User-Account-Management mit Oracle Internet Directory und Active Directory

Autoren: Rastislav Hil, Cyrill Müller, Michael Gysi-Gander, Trivadis AG

Oracle-User-Accounts und deren Berechtigungen werden meist für jede Datenbank separat angelegt und gepflegt. In großen Umgebungen ist dieses Verfahren nicht nur sehr aufwändig, es bringt in der Regel auch Sicherheitsprobleme mit sich.

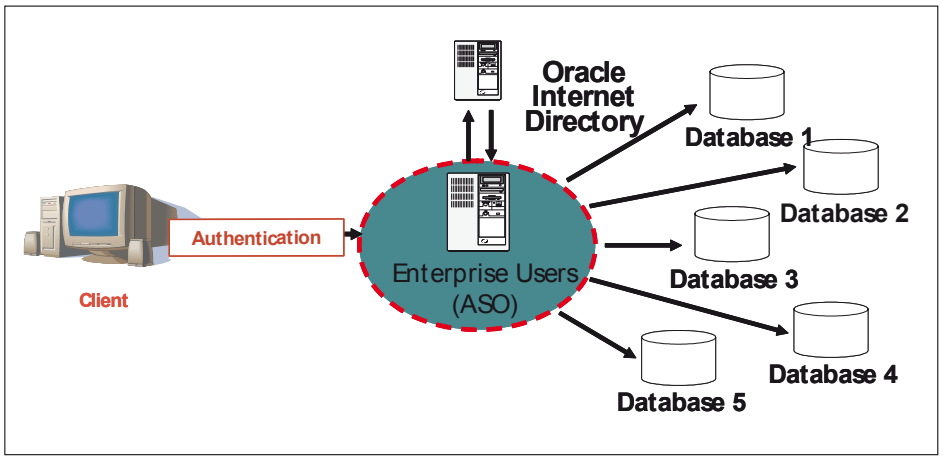

Abbildung 1: Benutzer-Authentifikation über OID

Einerseits wird für jedes System die Sicherheit unterschiedlich stark implementiert, andererseits besteht kaum Transparenz, welche Benutzer auf welche Daten Zugriff haben. Oracle bietet mit dem Oracle Internet Directory die Möglichkeit, User-Accounts und Berechtigungen zentral zu verwalten. Dies vereinfacht die Administration der User-Accounts und kann für die lückenlose Durchsetzung von Security-Policies eingesetzt werden. Was geschieht aber, wenn bereits ein übergeordnetes Directory für die Security-Administration besteht?

Dieser Beitrag zeigt, wie ein zentralisiertes Identity-Management basierend auf Oracle Internet Directory (OID) und Microsoft Active Directory (AD) umgesetzt werden kann. Anhand von zwei ausgewählten Einsatzszenarien wird demonstriert, wie diese Infrastruktur genutzt werden kann:

- Single Sign-on für WebForms (mit Active Directory Passwort)
- DB ProxyAuthentication für Java n-Tier Environment

Zentrale Verwaltung von User-Accounts

Die zentrale Benutzer-Verwaltung der Datenbank-User ist bei Oracle ein Teil der Enterprise-User-Security-Funktionalität. Ein Enterprise-User hat in der Datenbank keinen persönlichen Account mehr, das heißt, er "existiert" nicht direkt in der Datenbank. Dafür ist er im unternehmensweiten Verzeichnisdienst (OID) vorhanden; seine Rechte sind dort über globale Rollen definiert. Der Anmeldevorgang eines Enterprise-Users läuft wie folgt ab:

1. Benutzer meldet sich an der DB an
2. DB sucht das lokale Schema, da es aber nicht existiert, kommt es zum nächsten Schritt
3. DB sucht den Benutzer im Directory (OID)
4. OID prüft Authentifikation
5. OID gibt das Mapping auf ein Shared Schema zurück
6. OID gibt optional eine globale Rolle zurück
7. Benutzer erhält anhand seiner Berechtigungen eine DB-Session

OID ist ein LDAP-v3-konformer Verzeichnisdienst und integraler Bestandteil des Oracle Identity Managements, wie es auf der anderen Seite Active Directory bei einer Microsoft-Infrastruktur darstellt. Für die Speicherung der Daten benutzt OID eine Oracle-Datenbank, in der Verzeichniselemente wie Einträge, Objekte, ACLs (Zugriffsberechtigungen) usw. abgelegt werden.

