS Kontext fast gleicher Sinn wie Rechte, und bieten der Zugang oder erlauben bestimmte Tätigkeit auf einzelne Business Objekt. Solche Business Objekt kann auf Datenbankniveau gleich wie Applikation Niveau Zugang mehrere Objekte haben.

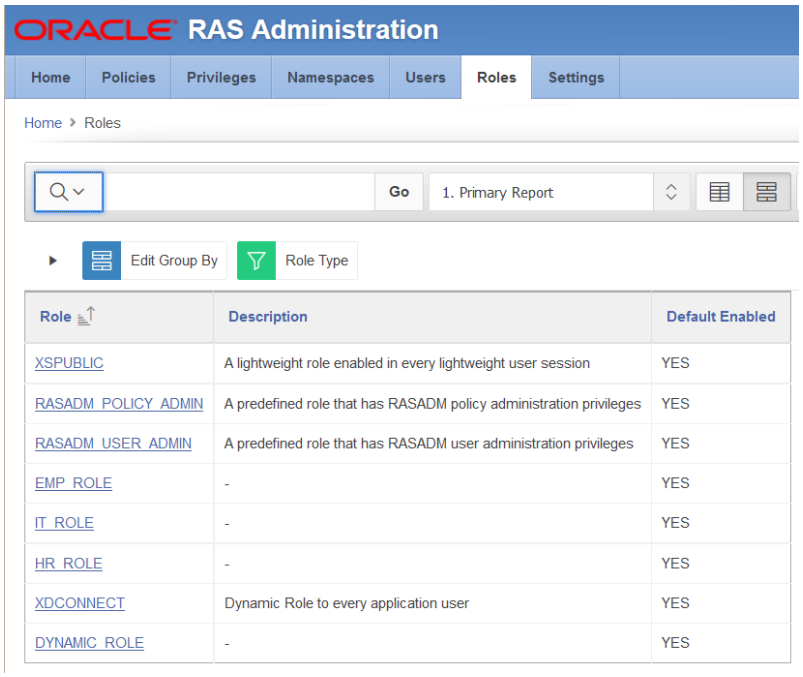

Abb. 4 RAS Administration Roles Tab

Application Role ist eine Gruppe von Application Priveleges

ACL ist die Liste an Privilegien die für einen User oder Applikation Role gilt. Ausnahme sind Column Constraints – diese sind direkt einem Application Privilege zugewiesen.

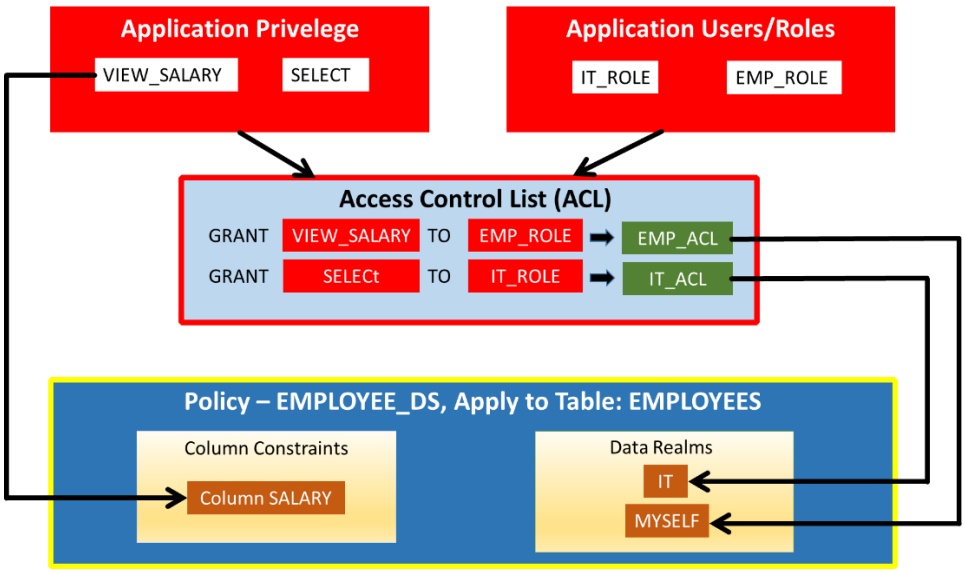

Abb. 5 RAS Policy Components

Applikationsdesign: einen Auswahl zwischen External oder Internal RAS Users.

