Einsatz von Java EE Security für APEX mit Oracle REST Data Services

Rico Haupt
Robotron Datenbank-Software GmbH
Dresden

Schlüsselworte

Oracle Fusion Middleware 11g, Oracle REST Data Services, ORDS, Oracle Application Express, APEX, JEE, Single Sign-on, JEE, Tomcat, WebLogic

Einleitung

Die Durchsetzung von Sicherheitsrichtlinien für IT-Systeme in Unternehmen und Organisationen hat weiterhin große Bedeutung, um Datenschutz und Compliance sicherzustellen. Ein integraler Bestandteil ist dabei das Identity- und Access-Management. Dabei geht es um die effective Verwaltung von Benutzern und die Durchführung einer Zugriffskontrolle mit dem Ziel, dass nur berechtigte Personen Zugriff auf Anwendungen und Daten haben. Ziel und auch Schwierigkeit ist es dabei, die vorhandenen Anwendungen im Unternehmen bzw. in der Organisation in ein einheitliches Identity- und Access-Management zu integrieren.

Der Vortrag behandelt Möglichkeiten, Oracle Application Express Anwendungen in eine einheitliche Zugriffskontrolle zu integrieren, die den Sicherheitsanforderungen eines Unternehmens bzw. einer Organisation entspricht. Exemplarisch erfolgt die Umsetzung mit ORDS in den Application Servern Tomcat und WebLogic Server.

Grundlagen und Begriffsklärung

Oracle REST Data Services (ORDS) ermöglicht die Bereitstellung von REST Web Services. In Version 2.0 wurden die REST Services in Zusammenhang mit Oracle Application Express (APEX bereitgestellt und entwickelt. In Version 3.0 können die REST Services auch unabhängig von APEX in der Oracle Database installiert werden. Für den Zugriff mit HTTP wird ORDS als JEE Web-Anwendung in einen Java Application Server bereitgestellt. Unterstützt werden der Oracle WebLogic Server, Glassfish und Tomcat.

Oracle Application Express (APEX) ist eine Entwicklungsumgebung von Web-basierten Anwendungen innerhalb der Oracle Database. Als HTTP-Listener können das Embedded PL/SQL Gateway der Oracle Datenbank, der Oracle HTTP Server mit mod_plsql sowie Oracle REST Data Services verwendet werden.

REST (Representational State Transfer) definiert Architekturprinzipien für das Design von Web Services. REST soll dabei eine einfache Alternative zu SOAP-basierten Web Services darstellen. SOAP, das auf XML basiert wird durch den Zugriff auf Ressourcen mit den bekannten HTTP Methoden (GET, POST, PUT, DELETE,…) ersetzt. Die Zugriffe erfolgen dabei ebenfalls statuslos – jede Aktion ist also unabhängig von einer Anderen zu betrachten. Zur Representation der Daten werden dabei üblicherweise die Standards JSON, HTML und CSV verwendet.

Konzept

Das Konzept beruht auf der Nutzung der JEE Security Features der Java Application Server. ORDS wird als Web Archiv bereitgestellt und besitzt wie jede JEE Deployment Deskriptoren. In den Deploymentdeskriptoren können Festlegungen zur Authentifzierungs und Zugriffskontrolle getroffen werden, die dann vom Application Server umgesetzt werden.
Gemeinsam haben die vorgestellten Lösungen, dass sie als Nutzer- bzw. Gruppenbasis LDAP-Verzeichnisdienste, konkret das Active Directory, integrieren.

Für unterschiedliche Mechanismen der Benutzerauthentifizierung und Benutzeridentifizierung bieten Tomcat und WebLogic Server verschiedenen Möglichkeiten.

Beim WebLogic Server lassen sich zwei Arten unterscheiden: Authentication Provider und Identity Asserter.
Authentication Provider ermöglichen es andere Systeme, wie Verzeichnisdienste oder Datenbanken, als Quellen für Benutzer und Gruppen einzubinden und darauf basierend die Authentifizierung durchzuführen.
Identity Asserter wiederum dienen dazu aus übergebenen Security Token die Benutzeridentität zu ermitteln. Die Authentifizierung wurde dabei von dritter Seite bereits durchgeführt.

Beim Tomcat Server erfolgt die Konfiguration durch sogenannte Realms. Neben der internen Nutzerdatenbank können auch LDAP-Verzeichnisdienste und Datenbanksysteme angebunden werden.