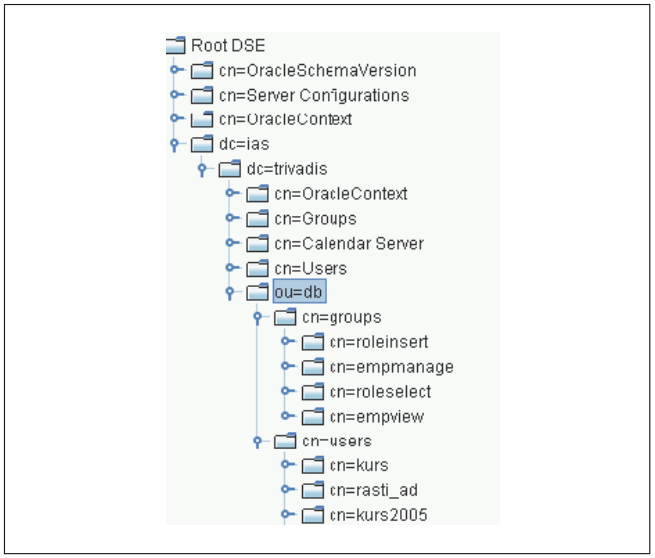

Abbildung 2: Enterprise-User und Rollen werden im Enterprise Security Manager (oemapp esm) verwaltet

Synchronisation mit Active Directory

OID unterstützt die Synchronisation mit beliebigen LDAP-Verzeichnissen. Damit ist es möglich, User-Accounts aus einem Microsoft Active Directory zu übernehmen. Die

Synchronisierung in OID wird durch den Directory-Synchronization-Service gesteuert, der Bestandteil von "Oracle Directory Integration and Provisioning Server" ist.

In einem Profil werden die Synchronisation mit Microsoft Active Directory konfiguriert und das Mapping der LDAP-Attribute in einer Datei festgehalten. Das System synchronisiert nur die LDAP-Einträge, die der "Searchbase" entsprechen, und nur jene LDAP-Attribute, die im Mapping festgelegt sind. Beispiel: Active Directory<>Oracle Internet Directory

- LDAP-Einträge: `ou=users, dc=trivadis, dc=ch:ou=users, dc=trivadis,dc=ch`
- LDAP-Attribute: `userAccountControl:::user : orclisenabled: :orcluserv2`

Der Zugriff auf Active Directory erfolgt über das LDAP-Protokoll. Die Kommunikation läuft über den Port 389 und verwendet einen dedizierten User-Account (synchronization connector), der in der Regel folgende Rechte im Active Directory besitzt:

- LDAP Connect
- Read Attr.:highestCommittedUSN
- Read Attr.:USNChanged
- Read Sync. Entries
- Read Deleted Objects

Bei der Synchronisation können folgende Standard-Fälle abgedeckt werden:

- Benutzer erstellen
- Benutzer löschen (Account nicht mehr vorhanden)
- Gruppen zuteilen/Zuteilung aufheben

Damit sind grundsätzlich zwei Einsatzbereiche für die Synchronisation mit AD denkbar:

1. DB-User-Accounts für Oracle Enterprise User Security werden automatisch über Active Directory erstellt
2. DB-Rechte werden automatisch via Active Directory basierend auf Gruppen-Mitgliedschaften im AD zugeteilt

OID-Plugin

Die genannten Fälle können "Out of the Box" mit AD-OID-Synchronisation umgesetzt werden. Bei weiteren Anforderungen hat man die Möglichkeit, mit OID-Plugins ein gewünschtes Synchronisationsverhalten zu programmieren. Dies ist zum Beispiel notwendig, um Benutzer zu deaktivieren (Account ist noch vorhanden, aber User darf nicht einloggen).

