Zentrale Benutzerverwaltung mit EUS, OUD und Active Directory

Stefan Oehrli, Trivadis AG

Sicherheit ist heutzutage eine der zentralen Herausforderungen für On-Premise- und Cloud-basierte Datenbanken. Mit der Umsetzung von Sicherheitskonzepten für Datenbanken werden diese Herausforderungen gemeistert. Doch viele dieser Bestrebungen sind nur sinnvoll, wenn bereits die Authentifizierung und Autorisierung mit entsprechender Sorgfalt umgesetzt wird.

Anstelle der dezentralen Verwaltung der Benutzer, Rechte und Rollen in jeder Datenbank ist es übersichtlicher und vor allem sicherer, diese zentral zu verwalten. Dieser Artikel zeigt, wie die Benutzerverwaltung mit Oracle Enterprise User Security und Oracle Unified Directory zentral verwaltet und mit MS Active Directory integriert werden kann.

In der Regel wird für die Authentifizierung des Oracle-Datenbank-Benutzers die lokale Authentifizierung in der Datenbank mit Benutzername und Passwort verwendet. Mit einem entsprechend sicheren Passwort ist diese Methode für viele Umgebungen weiterhin eine akzeptable Authentifizierungsmethode. Sie entspricht allerdings nicht mehr den aktuellen Sicherheitsempfehlungen. Je mehr Benutzer und Datenbanken vorhanden sind, desto unübersichtlicher und somit unsicherer wird dabei die Benutzer-Administration. Wo hat welcher Benutzer Zugriff? Wer hat wo welche Rechte? Werden bei Funktions- oder Rollenwechseln keine Rechte vergessen? Viele dieser Fragen lassen sich bei einer dezentralen Benutzerverwaltung nicht so einfach beantworten.

Die verzeichnisbasierte Authentifizierung beziehungsweise Oracle Enterprise User Security (EUS) bietet eine alternative Methode zur Authentifizierung von Datenbankbenutzern. Im Gegensatz zur lokalen Datenbank-Authentifizierung geht EUS über die reine Authentifizierung hinaus. Mit einem Oracle-Verzeichnis lassen sich Benutzer, Gruppen und verschiedene Rollen verwalten, also auch die Autorisierung. Der wesentliche Vorteil dieser Lösung ist die zentrale Verwaltung. Es gibt nur eine Stelle, an der Benutzer erstellt und Berechtigungen erteilt werden. So ist es beispielsweise nicht mehr nötig, dass der Datenbank-Administrator in jeder seiner Datenbanken Benutzer anlegt oder löscht.

EUS setzt auf die Oracle-Identity-Management-Infrastruktur, die wiederum einen LDAP-kompatiblen Verzeichnis-

dienst verwendet, um Benutzer zentral zu speichern und zu verwalten. Dabei werden folgende Verzeichnisdienste unterstützt:

- OID Oracle Internet Directory ist ein LDAP-v3-konformes Verzeichnis. Es basiert auf einer Oracle-Datenbank und ist vollständig in Oracle Fusion Middleware und Oracle Applications integriert.
- OUD Oracle Unified Directory ist ein Java-basiertes Verzeichnis. Es ist eine All-in-one-Lösung mit Storage, Proxy, Synchronisations- und Virtualisierungsfähigkeiten.

Die eigentliche Authentifizierung erfolgt bei EUS je nach Konfiguration entweder durch Passwort, Kerberos oder SSL-Authentifizierung. Die Benutzer profitieren dabei von einer Single-Sign-on (SSO) oder Single-Passwort-Authentifizierung.

Trotz der Vorteile einer zentralen Benutzerverwaltung wird bei vielen Unternehmen die Authentifizierung und Autorisierung weiterhin dezentral in den Oracle-Datenbanken gelöst. Die Gründe dafür sind vielfältig. Ohne im Detail weiter auf diese einzugehen, lässt sich aber mit Sicherheit sagen, dass viele Unternehmen nicht die nötigen Ressourcen aufbringen können, um einen weiteren mehr oder weniger komplexen Verzeichnisdienst zu betreiben.

Obwohl Benutzer- und Gruppen-Informationen bereits zentral in einem MS-Active-Directory oder LDAP-Verzeichnis verwaltet werden, lassen sich diese nicht direkt für EUS verwenden. Es braucht immer zwingend ein Oracle-Verzeichnis wie OID oder OUD. Das bedeutet, dass neben einem weiteren Verzeichnis auch eine Synchronisation aufgebaut und betrieben werden muss. Dies führt zu einer redundanten Datenhaltung sowie komplexeren Architektur und birgt das Risiko erhöhter Fehleranfälligkeit, was schlussendlich nicht im Interesse des Kunden ist.