In diesem Vortrag wird das Active Directory integriert und die Authentifzierung auf Basis von SPNEGO/Kerberos konfiguriert.

Umsetzung

1. Integration der Nutzer und Gruppenbasis

Tomcat Server:

Editieren der TOMCAT_HOME/conf/server.xml

```
<Realm className="org.apache.catalina.realm.JNDIRealm"
 debug="99"
 connectionURL="ldap://active-directory-hostname:389"
 authentication="simple"
 referrals="follow"
 connectionName="cn=ldapuser,cn=users,dc=test,dc=de"
 connectionPassword="passwort"
 userSearch="(samaccountname={0})"
 userBase="cn=users,dc=test,dc=de"
 userSubtree="true"
 roleSearch="(uniquemember={0})"
 roleName="cn"
 roleSubtree="true"
 roleBase="cn=groups,dc=test,dc=de"/>
```

WebLogic Server:

Anlegen des Active Directory Authentication Provider

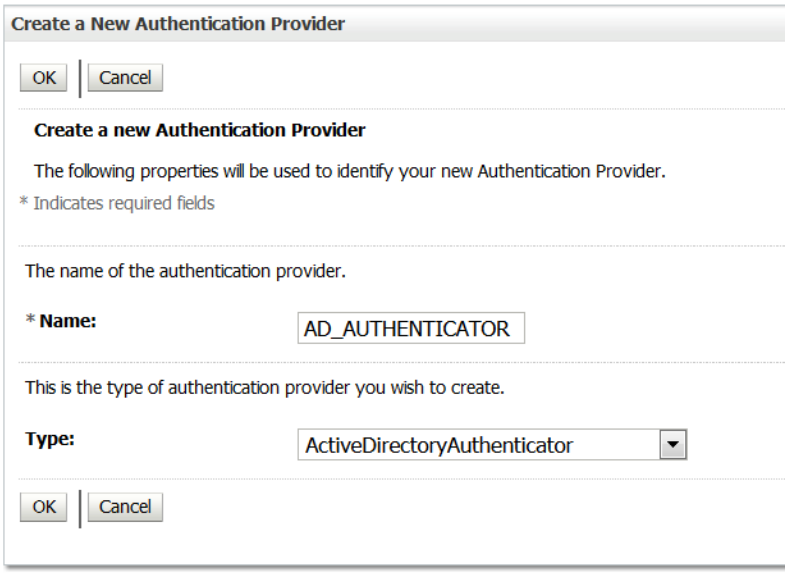

Abb. 1: Create a New Authentication Provider

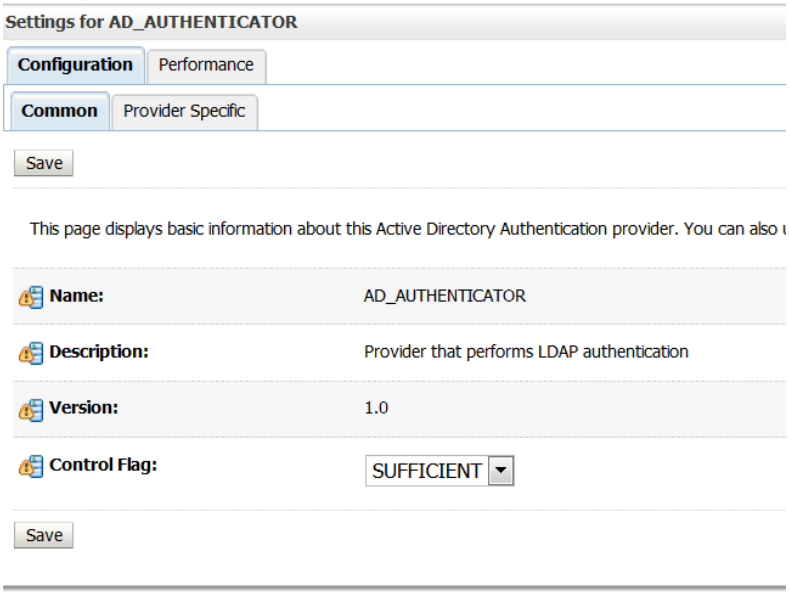

Abb. 2: Modify Common Authentication Provider Settings

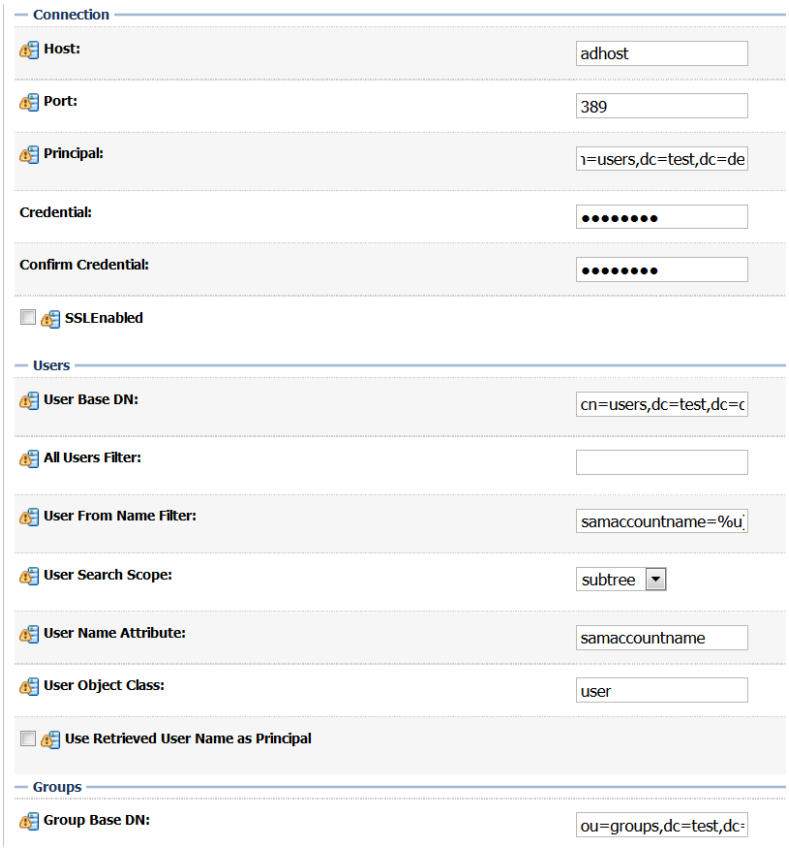

Abb. 3: Modify Provider Specific Authentication Provider Settings

Anlegen des Identity Asserter

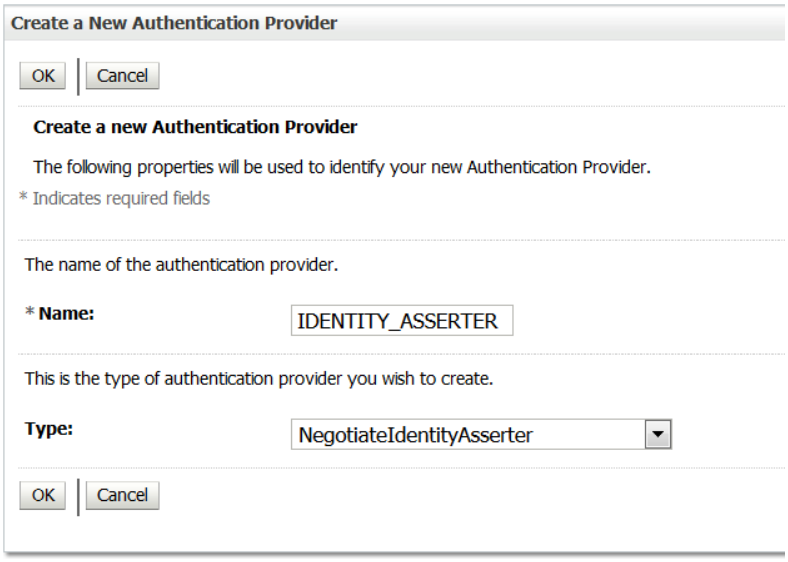

Abb. 4: Create a New Identity Asserter