OID Plug-In erweitert die Funktionalitäten von OID.

	OID Out of Box	OID - PLUGIN	Σ
Benutzer erstellen	☑	☑	☑
Benutzer deaktivieren	☐	☑	☑
Benutzer löschen	☑	☑	☑
Gruppen verwalten	☑		☑

Abbildung 3: Single Sign-on für WebForms mit Active Directory Passwort

Ein typisches Problem: Eine Firma setzt Web Forms für Unternehmensapplikationen ein und benutzt Microsoft Active Directory als zentrales Benutzerverzeichnis und Voraussetzung für diverse Dienste. Die Benutzerverwaltung ist umständlich, da jeder Benutzer im Active Directory sowie in jeder Forms-Datenbank erstellt werden muss. Es entsteht ein beträchtlicher Verwaltungsaufwand, der mit zunehmender Anzahl von Benutzern und Applikationen wächst. Für den Benutzer ist es mühsam, sich verschiedene IDs und Passwörter zu merken. Aus Sicht der Unternehmung ist es schwierig, bei der großen Zahl von User-Accounts, die auf verschiedene Systeme verteilt sind, den Überblick zu wahren und sicherzustellen, dass keine "Account-Leichen" oder falsche Berechtigungen existieren.
Die Lösung ist:

- Zentralisierung der Benutzerverwaltung in OID
- Synchronisation der Oracle-Benutzer mit AD, sodass diese nur im AD gepflegt werden müssen
- Einführung von Single Sign-on zur Vereinfachung für den Benutzer und Erhöhung der Sicherheit

Out-of-the-Box-Lösung

Oracle Application Server Single Sign-on (SSO) Integration bietet ein Login für alle Web-Applikationen in der gleichen Authentisierungsdomäne. Die Voraussetzungen dafür sind ein existierendes OID und Oracle Single Sign-on. Diese Funktionalität wird "Out of the Box" mit dem Oracle Application Server geliefert. Im OID sind für SSO-Benutzer ein Datenbank-User sowie ein Kennwort abgelegt. Diese Einträge können entweder vom Benutzer selbst, von den Administratoren oder mit einem Script erstellt werden. Bei diesem Verfahren müssen dem Benutzer bzw. Administrator der DB-User und das Kennwort bekannt sein (für die Erstellung), was keine große Verwaltungserleichterung darstellt.

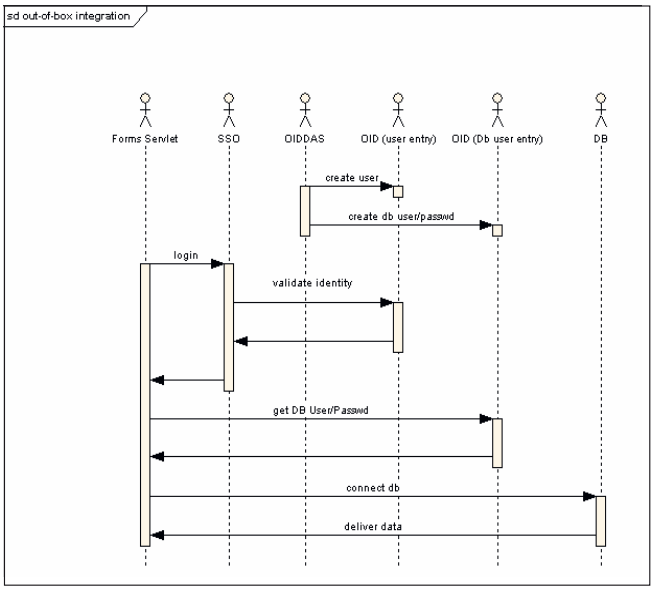

Abbildung 4: Einfache Out-of-the-Box-Lösung

Verbesserte Lösung: Die Erweiterung und Anpassung der Integration beinhaltet die Benutzer-Synchronisation mit