Oracle Unified Directory AD Proxy Server

Im Gegensatz zu OID kann OUD aber nicht nur als reiner Verzeichnis-Server, sondern auch als Proxy-Server betrieben werden. Dabei nutzt OUD die Benutzer- und Gruppen-Informationen im bestehenden Verzeichnis und ruft diese direkt mit LDAP-Abfragen ab. EUS-spezifische Informationen wie Datenbank-Registrierungs-Informationen, Benutzer- und Rollen-Zuordnungen sowie weiteren EUS-spezifischen Metadaten werden lokal im OUD gespeichert. Auf diese Weise arbeitet OUD als Echtzeit-Interpreter für die Oracle-Datenbanken.

Anfragen nach Benutzer-Informationen werden transparent beantwortet, unabhängig davon, ob diese lokal im OUD abgelegt sind oder aus einem zentralen Verzeichnis abgerufen werden. Somit müssen die Daten nicht synchronisiert oder redundant abgelegt werden, was wiederum die Gesamtbetriebskosten reduziert.

Abbildung 1 zeigt den schematischen Aufbau einer EUS-Datenbank mit einem OUD-AD-Proxy und MS-Active-Directory als zentralen Verzeichnis-Server. Der Verbindungsaufbau vom Benutzer „Scott" und damit die Authentifizierung und Autorisierung ist mit den Pfeilen dargestellt.

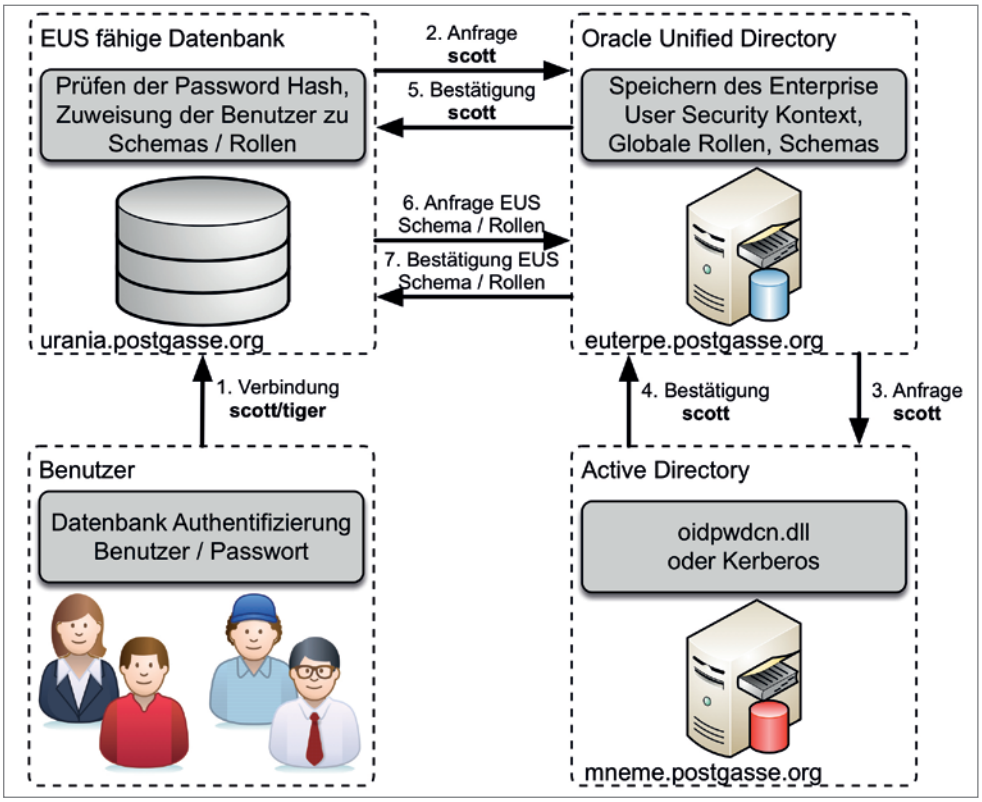

Abbildung 1: Enterprise-User-Security mit Active-Directory-Integration

Authentifizierung

Je nach Konfiguration stehen Passwort, Kerberos oder SSL-Authentifizierung zur Verfügung. Am einfachsten ist die Passwort-Authentifizierung, bei der sich der Datenbank-Benutzer mit seinem Windows-Account-Passwort an der Datenbank anmeldet, wobei auch diese nicht ganz ohne Hürden umgesetzt werden kann. Die vom Microsoft verwendeten Passwort-Hashes sind nicht direkt mit Oracle-Datenbanken kompatibel. Daher muss auf jedem Active-Directory-Domain-Controller das Oracle-Password-Change-Notification-Plug-in („oidpwdcn.dll") installiert sein. Dazu gehört auch eine Active-Directory-Schema-Erweiterung, damit das Plug-in den Oracle-spezifischen Hashwert abspeichern kann. Wird hingegen Kerberos als Authentifizierungsmethode eingesetzt, sind das Plug-in und die Schema-Erweiterung nicht erforderlich. Dafür kann mit Kerberos zusätzlich ein Single-Sign-on (SSO) erreicht werden.