Data Realms, Policies und ACL werden in der Datenbank gespeichert, weil diese mit der Hilfe des RAS PL\SQL Packages (oder RASADM) angelegt und verwaltet werden. Gleiches gilt für Application Roles und Application Privileges.

Aber die Situation für Anwendungsbenutzer ist nicht so einfach. Selbstverständlich gibt es die in RAS eingebaute Möglichkeit, den Benutzer im RAS anzulegen. Aber die RAS-Option ist nur in der Enterprise Edition kostenfrei vorhanden und Betriebe die eine solche teure Lösung kaufen, brauchen fast immer ein komplexeres User Management.

Es gibt aber eine Möglichkeit, sogenannte External User mit RAS zu verwenden. Die Information über solche User wird nicht in RAS gespeichert. Die Datenzutrittskontrolle wird für diese Benutzer durch Zuweisung von dynamischen Applikationsrollen beim Login gesteuert. Diese Möglichkeit ist am besten für Betriebe anwendbar, die ein zentrales Usermanagement System wie Active Directory oder Single Sign On haben.

Deswegen ist wichtigste Frage, wo die für die Anmeldung benötigten Information gespeichert werden? Wo werden Benutzerrollen angelegt? Wer und wie sollen diese Rollen zugewiesen werden?

Am besten ist eine Antwort auf diese Frage in der Design-Phase zu bekommen. Dafür hilft folgenden Zusammenfassung mit Vor- und Nachteilen.

RAS Stammbenutzer

- RAS wird als zentrale Ablage von Benutzerdaten verwendet, alle Applikation-Benutzernamen werden in der Datenbank gespeichert.
- Mit XS_PRINCIPAL.SET_PROFILE kann man dem User unterschiedliche Profile zuweisen: zum Beispiel Kennwortkomplexität, Begrenzung der Nutzung von Datenbankresourcen.
- Benutzer- und Rollenverwaltung mit Hilfe des XS_PRINCIPAL Packages oder direkt in RASADM
- Direct Connection mit der Datenbank sollte für Applikationsuser vorhanden sein (Default Datenbankschema beim Anlegen der Benutzer definieren) – besonders bequem, wenn End-User aus vielen unterschiedlichen Applikationen einen Zugriff zur Datenbank haben sollen
- Im APEX die Database Account Authentifizierungsmethode benutzen
- Für APEX Autorisation Schemas kann man ORA_CHECK_ACL oder ORA_CHECK_AUTORIZATION Type verwenden.
- Mehrere Verwaltungsmöglichkeiten sind nur RAS Stammbenutzer möglich: DBMS_XS_SESSIONS ENABLE_ROLE/DISABLE_ROLE und anderen Prozeduren sind nur für RAS User erlaubt.

External Users:

- Die Applikations-Benutzernamen und Kennwörter außerhalb von RAS gespeichert.
- Jede beliebige Authentifizierungsmethode verwendbar: (individuell gestaltet auf eigene "USERS" Tabelle, LDAP, Single Sign-On, APEX Workspace Users, oder HTTP Header Variable etc).
- Alle Rollen sollten mit APEX_AUTHORIZATION.ENABLE_DYNAMIC_GROUPS in Post-Authentication Prozedur hinzugefügt werden.

- Für APEX Autorisation Schemas kann man ORA_CHECK_ACL, ORA_CHECK_PRIVELEGE oder "Is In Group" autorisation type verwenden.

Aktivierung von RAS-Feature im APEX.