Microsoft Active Directory. Das Anmeldeverfahren des Oracle Single Sign-On greift bei der Überprüfung des Kennworts auf Active Directory zu, weil die Kennwörter nicht synchronisiert werden können. Damit sich im OID trotzdem ein gültiger User erstellen lässt, wird der Benutzer bei der Synchronisation mit zusätzlichen Attributen und Einträgen erweitert. Der OID-Plugin-Mechanismus realisiert diese automatische Erweiterung. Die zusätzlichen Informationen beinhalten z.B. ein Oracle-Datenbank-Kennwort mit einem beliebigen Wert (kann nicht für ein Login verwendet werden). Die Einträge sind in OID für Benutzer unsichtbar und werden nur vom Forms-Servlet bei der Single-Sign-on-Anmeldung sowie bei der Datenbank-Anmeldung im Zusammenhang mit Enterprise User Security gelesen.

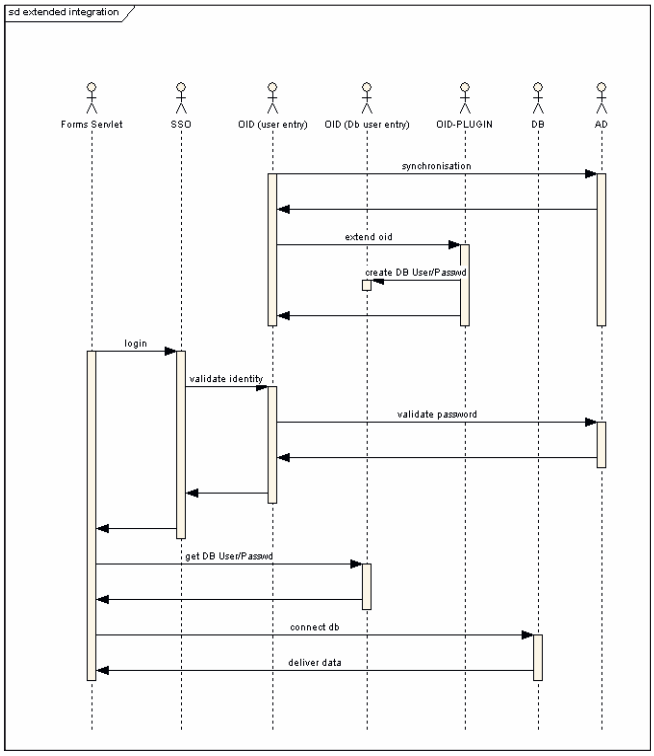

Abbildung 5: Erweiterte Out-of-the-Box-Lösung

Diese Integrationserweiterung verlangt keine Code-Änderung bei Oracle Web Forms und kann bei existierenden Unternehmensapplikationen basierend auf Oracle Web Forms eingebaut werden.

DB Proxy Authentication für Java n-Tier Environment

Problemstellung: Eine Firma entwickelt und betreibt Unternehmensapplikationen vorwiegend mit Java-Technologien und verwendet n-tier Architekturen. Die Datenbestände liegen in Oracle-Datenbanken, worauf aus den verschiedenen Applikationen mit unpersönlichen DB-User-Accounts zugegriffen wird. Dies liegt vor allem daran, dass für den DB-Zugriff der Java Application Server aus Performance-Gründen Connection Pools verwendet werden. Die Firma hat Microsoft Active Directory im Einsatz.

In dieser Konstellation muss man wenig Aufwand für die Verwaltung von DB-User-Accounts betreiben (in der Regel ein Account pro Applikation). Andererseits ist auf Stufe der Datenbank keine Steuerung der Zugriffsrechte mehr möglich und es kann auch nicht nachvollzogen werden, welcher Benutzer auf welche Daten zugegriffen oder Manipulationen getätigt hat (Auditing/Logging). Die Lösung ist:

- Zentralisierung beziehungsweise Neu-Erstellung der Oracle-Benutzerverwaltung in OID
- Synchronisation der Oracle-Benutzer mit AD, sodass diese nur im AD angelegt und gepflegt werden müssen
- Einsatz des ProxyAuthentication-Mechanismus zur Personalisierung der applikatorischen Datenbank-Verbindungen