Voraussetzungen

Für den Aufbau der in *Abbildung 1* vorgestellten Architektur sind folgende Voraussetzungen zu erfüllen:

- Oracle Enterprise Edition 12c Datenbank (12.1.0.2) mit LDAP-Client-Patch 19285025 und einer EUS-Test-Datenbank

- Oracle Unified Directory 11.1.2.3.0 mit aktuellem Bundle-Patch 25383162
- MS Active Directory Domain Controller auf Windows 2012 R2
- Oracle Enterprise Manager Cloud Control 13*c* R2 für die Konfiguration von Enterprise User Security. Alternativ kann hier auch eine ältere OEM-Version oder das EUSM-Command-Line-Utility verwendet werden (siehe MOS Note 1085065.1, EUSM, Command Line Tool for EUS Administration and Some EUS Good to Know)

Konfiguration Active Directory

Damit die EUS-Passwort-Authentifizierung auch mit Active-Directory funktioniert, sind ein OUD-Passwort-Synchronisations-Plug-in sowie eine Schema-Erweiterung notwendig. OUD liefert die entsprechende Java-Klasse, um das Attribut „orclCommonAttribute" zu erstellen. Die Schema-Erweiterung wird mit dem Aufruf der Java-Klasse direkt aus dem OUD-Home-Verzeichnis „$OUD_HOME/OUD1/config/EUS/ActiveDirectory" erstellt *(siehe Listing 1)*.

Das OUD-Passwort-Synchronisations-Plug-in „oidpwdcn.dll" liegt im selben Verzeichnis und steht als 32- oder 64-Bit-Version zur Verfügung. Die DLL-Datei muss für die Installation auf dem Active-Directory-Server in das Verzeichnis „C:\Windows\system32" kopiert werden. Damit sie zum Einsatz kommt, muss in der Registry zusätzlich der Registry Key „HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Lsa\Notification" um „oidpwdcn" erweitert werden. Mit dem Restart des Active-Directory-Servers ist die Installation abgeschlossen. Sind mehrere Active-Directory-Server vorhanden, muss das Plug-in analog auf jedem Server installiert sein. Wird dagegen ausschließlich Kerberos zur Authentifizierung verwendet, sind die oben aufgeführten Anpassungen nicht erforderlich.

Aufbau der OUD-Proxy-Instanz

Für das Erstellen der verschiedenen OUD-Instanzen liefert Oracle entsprechende Setup-Programme, die alle im OUD-Installations- beziehungsweise OUD-Home-Verzeichnis bereitstehen. Je nach Bedarf können die Programme mit einer grafischen Oberfläche, als Command-Line-Installation oder auch als Silent-Installation ausgeführt werden. Die Silent-Installation ist vorwiegend hilfreich, wenn mehrere Instanzen mit einem Skript automatisiert aufgebaut werden sollen.

Die OUD-Proxy-Instanz selbst wird mit „oud-proxy-setup" erstellt. Ohne Angabe eines Parameters wird direkt die grafische Oberfläche für die Eingabe der Server- und Instanz-Konfiguration gestartet. Für EUS und die MS-AD-Integration muss dabei zwingend LDAPS aktiviert sein. Entweder wird ein kundenspezifisches SSL-Zertifikat verwendet oder wie beim Aufbau der Testumgebung ein selbstsigniertes Zertifikat *(siehe Abbildung 2)*. Für Produktionsumgebungen wird empfohlen, jeweils nur korrekt signierte Zertifikate zu verwenden.

```
java ExtendAD -h mneme.postgasse.org -p 389 -D cn=administrator,cn=user
s,dc=postgasse,dc=org -w <pwd> -AD
dc=postgasse,dc=org -commonattr
```

