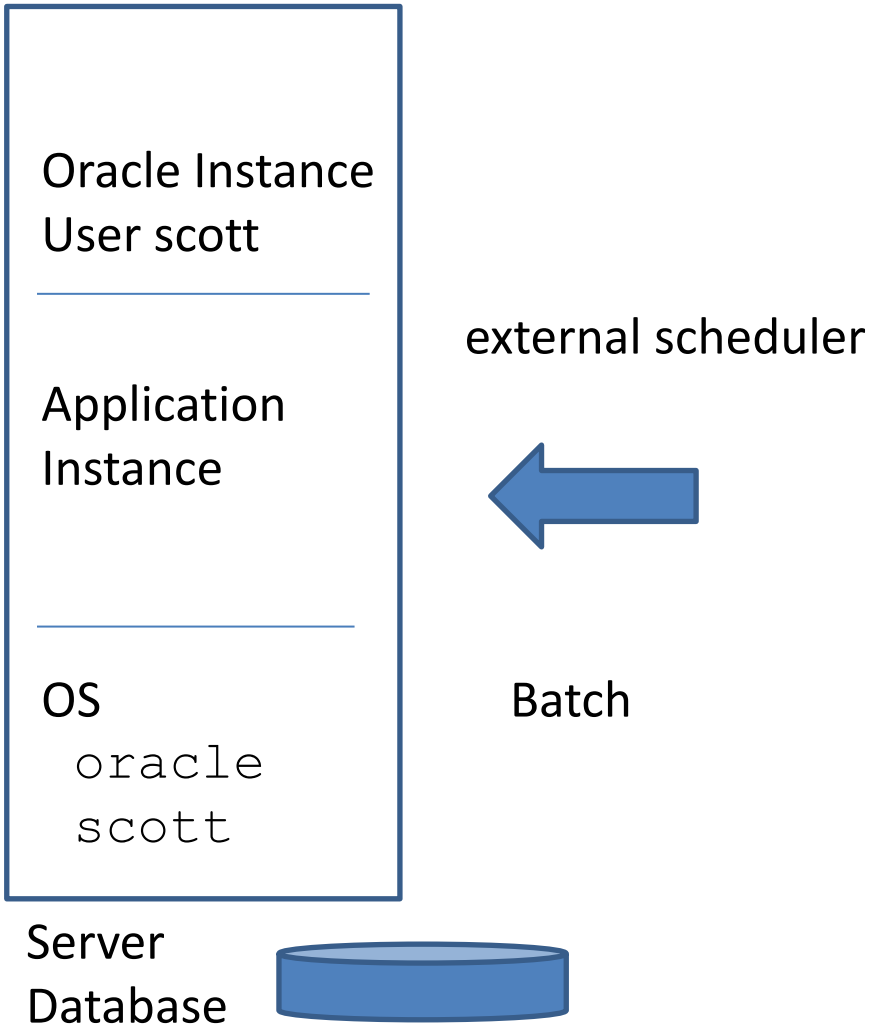
Proxy Authentication and Secure External Password Store

Matthias Mann, Database Community

11.04.2014, DOAG Webinar

Oracle Secure External Password Store

Application Centric System Architecture with external Authentication



Oracle Secure External Password Store

External Authentication

```
SQL> create user scott profile pf_01 identified  externally;
$ whoami
$ scott
$ sqlplus /
SQL>show user
SQL>User is "scott"
```