Um das RAS Feature zu nutzen, muss man zuerst das Feature für eine bestimmte Instanz einschalten. Dafür muss man sich in den INTERNAL Workspace einloggen und dort unter „Manage Instance“ => „Security“ im „Real Application Security“ Abschnitt „Allow Real Application Security“ auf „Yes“ stellen.

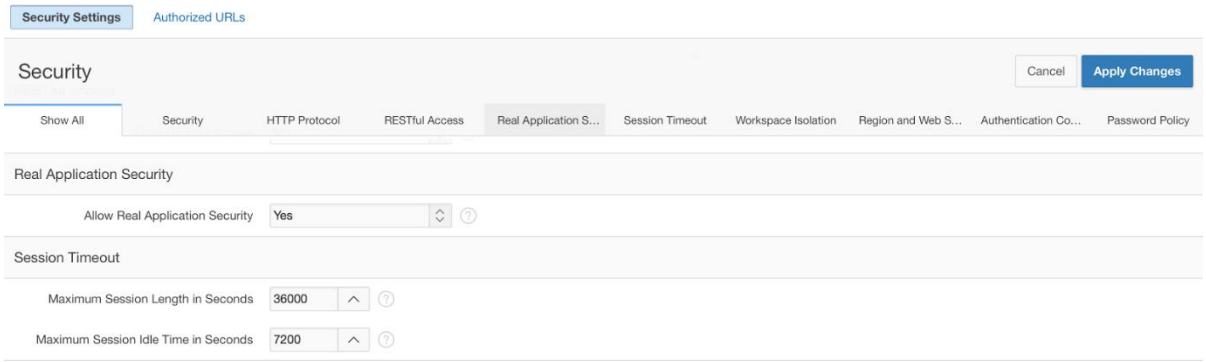

Abb. 6 Security Tab on Instance Management

Dann wird in den User Workspaces bei „Authentification Scheme“ in den Security Settings der Tab „Real Application Security“ angezeigt. Dort kann man die wichtigsten Sicherheitsparameter für RAS einstellen.

- **RAS Mode** – hier gibt es Auswahl zwischen „Internal Users“ – das sind RAS Stammbenutzer und „External Users“.
- **Dynamic Roles** – hier wählt man die Liste an dynamische Rollen aus, die für alle Benutzer beim Logon automatisch zugewiesen werden sollen.
  Hier sieht man, dass in meiner Demo alle Benutzer standardmäßig „“ dynamische Rollen bekommen.
- **Namespaces** – Die bei der Applikation verwendeten Namespaces. Dieser Begriff ist im Application Context benutzbar.
- Wenn man sich für die Design Variante mit externen Benutzer entschieden hat, sollte man zusätzlich die Login Funktion anlegen. Hier wird die Zugehörigkeit eines Benutzers zu einer bestimmten Nutzergruppe abgefragt. Diese Liste wird dann mit APEX_AUTHORIZATION.ENABLE_DYNAMIC_GROUPS der RAS Engine übergeben.