Listing 1

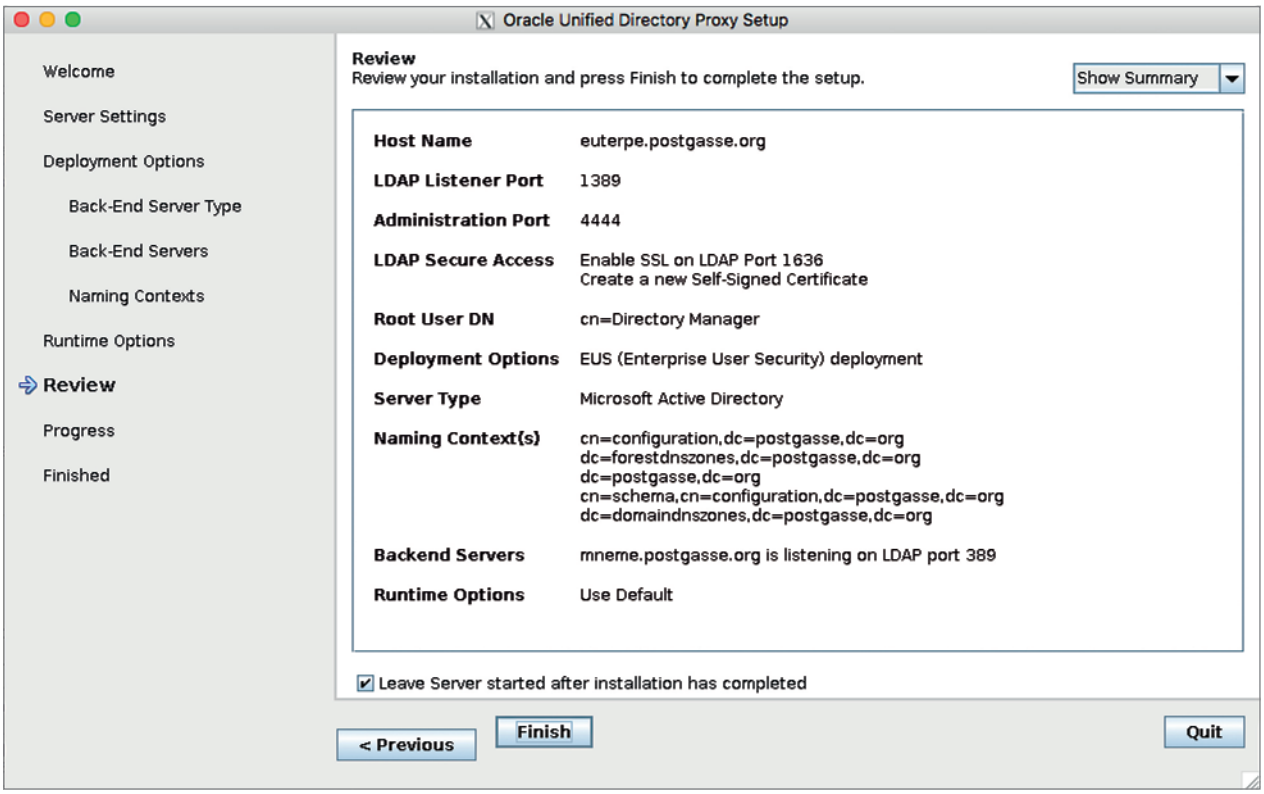

Abbildung 2: Zusammenfassung des OUD-Proxy-Setup

Umgebungsvariable	Pfad
ORACLE_BASE	/u00/app/oracle
OUD_HOME	/u00/app/oracle/product/oud11.1.2.3
ORACLE_HOME	/u00/app/oracle/product/oud11.1.2.3/OUD1
Verzeichnis für die OUD Instanzen	/u00/app/oracle/instances
INSTANCE_NAME	../../instances/oud_ad_proxy

Tabelle 1

In den nächsten Schritten werden als Proxy-Konfiguration „Configure EUS" festgelegt und MS-Active-Directory als Datenquelle mit dem entsprechenden EUS-Namenskontext ausgewählt. *Abbildung 2* zeigt zusammenfassend alle Einstellungen. Durch die Bestätigung mit dem Button „Finish" wird anschließend eine OUD-Proxy-Instanz erzeugt. Als Standard entsteht dabei immer eine Instanz mit dem Namen „ASINST_1" im OUD-Home-Verzeichnis.

Grundsätzlich wird empfohlen, die OUD-Instanz explizit zu benennen und in einem anderen Verzeichnis beziehungsweise auf einem anderen Filesystem als die OUD-Software anzulegen. Ähnlich wie bei der Oracle Flexible Architecture (OFA) ist so sichergestellt, dass Daten und Konfiguration getrennt von der Software verwaltet werden. Um eine OUD-Instanz explizit zu benennen, muss der Instanz-Name vor dem Aufruf von „oud-proxy-setup" via Umgebungsvariable „INSTANCE_NAME" festgelegt sein. Der Instanz-Name ist dabei als Pfad relativ zum OUD-Home-Verzeichnis anzugeben. Für die EUS-Test-Umgebung wurde als Instanz-Name „oud_ad_proxy" gewählt. *Tabelle 1* zeigt die verschiedenen Verzeichnisse und Umgebungsvariablen.