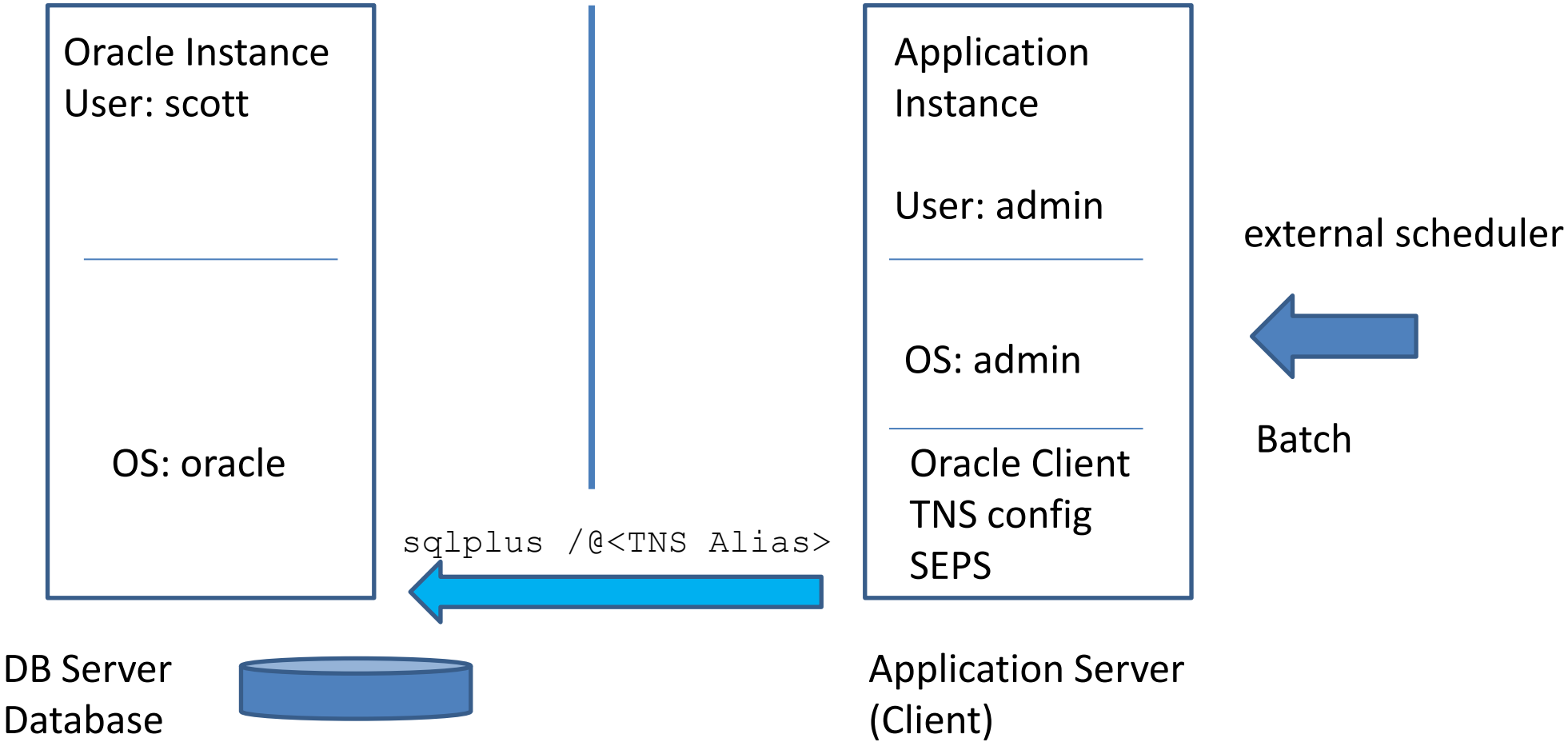
- client host OS acts as trusted authority
- requires identical user scott on OS level
- possible local to the DB and remote (remote_os_authent)

- used in the past for batch scripts (no need for passwords)
- not compatible with multi tier architectures
- remote_os_authent: uncalculable security risk => should not be used

- alternative: wallet based authentication (client / server based)

Oracle Secure External Password Store

Multi Tier Architecture with SEPS



Oracle Secure External Password Store

Step by Step Configuration on Application Server (Client)

Prerequisites

```
ORACLE_HOME                    - path to the client software installation
TNS_ADMIN                      - configuration files for naming resolution
PATH=$ORACLE_HOME/bin:$PATH
```

File	configuration item	parameter
sqlnet.ora	naming resolution password wallet path	NAMES.DIRECTORY_PATH sqlnet.wallet_override wallet_location
ldap.ora	for directory based naming resolution	data of the LDAP server
tnsnames.ora	database service addresses	list of pairs of aliases with there definitions in its own syntax