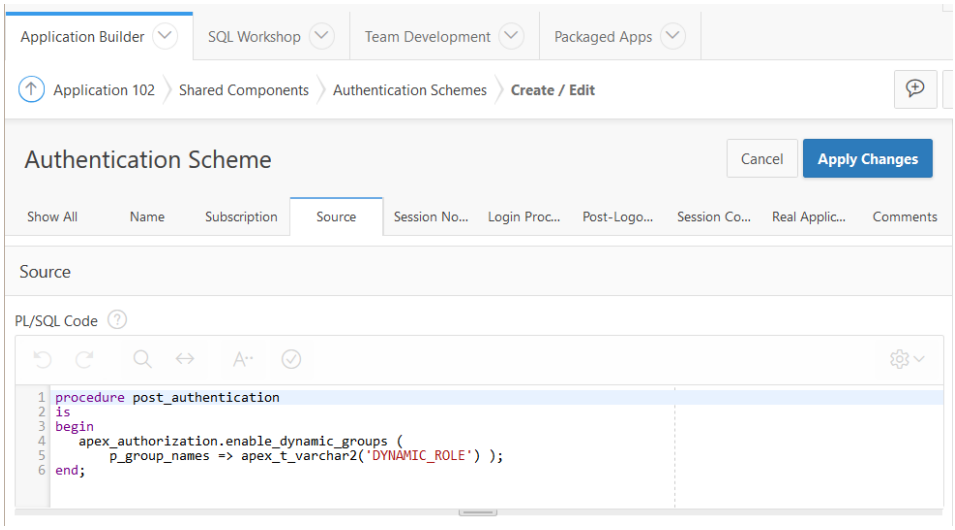

Abb. 7 Benutzerdefinierte Post-Authentication Funktion

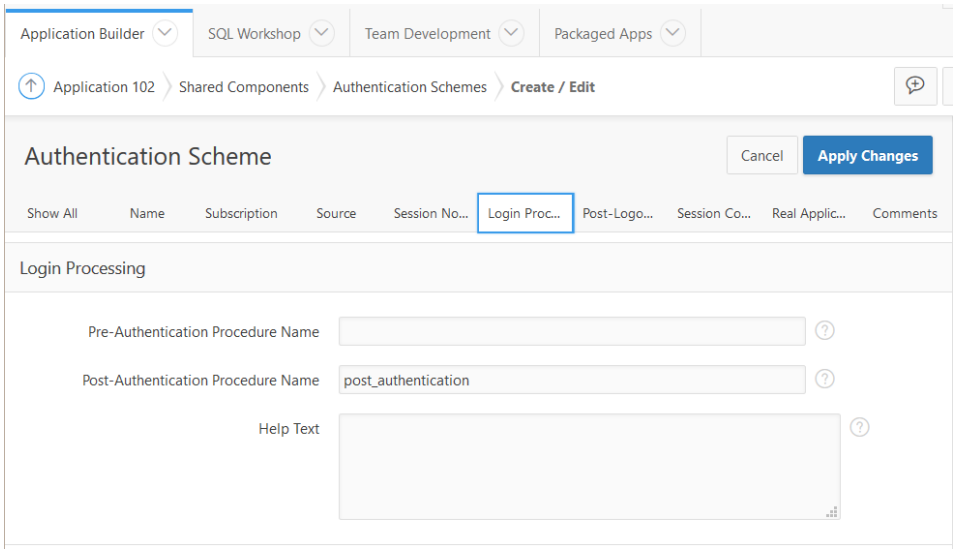

Abb. 8 Einen Aufruf von Post-Authentication Funktion

Wenn alle Schritte korrekt durchgeführt wurden, bekommt jeder Benutzer einen persönlich eingeschränkten Datenzugriff.

Am Schluss möchte ich die wichtigsten Vorteile von RAS für mich als APEX – Entwickler zusammenfassen:

- Diese Technologie braucht meistens weniger Datenbank-Ressourcen als Custom Views, die oft zum gleichen Zweck verwendet werden.
- Man kann komplexe Zugriffregeln anlegen oder anpassen ohne das es Auswirkungen auf das Datenbankschema hat. Das bedeutet: Es kommt zu keinen Invalid Objects und Rekompilation.
- RAS Security – kann man relativ einfach in andere Systeme integrieren, die Zugriff auf die Datenbank brauchen (zum Beispiel Reporting System).

- Es gibt eine fertige Management Web-Applikation von ORACLE, RASADM.

Kontaktadresse:
Pavel Glebov
FRT Consulting GmbH
Liebenauer Hauptstrasse 2-6
A-8041 Graz

Telefon: +43 316 71 12 12
Fax: +43 316 71 12 12 - 99
E-Mail pavel.glebov@frt.at, nikolaus.sperat@frt.at
Internet: www.frt.at