Bevor die neu erstellte OUD-Proxy-Instanz nun für EUS verwendet werden kann, braucht es noch kleinere Anpassungen. Als Erstes werden das Proxy-Workflow-Element angepasst und die Remote-Bind-Anmelde-Informationen festgelegt. Im Folgenden wird dabei der AD-Benutzer „OUD Admin" verwendet. Grundsätzlich kann dies aber ein beliebiger AD-Benutzer sein, der vollen Lesezugriff auf Benutzer- und Gruppen-Informationen hat. Zusätzlich wird auch der Client-Connection-Mode auf „use-specific-identity" gesetzt. Diese Einstellung ist nötig, wenn das externe Verzeichnis keine anonymen Bind-Anfragen zulässt. Die Anpassungen werden mit dem OUD-Command-Line-Admin-Tool „dsconfig" ausgeführt *(siehe Listing 2)*.

Damit EUS die Benutzer- und Gruppen-Informationen findet, müssen die entsprechenden Einträge im EUS-Kontext ebenfalls angepasst werden. Oracle liefert hierzu im Verzeichnis „$OUD_HOME/OUD1/config/EUS" eine LDIF-Template-Datei „modifyRealm.ldif", die wie folgt angepasst werden muss:

- Ersetzen von „dc=example,dc=com" mit dem korrekten Namenskontext
- Ersetzen von „ou=people" und „ou=groups" durch den entsprechenden Ort für die Benutzer- beziehungsweise Gruppen-Informationen

Das angepasste LDIF-Template wird anschließend mit „ldapmodify" geladen: „ldapmodify -h euterpe.postgasse.org -p 1389 -D „cn=Directory Manager" -w <PASSWORT> -f modifyRealm.ldif". Grundsätzlich ist die OUD-Proxy-Instanz nun bereit für EUS. Oracle-11*g*-Datenbanken können direkt in OUD registriert und für EUS konfiguriert werden.

Für Oracle 12*c* beziehungsweise EUSM 12*c* und Oracle Cloud Control 12*c*/13*c* ist hingegen noch eine weitere Anpassung notwendig. Die aktuellen Versionen der Oracle-Datenbank und Oracle-Cloud-Control verwenden den SASL-Mechanismus, um sich mit der OUD-Instanz zu verbinden. Allerdings funktioniert dies in OUD standardmäßig nicht. Das Admin-Passwort des EUS-Users muss zusätzlich zwingend mit einem reversiblen Algorithmus abgespeichert sein. Dazu ist das Root-Passwort-Profile zu erweitern, sodass neue Passwörter auch mit dem AES-Algorithmus abgespeichert werden *(siehe Listing 3)*. Mit einem expliziten Neusetzen des Passworts ist sichergestellt, dass auch das neue Passwort-Storage-Schema verwendet wird *(siehe Listing 4)*.

Für Produktionsumgebungen ist es sinnvoll, dass anstelle des Root-Passwort-Profils ein dezidiertes Passwort-Profil für EUS sowie entsprechende EUS-Admin-Benutzer erstellt werden.

Datenbank-Konfiguration

Damit die Datenbank im OUD registriert werden kann, wird im Netzwerk-

```
dsconfig set-workflow-element-prop \
  --element-name proxy-we1 \
  --add exclude-list:cn=directory\ manager \
  --add exclude-list:cn=oraclecontext,dc=postgasse,dc=org \
  --set remote-root-dn:"cn=OUD Admin,cn=users,dc=postgasse,dc=org" \
  --set remote-root-password:<PASSWORT> \
  --set remote-ldap-server-bind-dn:"cn=OUD Admin,cn=users,dc=postgasse,dc=org" \
  --set remote-ldap-server-bind-password:<PASSWORT> \
  --set ldap-server-extension:proxy1 \
  --set client-cred-mode:use-specific-identity \
  -h euterpe.postgasse.org -p 4444 -X \
  -D cn=directory\ manager -j $etc/oud_ad_proxy.pwd -n
```

Listing 2

Verzeichnis der Datenbank die „ldap.ora“-Datei angepasst. Dazu werden der OUD-Server und der Default-Kontext angegeben. Im Folgenden ist die „ldap.ora“-Datei aufgeführt, wie sie bei der in *Abbildung 1* beschriebenen Testumgebung verwendet wird. Als Directory-Server-Typ muss auch bei OUD „OID“ angegeben sein *(siehe Listing 5)*.

In „sqlnet.ora“ wird die Namesauflösung angepasst und neu „LDAP“ als zusätzliche Quelle angegeben. Dazu ein Auszug aus der Datei: „NAMES.DIRECTORY_PATH=(LDAP, TNSNAMES, EZCONNECT)“.