Oracle Secure External Password Store

sqlnet.ora

```
# example sqlnet.ora
# define resolution order
# ezconnect = Default
NAMES.DIRECTORY_PATH=(ldap,tnsnames,ezconnect)
# look for passwords in a wallet and use a wallet
sqlnet.wallet_override=true
wallet_location = (source = (method=file)(method_data=(directory = <path to wallet>)))
```

Be aware that sqlnet.ora has very sensitive formatting! Blank at the beginning of the line means that previous line is continued. If the first character of a line is not a blank, then it is supposed to be new directive.

ldap.ora (not necessary for our example)

```
DIRECTORY_SERVERS= ( <DNS of LDAP Server>:389 )
Default_Admin_Context = "dc=oracle,dc=<…>,dc=<…>"
DIRECTORY_SERVER_TYPE = AD
```

tnsnames.ora

```
<alias> =
     (DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(Host= <DNS)(PORT=1521))
                  (CONNECT_DATA=(SERVICE_NAME= <servicename>))
     )
```

Oracle Secure External Password Store

(1) make a password store on the client

```
$TNS_ADMIN>
> mkstore -wrl $TNS_ADMIN -create
Oracle Secret Store Tool : Version 11.2.0.1.0 - Production
Copyright (c) 2004, 2009, Oracle and/or its affiliates. All rights reserved.

Enter password:
Enter password again:
$TNS_ADMIN>
> ls -l
total 22
-rw-------   1 oracle   oinstall    3589 Nov 13 13:19 cwallet.sso
-rw-------   1 oracle   oinstall    3512 Nov 13 13:19 ewallet.p12
-rw-------   1 oracle   oinstall      49 Nov  6 08:07 sqlnet.ora
-rw-------   1 oracle   oinstall      49 Nov  6 08:07 tnsnames.ora
```

N.B.: choose a wallet password (and remember it !!)

ewallet.p12: PKCS#12 wallet file (public-key cryptography standard)
cwallet.sso: SSO wallet file: obfuscated mirror copy of the ewallet.p12

Oracle Secure External Password Store

(2) prepare sqlnet.ora and tnsnames.ora

(3) place credentials of <user>@<alias> in wallet

```
> mkstore -wrl $TNS_ADMIN -createCredential '<alias>' '<user>' <password>
Oracle Secret Store Tool : Version 11.2.0.1.0 - Production
Copyright (c) 2004, 2009, Oracle and/or its affiliates. All rights reserved.
Enter wallet password:

Create credential oracle.security.client.connect_string1
```

N.B.: you will be asked the wallet password

Oracle Secure External Password Store

(4) list wallet entries

```
> mkstore -wrl $TNS_ADMIN -listCredential
Oracle Secret Store Tool : Version 11.2.0.1.0 - Production
Copyright (c) 2004, 2009, Oracle and/or its affiliates. All rights reserved.
Enter wallet password:

List credential (index: connect_string username)
1: <alias> <user>
```

(5) test "passwordless" login

```
> sqlplus /@<alias>

Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - 64bit Production
With the Partitioning, Real Application Clusters and Automatic Storage Management
options

SQL> show user
USER is "<user>"
```

N.B.: the wallet password will NOT be asked

Oracle Secure External Password Store

Management of wallet entries

mkstore [-wrl wrl]
[-list]
[-listCredential]
[-viewEntry]
[-modifyCredential connect_string username password]
[-deleteCredential connect_string]
[-createCredential connect_string username password]
[-modifyEntry]

Important:

The connect_string can be considered like a primary key in the wallet.
For connecting 2 different users to the same DB you need 2 aliases.