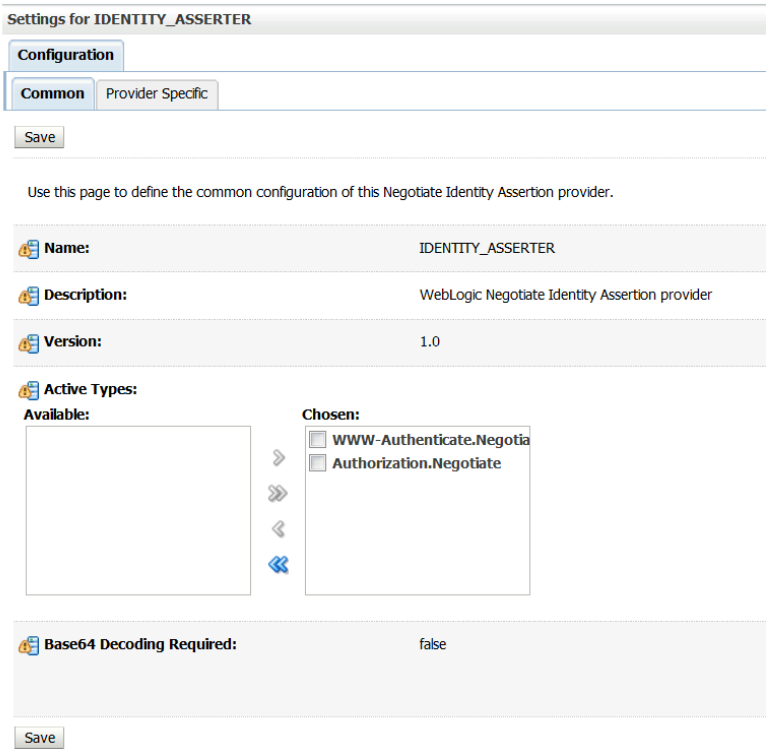

Abb. 5: Modify Common Identity Asserter Settings

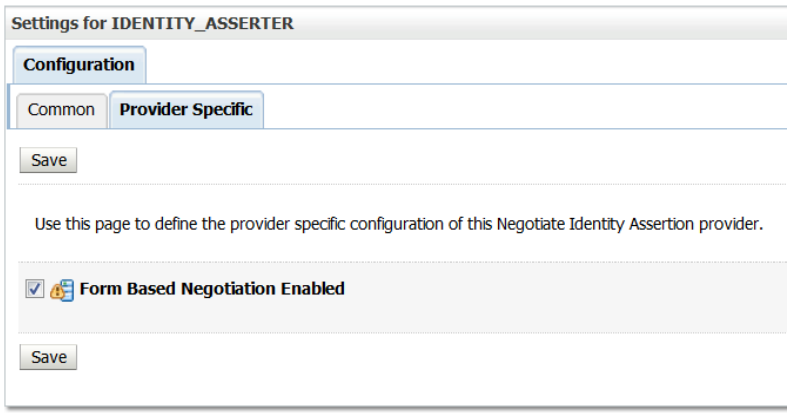

Abb. 6: Modify Provider Specific Identity Asserter Settings

Anschließend wird die richtige Abarbeitungsreihenfolge für die Provider festgelegt (Abb. 7). Die beiden standardmäßig eingerichteten „DefaultAuthenticator“ und „DefaultIdentityAsserter“ bleiben erhalten. Darüber werden die Benutzer für den Betrieb des Systems bereitgestellt, sodass auch ohne verfügbares Active Directory die Instanz gestartet und verwaltet werden kann.