Die eigentliche Registrierung der Datenbank in OUD erfolgt mit dem „dbca“ interaktiv wie in *Abbildung 3* oder „silent“ via Command-Line. Für die Registrierung ist die Anmelde-Information des EUS-Admins erforderlich. Als Standard-Database-CN wird der DB-Unique-Name verwendet, optional lässt sich dieser aber auch anpassen. Ist für diese Datenbank noch kein Oracle-Wallet vorhanden, legt der „dbca“ ein neues Wallet im Admin-Verzeichnis der Datenbank an und speichert darin die „OUD DB“-Credentials.

Neben der OUD-Registrierung und dem Wallet passt der „dbca“ zusätzlich auch die Datenbank-Parameter „ldap_directory_access“ sowie „ldap_directory_sysauth“ entsprechend an. Die Basis für EUS ist nun sichergestellt. Was nun noch fehlt, sind entsprechende EUS-Rollen beziehungsweise Datenbank-Benutzer und -Rollen sowie die Zuweisung zu den verschiedenen EUS-Datenbanken. Als einfaches Beispiel wird mit SQL*Plus ein globales Shared-Schema erstellt, das die lokalen Rolle „connect“ erhält *(siehe Listing 6)*. Anschließend wird mit EUSM für die Datenbank „TDB121A“ ein einfaches Mapping für alle Benutzer einer bestimmten Organisations-Unit („ou“) erstellt. Somit kann sich nun jeder Benutzer in der Organisations-Unit „ou=People,dc=postgasse,dc=org“ in MS-AD mit der Datenbank verbinden. Die Abfrage des „SYS_CONTEXT“ gibt entsprechend Auskunft über die Enterprise Identity *(siehe Listing 8)*.

Das Beispiel zeigt, wie einfach man ein „1:1“-Mapping erstellen und sich anschließend als Enterprise-Benutzer an der Datenbank anmelden kann. In der Praxis wird das Mapping komplexer ausfallen. Gewisse EUS-Benutzer werden globale private Schemata haben, andere dagegen mit globalen Shared-Schemata und Enterprise-Rollen arbeiten. In der Kombination mit Proxy-Rechten und Administrativen-Privilegien wie „SYSDBA“ lassen sich so umfangreiche und komplexe Benutzer- und Rollen-Konzepte umsetzen.

```
dsconfig set-password-policy-prop \
   --policy-name "Root Password Policy" \
   --add default-password-storage-scheme:"AES" \
   -h euterpe.postgasse.org -p 4444 -X \
   -D cn=directory\ manager -j $etc/oud_ad_proxy.pwd -n
```

Listing 3

```
ldappasswordmodify -h euterpe.postgasse.org -p 4444 \
   -D "cn=Directory Manager" -j $etc/oud_ad_proxy.pwd -Z \
   -c <PASSWORT> -n <PASSWORT>

ldapsearch -h euterpe.postgasse.org -p 4444 \
   -D "cn=Directory Manager" \
   -j $etc/oud_ad_proxy.pwd -Z -X \
   -b "cn=Directory Manager,cn=Root DNs,cn=config" \
   -s base objectclass=* userpassword

dn: cn=Directory Manager,cn=Root DNs,cn=config
userpassword: {AES}AfLk8S3k8k3ekXLV6qaP+mFUe/DStQ4ngUAdo6P5EjOKOM-
m4AN27Xw==
userpassword: {SSHA512}YSI4tlv3gO4z0czFU9EdTOGnfmGhQqOtbWUopV7xgcPDlYS/
Eea3ydQUt
 YT5Qog/ngZ8w4M3EHNf4/MObnPC8M+lblEIKivf
```

Listing 4

```
DIRECTORY_SERVERS= (euterpe.postgasse.org:1389:1636)
DEFAULT_ADMIN_CONTEXT = "dc=postgasse,dc=org"
DIRECTORY_SERVER_TYPE = OID
```

Listing 5

Patch, Bugs und weitere Sorgen

Im Zusammenhang mit LDAP, SSL und der Oracle-Datenbank gibt es noch einen Bug, der zwingend behoben werden muss, damit die Anmeldung so wie oben beschrieben funktioniert. Der Bug 19285025 ist bis dato in keinem Base-Release oder Patch Set Update behoben. Daher muss der entsprechende Patch 19285025 weiterhin explizit bei allen aktuellen Versionen wie Oracle 11.2.0.4, 12.1.0.2 und leider auch 12.2.0.1 installiert werden.

Neben diesem Bug gibt es im SSL- und Kerberos-Umfeld generell den einen oder anderen Bug. Diese können mit entsprechenden Workarounds oder Patches umgangen werden. Nichtsdestotrotz ist das Troubleshooting nicht immer ganz einfach. Schließlich ist die Architektur bei der zentralen Benutzerverwaltung komplexer. Neben der Datenbank sind eben auch OUD und MS-AD bei einer Anmeldung an der Datenbank beteiligt. Die effektiven Ursachen von ORA-01017 oder ORA-28030 sind daher häufig nicht direkt identifizierbar. Mithilfe des OUD Access Log, SQL Net Tracing oder eines entsprechenden Oracle-Events kommt man der effektiven Ursache jedoch in der Regel schnell auf die Spur.