Oracle Secure External Password Store

```
> mkstore -wrl $TNS_ADMIN -nologo -list
Enter wallet password:   ***
Oracle Secret Store entries:
oracle.security.client.connect_string1
oracle.security.client.password1
oracle.security.client.username1

> mkstore -wrl $TNS_ADMIN -nologo -listCredential
Enter wallet password:   ***
List credential (index: connect_string username)
1: USQASOP02_BACKUP sys

> mkstore -wrl $TNS_ADMIN -nologo -viewEntry oracle.security.client.connect_string1
Enter wallet password:   ***
oracle.security.client.connect_string1 = USQASOP02_BACKUP

> mkstore -wrl $TNS_ADMIN -nologo -viewEntry oracle.security.client.username1
Enter wallet password:   ***
oracle.security.client.username1 = sys

> mkstore -wrl $TNS_ADMIN -nologo -viewEntry oracle.security.client.password1
Enter wallet password:   ***
oracle.security.client.password1 = pwd   (displayed in clear text)
```

Oracle Secure External Password Store

```
> mkstore -wrl $TNS_ADMIN -nologo -modifyCredential <alias> username newpwd
Enter wallet password:   ***

Modify credential
Modify 1

> mkstore -wrl $TNS_ADMIN -nologo -modifyEntry oracle.security.client.connect_string1 <newalias>
Enter wallet password:   ***


> mkstore -wrl $TNS_ADMIN -nologo -deleteCredential <alias>
Enter wallet password:   ***

Delete credential
Delete 1
```

Oracle Secure External Password Store

Instant Client

- contains no wallet management utility
- workaround: create wallet with server binaries owner and copy afterwards

Works also with Easy Connect syntax (EZCONNECT)

no tnsnames.ora or LDAP directory necessary
wallet mechanism works in the same way
instead of aliases the connect strings will be used

```
# output formatted for better readability
mkstore -wrl ${TNS_ADMIN}/wallet
        -createCredential "DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)
                           (HOST=uiqrtosq0.internal.unicreditgroup.eu)(PORT=1521))
                           (CONNECT_DATA=(SERVICE_NAME=UIQRTOSQ0)))" <username> <password>


> mkstore -wrl ${TNS_ADMIN}/wallet -nologo -listCredential

List credential (index: connect_string username)
22: (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)
                  (HOST=uiqrtosq0.internal.unicreditgroup.eu)(PORT=1521))
                  (CONNECT_DATA=(SERVICE_NAME=UIQRTOSQ0))) username
```

Oracle Secure External Password Store

Automation and Security of Wallet management

- Wallet password can be omitted.
- With 11gR2 wallets have an auto_login functionality.

```
> orapki wallet create -wallet $TNS_ADMIN -auto_login_only
>ls -l
total 8
-rw------- 1 tuoraumg atwuser 3589 Apr  5 12:56 cwallet.sso
```

- to enhance security it is possible to prevent the wallet usage on other servers:

```
> orapki wallet create -wallet $TNS_ADMIN -auto_login_local
```

- After creation the wallet may be managed using the mkstore command.

Change the wallet password

```
> orapki wallet change_pwd -wallet $TNS_ADMIN oldpwd pwd1 -newpwd pwd2
```

Oracle Secure External Password Store

Using a Secure External Password Store with the JDBC Thin Driver

- create and fill the wallet as described before

JDBC Source Program

```
import java.sql.*;
import java.util.Properties;
import oracle.jdbc.pool.OracleDataSource;

public class Class2 {
public static void main(String[] args) throws SQLException {


DriverManager.registerDriver(new oracle.jdbc.OracleDriver());
String url = "jdbc:oracle:thin:/@(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(
                                        HOST=<DNS>)(PORT=1521))(CONNECT_DATA=(SERVICE_NAME=<service>)))";
Properties props = new Properties();

Connection conn = DriverManager.getConnection(url,props);
Statement stmt = conn.createStatement();
ResultSet rset = stmt.executeQuery(
"select 'Connected as ' || user from user_users");
while (rset.next())
System.out.println(rset.getString(1));
rset.close();
stmt.close();
conn.close(); }
}
```

Oracle Secure External Password Store

Compiling and Running

```
export ORACLE_HOME=…
export JAVA_HOME=…
export TNS_ADMIN=…

$JAVA_HOME/bin/javac
           -cp $ORACLE_HOME/jdbc/lib/ojdbc6.jar:$ORACLE_HOME/jlib/oraclepki.jar:. Class2.java


$JAVA_HOME/bin/java -cp $ORACLE_HOME/jdbc/lib/ojdbc6.jar:$ORACLE_HOME/jlib/oraclepki.jar:.
                    -Doracle.net.wallet_location=$TNS_ADMIN Class2
Connected as SCOTT
```

The wallet location can also be specified inside the JDBC source code as:

```
// Set the wallet location
props.setProperty("oracle.net.wallet_location",
"(SOURCE=(METHOD=file)(METHOD_DATA=" +
"(DIRECTORY=<wallet dir>)))");
```

Please note that in addition to the JDBC driver you will need the oraclepki.jar.