Oracle bietet mit Proxy Authentication die Möglichkeit, mehrere logische Benutzer zur Datenbank zuverbinden und trotzdem einen Connection Pool verwenden zu können. Die logische User-Identifikation wird dann bis in die Datenbank heruntergeführt, sodass die Sessions eindeutig identifiziert werden können und entsprechend auch nochmals in der Datenbank spezifische Benutzer-Rechte zugewiesen bekommen. Der Benutzer ist bekannt und seine Aktionen lassen sich z.B. via DB Auditing aufzeichnen. Proxy Authentication ermöglicht somit, Sicherheitsvorgaben auf Stufe der DB zu erfüllen, ohne dabei die Vorteile von Connection Pools in n-Tier Umgebungen einzubüßen. So kann in Web-Umgebungen die Proxy Authentication benutzt werden, um die Identifikation von Benutzern von der Applikation bis zur Datenbank zu propagieren. Zunächst wird in der Datenbank ein so genannter Proxy-User eingerichtet, der normalerweise über sehr wenig Rechte verfügt (Create Session). Dieser User Account wird dann nur noch für die Erstellung der Connection Pools eingesetzt.

Nun müssen auch die End-User persönliche DB-Accounts erhalten. Diese können auf jeder Datenbank einzeln oder, wie bereits mehrfach beschrieben, mit OID und Enterprise User Security zentral angelegt werden. Bei der Synchronisation der Benutzer zwischen Active Directory und OID muss wiederum beachtet werden, dass unter Umständen fehlende Funktionalität mithilfe des OID-Plugin-Mechanismus zu implementieren ist.

In dieser Enterprise-User-Umgebung wird dem Shared-User (verbleibender lokaler DB-User) das Recht erteilt, sich durch den Proxy User zu verbinden.

```
ALTER USER shared_user GRANT CONNECT
THROUGH proxy_user;
```

Datenbankseitig ist der User jetzt bekannt (OID) und die Applikationsdatenbank ist für den Zugriff via Connection Pools (Proxy-User) und Proxy Authentication (Shared-User-Berechtigung) bereit. Auf der Stufe der Applikation ist nun der DB-Zugriff im Sourcecode zu ändern. In Java kann wie folgt eine Proxy-Verbindung auf- bzw. abgebaut werden:

```
//Get Connection from Pool
InitialContext ic = new InitialContext();
DataSource nativeDS = (DataSource) ic.lookup(native_ds);
OracleConnection oconn =(OracleConnection)nativeDS.getConnection();

// Open Proxy Session
Properties prop = new Properties();
prop.put(OracleConnection.PROXY_USER_NAME, username);
oconn.openProxySession(OracleConnection.PROXYTYPE_USER_NAME, prop);

//… do your work …

// Close Proxy Session
oconn.close(OracleConnection.PROXY_SESSION);
oconn.close();
```

Dies kann grundsätzlich unter Verwendung des OCI oder des JDBC-THIN-Drivers geschehen, wobei jedoch beim THIN-Driver gewisse Restriktionen bestehen.

Fazit

Das User-Account-Management von Oracle lässt sich mit OID und Enterprise User Security zentralisieren und effizienter gestalten. Durch die vielseitigen und durch den OID-Plugin-Mechanismus beeinflussbaren Synchronisationsmöglichkeiten lässt sich die Account-Administration sogar an weitere Directories delegieren. In der Kombination mit Microsoft Active Directory ergeben sich viele Anwendungsbereiche. Neben den beiden gezeigten Beispielen sind denkbar:

- Zentralisiertes Account-Management via AD und Kerberos für SSO-DB-Login
- Zentralisiertes Account-Management via AD und Enterprise User Security für Forms-Applikationen

Literatur

Effective Oracle Database 10*g* Security by Design, David C. Knox, Oracle Press, 2004

Kontakt:
Rastislav Hil
Cyrill Müller
Mike Gysi-Gander
info@trivadis.com