Weitere Überlegungen

Die in *Abbildung 1* aufgezeigte Architektur ist für eine Test- und Engineering-Umgebung ausreichend. Für den produktiven

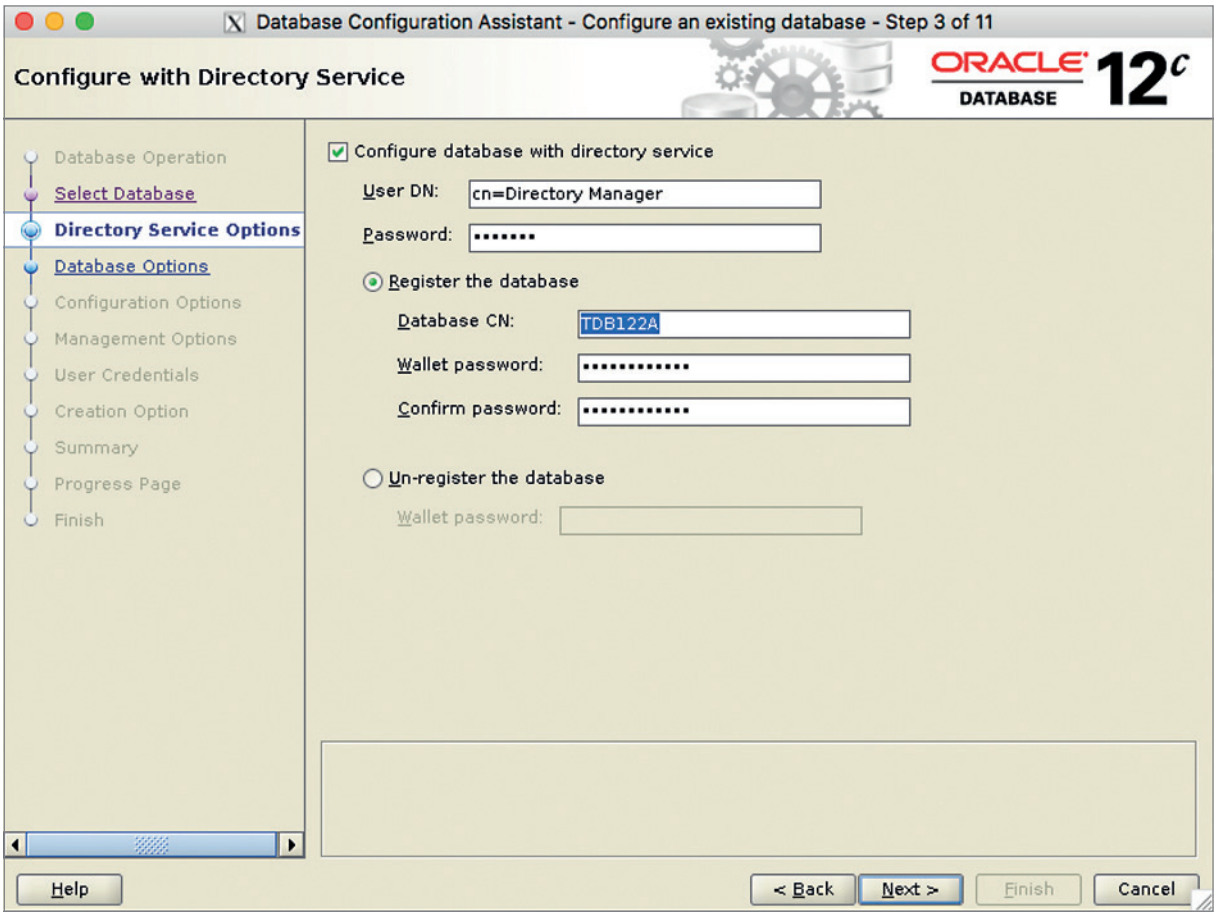

Abbildung 3: Datenbank-Registrierung mit „dbca"

Betrieb sind sicher weitere Maßnahmen nötig, damit die Hochverfügbarkeit und das Desaster Recovery sichergestellt werden können. Schlussendlich ist eine zentrale Benutzerverwaltung ein Single Point of Failure. Ist sie nicht verfügbar, können sich die Benutzer nicht mehr an den Datenbanken anmelden. OUD bietet verschiedene Möglichkeiten wie Online-Backup und Replikation, um die OUD-Instanzen sicher und hochverfügbar zu betreiben. Diese Maßnahmen sind in einem entsprechenden OUD-Konzept zu planen und umzusetzen.