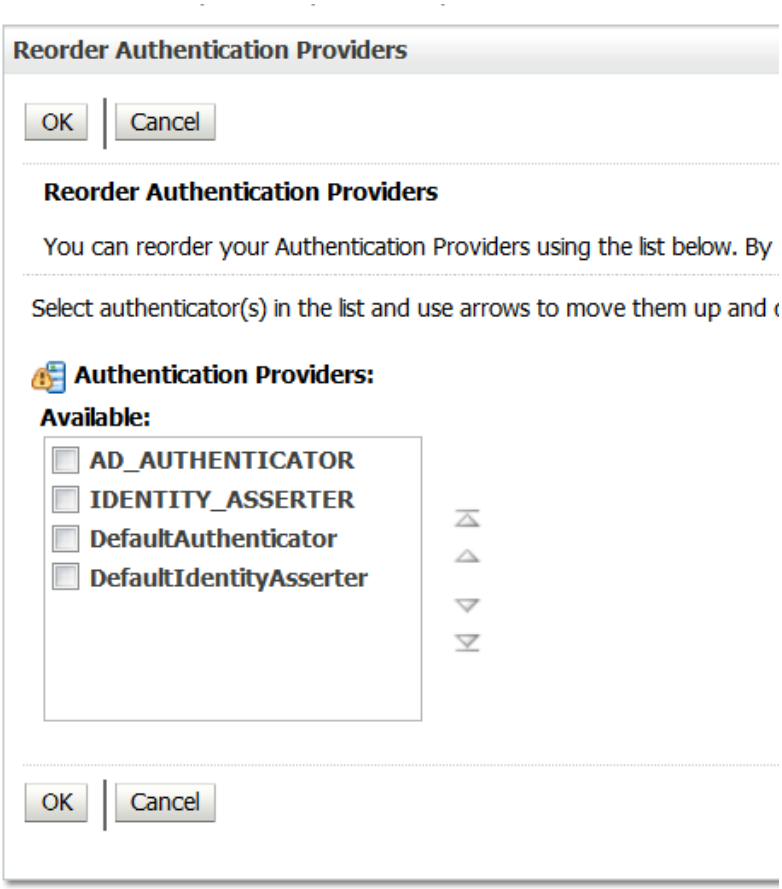

Abb. 7: Reorder Authentication Providers

2. SPNEGO/Kerberos Konfiguration

Tomcat Server:

Editieren der TOMCAT_HOME/bin/catalina.sh:

```
export JAVA_OPTS="-Dsun.security.krb5.debug=true -
Djavax.security.auth.useSubjectCredsOnly=false -
Djava.security.auth.login.config=/opt/oracle/krb5Login.conf -
Djava.security.krb5.realm=TEST.DE -Djava.security.krb5.kdc=test.de
$JAVA_OPTS"
```

WebLogic Server:

Managed Server Startup Argumente hinzufügen:

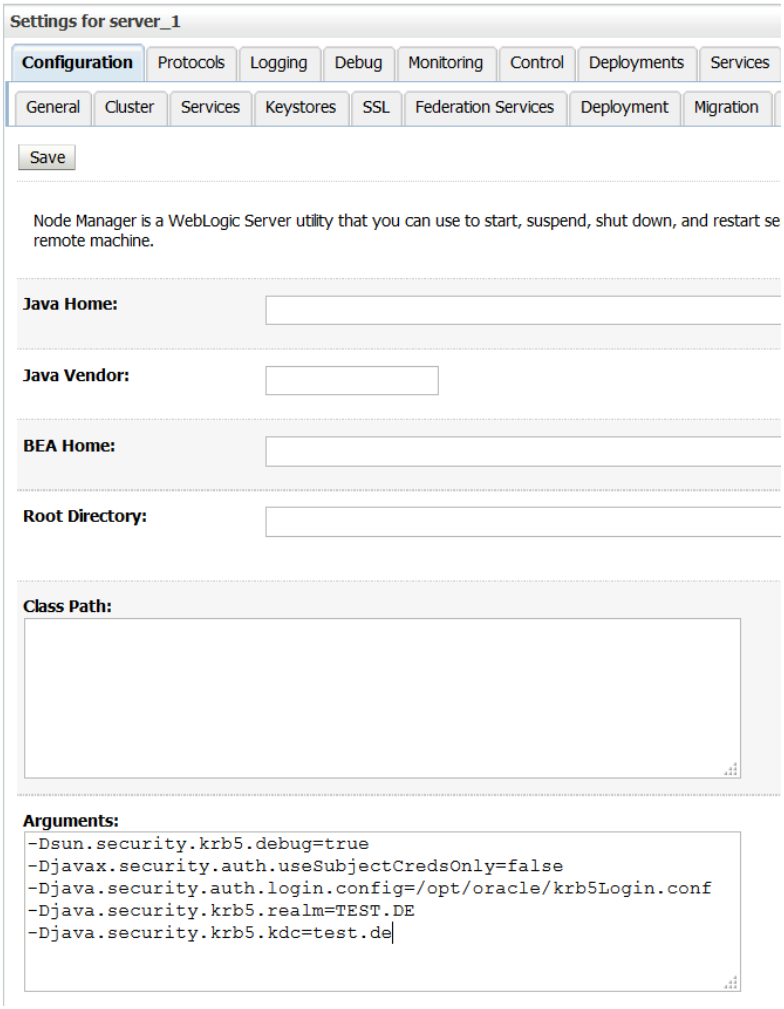

Abb. 8: Modify Managed Server Startup Arguments

3. JAAS Login Konfiguration

Erstellen der krb5Login.conf:

```
com.sun.security.jgss.krb5.initiate {
 com.sun.security.auth.module.Krb5LoginModule required
 principal="HTTP/apphost.test.de@TEST.DE"
 useKeyTab="true"
 keyTab="/opt/tomcattest/HTTP_apphost.test.de.keytab"
 storeKey="true"
 debug="true";
};

com.sun.security.jgss.krb5.accept {
 com.sun.security.auth.module.Krb5LoginModule required
 principal="HTTP/apphost.test.de@TEST.DE"
 useKeyTab="true"
 keyTab="/opt/tomcattest/HTTP_apphost.test.de.keytab"
 storeKey="true"
 debug="true";
};
```