Oracle Secure External Password Store

References

OTN

http://www.oracle.com/technetwork/database/security/twp-db-security-secure-ext-pwd-stor-133399.pdf

My Oracle Support

Using The Secure External Password Store [ID 340559.1]
How To Change The Wallet Password For A Secure External Password Store? [ID 557382.1]
Using a Secure External Password Store with the JDBC Thin Driver (Doc ID 1441745.1)

Proxy Authentication

Principle: Grant a database user the privilege to assume the identity of another

- Authorization Granularity

```
SQL>alter user A grant connect through B with all roles except <…>;
SQL>alter user A grant connect through B with no roles;
SQL>alter user A grant connect through B with role C;
SQL>alter user A grant connect through B;
```

- for use with Enterprise User Security (EUS)

```
SQL>alter user A grant connect through B authentication required;
SQL>alter user A grant connect through B authenticated using
    Distinguished Name;
SQL>alter user A grant connect through Enterprise Users;
```

Proxy Authentication

Example

```
SQL>create user proxy identified by proxypw;
SQL>grant connect to proxy;
SQL>alter user client grant connect through proxy with role read_only;
SQL>connect proxy[client]/proxypw

SQL> select sys_context('USERENV','CURRENT_SCHEMA') CURRENT_SCHEMA,
            sys_context('USERENV','SESSION_USER')   SESSION_USER,
            sys_context('USERENV','PROXY_USER')     PROXY_USER
     from dual;


CURRENT_SCHEMA   SESSION_USER   PROXY_USER
--------------   ------------   ----------
client           client         proxy
```