Fazit

Mit Oracle-Unified-Directory und dem AD-Proxy wird die Umsetzung einer zentralen Benutzerverwaltung mit Enterprise User Security wesentlich vereinfacht. Ein OUD-AD-Proxy ist grundsätzlich schnell aufgebaut, sodass man direkt den ersten Nutzen hat. Nichtsdestotrotz liegen

```
CREATE USER global_shared IDENTIFIED GLOBALLY;
GRANT connect TO global_shared;
```

Listing 6

```
eusm createMapping domain_name="OracleDefaultDomain" realm_
dn="dc=postgasse,dc=org" map_type=SUBTREE map_dn="ou=People,dc=pos
tgasse,dc=org" schema=GLOBAL_SHARED ldap_host="euterpe.postgasse.
org" ldap_port=1389 ldap_user_dn="cn=Directory Manager" ldap_user_
password=<PASSWORT>
```

Listing 7

```
SQL> connect oehrli
Enter password:
Connected.
SQL> show user
USER is "GLOBAL_SHARED"
SQL> SELECT
   sys_context('userenv','SESSION_USER') "USER",
   sys_context('userenv','AUTHENTICATED_IDENTITY') "AUTH_IDENTITY",
   sys_context('userenv','ENTERPRISE_IDENTITY') "ENTERPRISE_IDENTITY"
FROM DUAL;
USER             AUTH_IDENTITY  ENTERPRISE_IDENTITY
---------------  -------------- ----------------------------------------
GLOBAL_SHARED OEHRLI         cn=Stefan Oehrli,ou=People,dc=postgasse,dc=org
```

Listing 8

die Herausforderungen in den Details. So sind die MS-AD-Integration und Kerberos-Konfiguration bei komplexen AD-Architekturen um einiges aufwändiger. Zudem braucht es für den unternehmensweiten Einsatz zwingend ein geeignetes Benutzer- und Rollen-Konzept. Dafür muss häufig mehr Zeit eingeplant werden als für die einfache Konfiguration von OUD. Die verschiedenen Bugs im Kerberos-, LDAP- beziehungsweise SSL-Umfeld erschweren einem das Leben zusätzlich. Sind diese Hürden erst einmal gemeistert, hat man aber ein verlässliches und sicheres System für die zentrale Benutzerverwaltung von Oracle-Datenbanken.

Stefan Oehrli
stefan.oehrli@trivadis.com

Scripting mit SQLcl – Batch Scripts auf einem neuen Level

Robert Marz, its-people GmbH

SQLcl steht bereit, das gute alte SQL*Plus als Kommandozeile für die Datenbank abzulösen und dessen Funktionsumfang auf die Höhe der Zeit bringen. Die spannendste Neuerung ist das Scripting: Batch Scripts können in JavaScript verfasst werden. Aber auch andere Sprachen wie Python oder Lua sind möglich. Dadurch eröffnet sich ein wahres Füllhorn an Möglichkeiten, von denen „viele erstmal nicht offensichtlich sind.

SQLcl steht für „Oracle SQL Developer Command Line". Wie der Name vermuten lässt, wurde der komplette Funktionsumfang des SQL-Developer-Worksheet als Kommandozeile ausgekoppelt. SQLcl wird aktiv entwickelt: Zur Oracle Open World im September 2016 wurde ein Produktions-Release veröffentlicht, das seitdem alle sechs bis acht Wochen ein Update erhält.

SQLcl möchte das neue SQL*Plus werden und ist seit der Datenbank-Version 12c Release 2 im „$ORACLE_HOME/bin"-Verzeichnis zu Hause. Wer die neueste Datenbank-Version bereits im Einsatz hat, kann das Tool direkt benutzen. Alle anderen können es von der SQL-Developer-Seite des Oracle-Technologie-Networks herunterladen. Der Download ist mit einer Größe von rund 19 MB überschaubar. Die Installation ist mit dem Entpacken in ein beliebiges Verzeichnis abgeschlossen. Winzige Voraussetzung ist ein aktuelles Java im Suchpfad des Betriebssystems. Ein zusätzlicher Oracle-Client ist nicht nötig.

SQL*Plus „plus"

SQLcl deckt bis auf einige gewollte Ausnahmen den kompletten Funktionsumfang von SQL*Plus ab und enthält außerdem die Erweiterungen des SQL-Developer-Worksheet. Die Entwickler haben dem Tool darüber hinaus die Möglichkeit gegeben, Scripts in anderen Programmiersprachen auszuführen. Einzige Bedingung: Es muss eine Implementierung der Sprache existieren, die dem Standard „JSR-223" folgt und damit in der