4. Deployment Deskriptor bearbeiten

Tomcat Server:

ords.war WEB-INF/web.xml editieren und folgendes hinzufügen:

```
<security-constraint>
  <web-resource-collection>
    <web-resource-name>Success</web-resource-name>
      <url-pattern>/*</url-pattern>
      <http-method>GET</http-method>
      <http-method>POST</http-method>
    </web-resource-collection>
    <auth-constraint>
      <role-name>*</role-name>
    </auth-constraint>
</security-constraint>
<login-config>
  <auth-method>SPNEGO</auth-method>
</login-config>
<security-role>
  <role-name>*</role-name>
</security-role>
```

WebLogic Server:

ords.war WEB-INF/web.xml editieren und folgendes hinzufügen:

```
<security-constraint>
  <web-resource-collection>
    <web-resource-name>Success</web-resource-name>
      <url-pattern>/*</url-pattern>
      <http-method>GET</http-method>
      <http-method>POST</http-method>
    </web-resource-collection>
    <auth-constraint>
      <role-name>users</role-name>
    </auth-constraint>
</security-constraint>
<login-config>
  <auth-method>CLIENT-CERT</auth-method>
</login-config>
<security-role>
  <role-name>users</role-name>
</security-role>
```

5. ORDS-Konfiguration

java -jar ords.war

Eingabe von:

- Konfigurationsverzeichnis
- Datenbank-Host
- Datenbank-Port
- Datenbank-SID/Service Name
- APEX_PUBLIC_USER
- APEX_PUBLIC_USER Passwort
- APEX_REST_PUBLIC_USER
- APEX_REST_PUBLIC_USER Passwort
- APEX_LISTENER
- APEX_LISTENER Passwort

6. Deployment von ords.war

Tomcat Server:

Kopieren von ords.war ins Verzeichnis TOMCAT_HOME/webapps

WebLogic Server:

java weblogic.Deployer -username weblogic -password passwort -adminurl t3://localhost:7001 -deploy -source ords.war

7. Deployment des APEX Image Verzeichnisses

Tomcat Server:

Kopieren von APEX_DIR/images nach TOMCAT_HOME/webapps/i

WebLogic Server:

Konfiguration des image-Pfads als virtuellen Verzeichnispfad. Das image-Verzeichnis muss auf dem System vorhanden sein!

java -jar ords.war static /opt/oracle/apex/images

Deployen des entstandenen i.war:

java weblogic.Deployer -username weblogic -password passwort -adminurl t3://localhost:7001 -deploy -source i.war

8. Konfiguration des Authentifizierungsschemas für die APEX-Anwendung

1. Schema erstellen: Basiert auf einem vorkonfigurierten Schemas aus der Galerie
2. Name: SPNEGO/Kerberos Authentifzierung
3. Schematyp: HTTP-Header-Variable
4. HTTP-Header-Variablenname: REMOTE_USER
5. Aktion, wenn „Benutzername“ leer ist: Fehler anzeigen
6. Fehlermeldung: Fehler bei Authentifzierung

9. Anwendungsauthentifizierung testen

1. Aufruf der Anwendung: http://apphost.test.de:8080/ords/f?p=150
2. Keine APEX Anmeldeseite wird angezeigt
3. Session Benutzer entspricht Name des Windows-Nutzers

Fazit

APEX Anwendungen lassen sich durch Einsatz von ORDS und der Mächtigkeit von Java Application Servern effektiv in ein Identity- und Access-Management System integrieren. Die Benutzerverwaltung findet im Active Directory statt und die Authentifzierung erfolgt per Single Sign-on sicher auf Basis von SPNEGO/Kerberos.

Kontaktadresse:
Rico Haupt
Robotron Datenbank-Software GmbH
Stuttgarter Straße 29
D-01189 Dresden

Telefon:	+49 (0) 351-25859-2771
Fax:	+49 (0) 351-25859-3696
E-Mail	rico.haupt@robotron.de
Internet:	www.robotron.de