Proxy Authentication

Administration

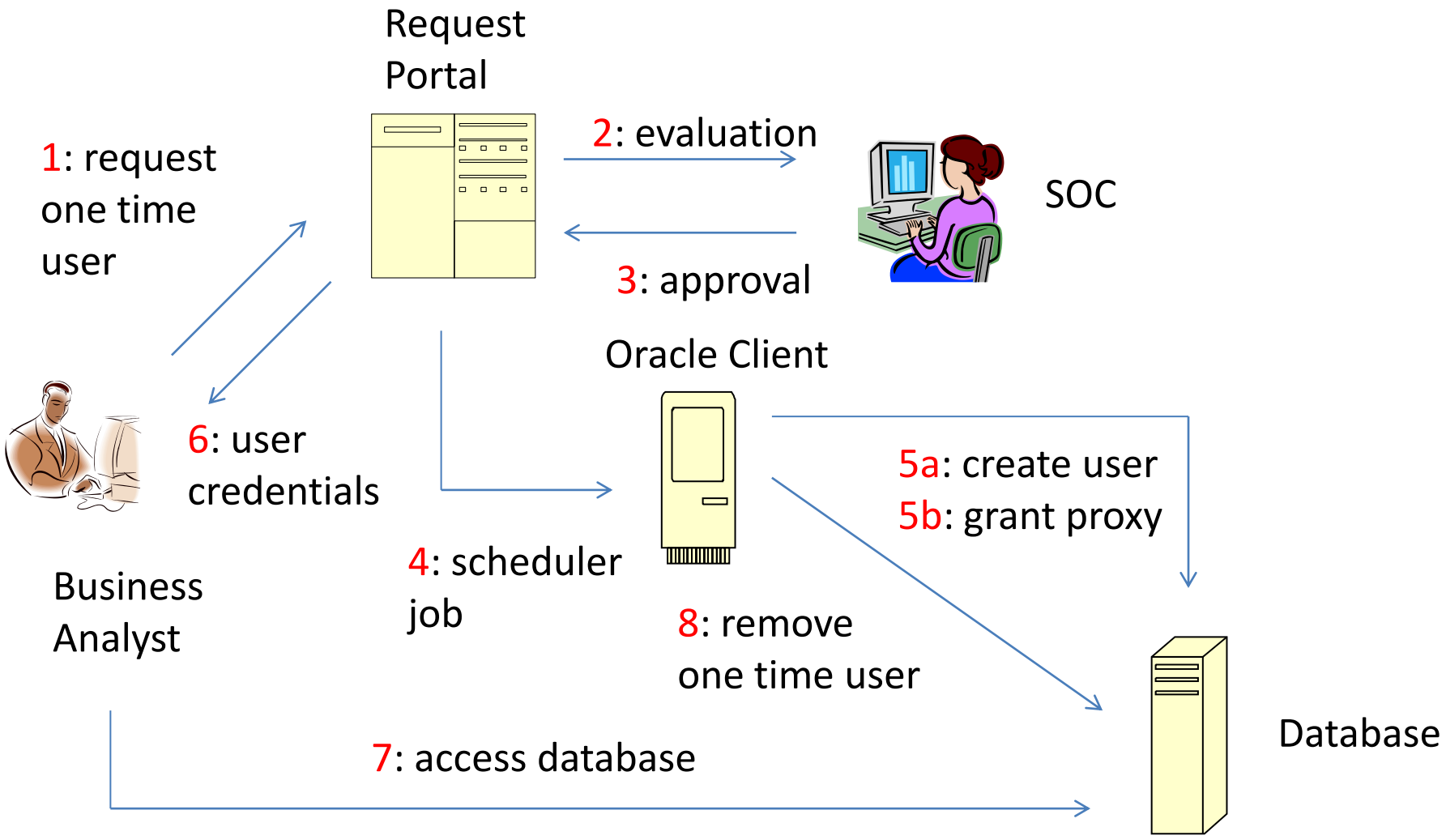
```
SQL>select * from dba_proxies where proxy='PROXY';

PROXY CLIENT AUT AUTHORIZATION_CONSTRAINT            ROLE       PROXY_AUT
----- ------ --- ---------------------------------- ---------- ---------
proxy client NO  PROXY MAY ACTIVATE ROLE            READ_ONLY  DATABASE


SQL>select * from proxy_roles;


PROXY            CLIENT           ROLE
---------------- ---------------- ---------
PROXY            CLIENT           READ_ONLY
```

Proxy Authentication

```
SQL>connect proxy[client]
select * from user_role_privs;

USERNAME                       GRANTED_ROLE                   ADM DEF OS_
------------------------------ ------------------------------ --- --- ---
CLIENT                         CONNECT                        NO  YES NO
CLIENT                         DV_REALM_OWNER                 NO  NO  NO
CLIENT                         READ_ONLY                      YES NO  NO
CLIENT                         READ_WRITE                     YES NO  NO
PUBLIC                         DV_PUBLIC                      NO  YES NO

SQL>set role read_write;
CLIENT@USORAOD91> set role read_write;
set role read_write
*
ERROR at line 1:
ORA-28156: Proxy user 'PROXY' not authorized to set role 'READ_WRITE' for client 'CLIENT'

CLIENT@USORAOD91> set role read_only;

Role set.
```

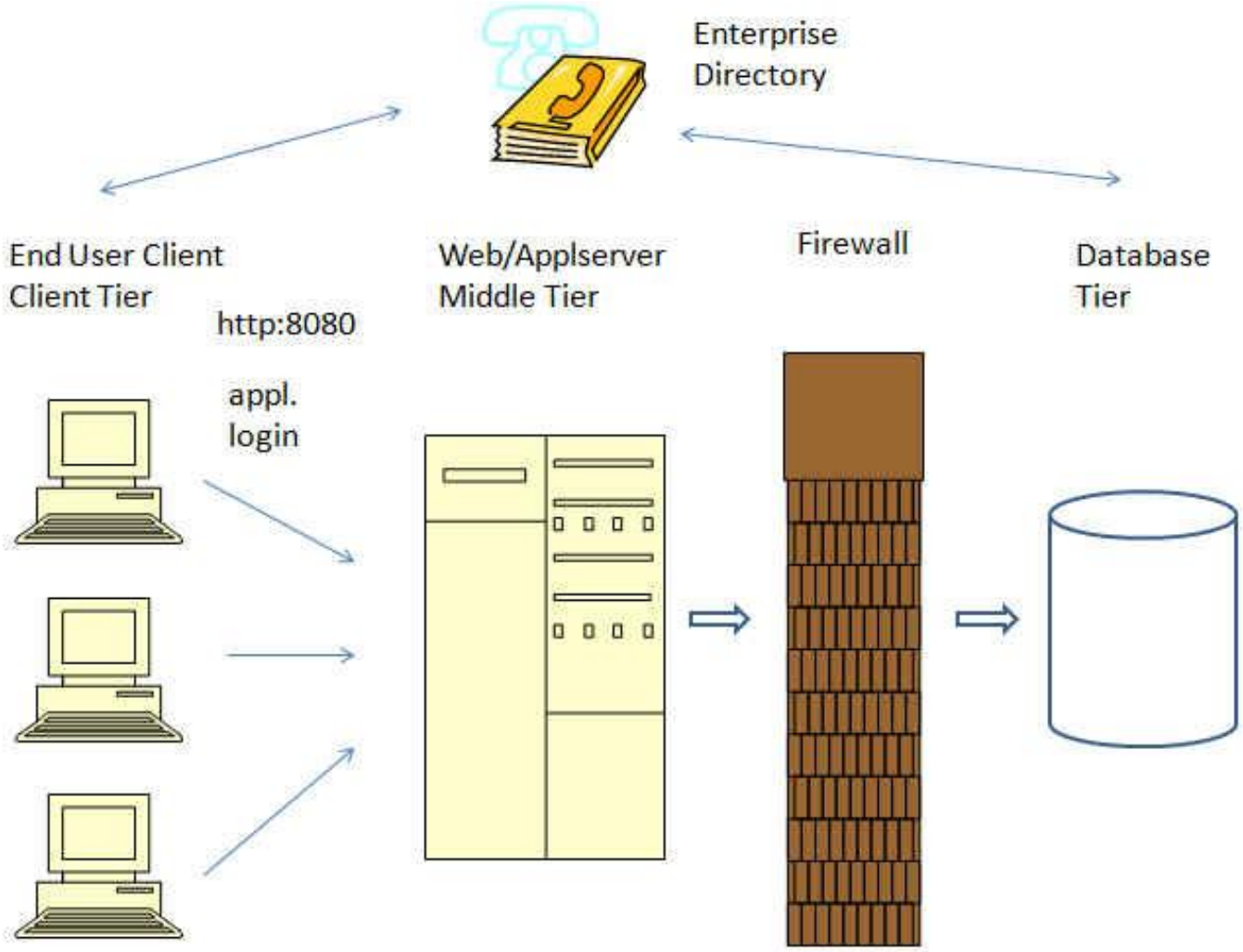
Proxy Authentication

Use Case: Controlled Access to the DB when application bypass is needed



Proxy Authentication

Use Case: Multi Tier Application



Proxy Authentication

- end user is captured by middle tier
- middle tier instead of database is authenticating the end user
- middle tier needs to invoke roles and privileges in the database on behalf of the end user

- challenges with regard to data access control:
  - identify the "real" end user
  - authenticate the end user => access limitation to objects and actions in the database
  - audit user activities in the database

Proxy Authentication

Architecture Approaches

1. Pass – through (Client / Server)
   1:1 relationship end user : db user
   user authentication in the database
   unsuitable for Web Applications

2. middle tier is responsible for user security in DB
   application user account has all privs for all end users in the system (contradicts the "least privilege" principle)
   difficult to audit

Proxy Authentication

3. Re – Authentication of the end user in the DB
   - application forwards login information to the DB for authentication

4. end user to session mapping via token – passing
   true end user not known to the DB
   token can be used for auditing
   connection pooling
   (dbms_session.set_client_identifier)

5. Proxy – Authentication
   - makes use of enterprise directory as trusted authority

Proxy Authentication